

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

**ANDRÉ ALVES DA CRUZ**

**DESENVOLVIMENTO DE UM SOFTWARE PARA CAPTURA E  
PROCESSAMENTO DE SINAIS ELETROFISIOLÓGICOS.**

NATAL - RN  
2014

**ANDRÉ ALVES DA CRUZ**

**DESENVOLVIMENTO DE UM SOFTWARE PARA CAPTURA E  
PROCESSAMENTO DE SINAIS ELETROFISIOLÓGICOS.**

Monografia apresentada à Universidade do  
Estado do Rio Grande do Norte – UERN –  
como requisito obrigatório para obtenção do  
título de Bacharel em Ciências da  
Computação.

**ORIENTADOR:** Prof. Dr. Wilfredo Blanco Fiquerola

NATAL - RN

2014

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

Cruz, André Alves da.

Desenvolvimento de um software para captura e processamento de sinais eletrofisiológicos. / André Alves da Cruz. – Natal, RN, 2014.

62 f.

Orientador(a): Prof. Dr. Wilfredo Blanco Fiquerola

Monografia (Graduação em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Departamento de computação. Campus de Natal-RN.

1. Aplicação - Computação - Monografia. 2. Sinais eletrofisiológicos. 3. Processamento de sinais . I. Fiquerola, Wilfredo Blanco. II. Universidade do Estado do Rio Grande do Norte. III. Título.

UERN/ BC

CDD 004

ANDRÉ ALVES DA CRUZ

**DESENVOLVIMENTO DE UM SOFTWARE PARA CAPTURA E  
PROCESSAMENTO DE SINAIS ELETROFISIOLÓGICOS.**

Este trabalho foi julgado adequado para fazer jus aos créditos da Disciplina de “Trabalho de Conclusão de Curso - TCC”, do Departamento de Informática e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

ORIENTADOR: Prof. Dr. Wilfredo Blanco Fiquerola

DEFESA PÚBLICA em Natal, 07 de agosto de 2014.

**BANCA EXAMINADORA**

---

Profº. Dr. Wilfredo Blanco Fiquerola – UERN  
Departamento de informática - DI – Campus de Natal - UERN  
Orientador

---

Profº. Me. Isaac de Lima Oliveira Filho – UERN  
Departamento de informática - DI – Campus de Natal - UERN

---

Profª. Dr. Adriana Takahashi – UERN  
Departamento de informática – DI – Campus de Natal - UERN

A Deus, por ser essencial em minha vida, autor de meu destino, meu guia, socorro presente na hora da angústia. Minha mãe, Maria Verônica, por seu cuidado e dedicação nos momentos difíceis, que me deu a perseverança para seguir. Ao meu pai Francisco Padilha, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Aos meus irmãos Anderson e Júnior pelas ajudas e incentivos, não medindo esforços para que eu chegasse até esta etapa da minha vida.

## **AGRADECIMENTOS**

A Deus, pelo dom da vida, pela ajuda e proteção, pela Sua força e presença constante, e por me guiar à conclusão de mais uma preciosa etapa de minha vida.

A meus pais Francisco Padilha da Cruz e Maria Verônica Alves da Cruz, que me ensinaram a viver e que com muita confiança, dedicação, força e amor, me proporcionaram a realização deste sonho.

Aos meus irmãos e minha família, por compreender todos os momentos em que precisei estar ausente na constante busca da minha construção profissional. Estes que com muito carinho e apoio, nunca mediram esforços para que eu chegasse a concluir esta etapa.

A minha namorada Taís, por estar comigo nos momentos altos e baixos de minha vida sempre dando apoio incondicional.

Ao professor e orientador Wilfredo Blanco Figuerola, pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia.

A Universidade do Estado do Rio Grande do Norte, a todos os professores e funcionários, por todo apoio e inspiração no amadurecimento dos meus conhecimentos e conceitos que me levaram a conclusão deste estudo monográfico.

Aos coordenadores do Curso de Bacharelado em Ciência da Computação, Isaac Filho e Bartira Rocha, pelo convívio, apoio, compreensão e amizade.

Aos amigos Ewerton da Cruz, Nielson Soares, Edson Padilha e todos os companheiros de turma que acompanharam de perto esta batalha, me incentivando e apoiando em momentos difíceis. Jamais me esquecerei das dificuldades ultrapassadas e amizades conquistadas.

Meus sinceros agradecimentos a todos que, de alguma forma, contribuíram para a construção desta monografia.

*Sem sonhos, a vida não tem brilho. Sem metas, os sonhos não têm alicerces. Sem prioridade, os sonhos não se tornam reais. Sonhe, trace metas, estabeleça prioridade e corra riscos para executar seus sonhos. Melhor é errar por tentar do que errar por se omitir! Não tenhas medo dos tropeços da jornada. Não podemos esquecer que nós, ainda que incompletos, fomos o maior aventureiro da história.*

*Augusto Cury*

## RESUMO

As últimas descobertas científicas mostram que todas as formas de vida, principalmente as células, organizam-se seguindo o mesmo padrão e os mesmos princípios básicos: o padrão de redes, com células interconectadas a sistemas formando redes neurais complexas. No decorrer deste novo século novos métodos para pesquisar o sistema nervoso estão sendo desenvolvidos, relacionando várias áreas científicas como: biologia, eletrofisiologia e computação. Dentre estas áreas destacam-se os conceitos de fisiologia do sistema nervoso dos seres vivos, ligados ao avanço da computação. A junção destes meios de pesquisa a sistemas neurais trazem novos rumos ao universo científico. Este trabalho teve como objetivo o desenvolvimento de um *software* livre capaz de capturar, visualizar e processar sinais eletrofisiológicos para registro e auxílio de pesquisas científicas. Objetiva, também, incentivar os alunos do Ensino Médio de escolas públicas da região a se tornarem pesquisadores. Tendo em vista as adversidades que o sistema educacional brasileiro enfrenta, este *software* ajudará na compreensão e evolução do ensino em aulas práticas de biologia ofertadas pelos professores do Ensino Médio.

**Palavras-chave:** Aplicação computação. Sinais eletrofisiológicos. Sistema nervoso. Processamento de sinais.

## **ABSTRACT**

The latest scientific discoveries show that all forms of life, especially the cells, organize themselves following the same pattern and the same basic principles: The pattern networks, with cells interconnected systems forming the complex neural networks. In the course of this this century, new methods to research the nervous system are being developed, linking various scientific fields in biology: biology, electrophysiology and computing. Among these areas highlight the concepts of physiology of the nervous system of living beings, aligned to discovery and the advancement of computing. The junction of these means of neural systems research to bring new direction to scientific universe. This paper aims to develop a free software able to capture, viewing and processing electrophysiological signals for recording and aid scientific research. It will provide for and encourage high school students from public schools in the region, to become researchers. Considering the adversities that the Brazilian educational system is, this software helps the understanding and development of biology teaching practices in lessons offered by the school teachers.

**Keywords:** Computer Application, Electrophysiological Signals, Nervous System, Signal Processing.

## LISTA DE FIGURAS

Figura 1 - Kit didático e seus componentes.....	13
Figura 2 - Cenário do Kit didático e seus componentes de <i>hardware</i> e <i>software</i> ....	14
Figura 3 - Representação de um neurônio biológico.....	17
Figura 4 - Representação esquemática do sistema nervoso de vertebrados.....	18
Figura 5 - Gráfico da forma de onda de um potencial de ação.....	20
Figura 6 - Representação transmissão de uma forma de onda.....	21
Figura 7 - Representação da sinapse.....	21
Figura 8 - Comparação entre sinapse elétrica e sinapse química.....	22
Figura 9 - Experimento em um cérebro de um coelho.....	23
Figura 10 - Sinal original ( <i>Figura A</i> ) e amostragem deste sinal ( <i>Figura B</i> ).....	24
Figura 11 - Exemplo de um sinal banda limitada.....	25
Figura 12 - Plot do ECG.....	27
Figura 13 - Plot do EMG.....	27
Figura 14 - Potencial de ação SPIKE de um neurônio.....	28
Figura 15 - Máquina de estados do sistema.....	33
Figura 16 - Diagrama de caso de uso do sistema.....	34
Figura 17 - Diagrama de sequência do usuário.....	37
Figura 18 - Diagrama de sequência do sistema.....	38
Figura 19 - Diagrama de Classes.....	40
Figura 20 - GUI do software ASE.....	42
Figura 21 - Janela de Novo Projeto.....	44
Figura 22 - Visualização dos dados na GUI do ASE.....	45
Figura 23 - Visualização dos Spikes gravados em XML dos canais esquerdo e direito	45
Figura 24 - Funções de Zoom-in, Zoom-out e Limiar do ASE.....	46
Figura 25 - Frequência de disparos, nível do sinal e Relógio.....	47
Figura 26 - Produto Final Kit Didático.....	48

## **LISTA DE TABELAS**

Tabela 1 - Parâmetros fixos em relação ao potencial de membrana do neurônio.. 28

## LISTA DE ABREVIATURAS

ASE: ANÁLISE DE SINAIS ELETROFISIOLÓGICOS.

GUI: Interface gráfica do usuário.

SPIKE: Potencial de Ação.

SPIKE TRAINS: Trem de pico.

UML: Linguagem Unificada de Modelagem.

IDE QT CREATOR 5.0.2: Ambiente de desenvolvimento integrado de software.

ICe: Instituto do Cérebro.

C++: Linguagem de Programação.

POO: Programação Orientada a Objetos.

QT: Toolkit de desenvolvimento de software com interface gráfica.

QWT: Qt Widgets para aplicações técnicas.

BACK-END: Etapa inicial coleta as funções de entrada do usuário.

UFRN: Universidade Federal do Rio Grande do Norte.

CABO P2: Cabo com conector para transmissão de sinais analógicos.

RN: Redes Neurais.

RNA: Redes Neurais Artificiais.

OPEN-SOURCE: Software Livre.

PD: Pure Date.

EEG: Eletroencefalograma.

ECG: Eletrocardiograma.

EMG: Eletromiograma.

PM: Potencial de Membrana.

SNC: Sistema Nervoso Central.

SNP: Sistema Nervoso Periférico.

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 KIT DIDÁTICO.....	12
1.2 MOTIVAÇÃO E JUSTIFICATIVA.....	15
1.3 OBJETIVOS.....	15
1.4 OBJETIVOS ESPECÍFICOS.....	15
1.5 ESTRUTURA DO TRABALHO.....	16
<b>2 SISTEMA NERVOSO.....</b>	<b>17</b>
2.1 MEMBRANA NEURAL E POTENCIAL DE MEMBRANA.....	19
2.2 REDES NEURAIS.....	19
2.3 POTENCIAL DE AÇÃO.....	20
2.4 SINAIS BIOLÓGICOS E ELETROFISIOLOGIA.....	23
<b>2.4.1 Teoria da amostragem.....</b>	<b>24</b>
<b>2.4.2 Teorema de nyquist-shannon.....</b>	<b>26</b>
<b>2.4.3 Características do sinal.....</b>	<b>26</b>
<b>3 METODOLOGIA.....</b>	<b>30</b>
3.1 LINGUAGENS E TECNOLOGIAS UTILIZADAS.....	30
3.2 MODELAGEM.....	31
<b>3.2.1 Máquina de estados.....</b>	<b>32</b>
<b>3.2.2 Diagrama de caso de uso.....</b>	<b>34</b>
<b>3.2.3 Diagramas de sequência.....</b>	<b>36</b>
<b>3.2.4 Diagrama de classes.....</b>	<b>39</b>
<b>4 RESULTADOS.....</b>	<b>42</b>
4.1 PRODUTO FINAL.....	48
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>51</b>
<b>6 TRABALHOS FUTUROS.....</b>	<b>52</b>
<b>REFERÊNCIA BIBLIOGRÁFICA.....</b>	<b>54</b>
<b>APÊNDICE A – DETALHAMENTO DO DIAGRAMA DE CLASSES.....</b>	<b>57</b>

## 1 INTRODUÇÃO

Por ser o cérebro o órgão que gerencia o sistema nervoso do organismo estão sendo desenvolvidas diversas pesquisas para entender o seu funcionamento. Ele é composto por bilhões de células (*neurônios*) conectadas que formam uma rede complexa, permitindo comandar e processar funções como equilíbrio, percepção, sentimentos e aprendizagem.

Os neurônios transmitem sinais entre eles através de uma estrutura chamada axônio capaz de transmitir impulsos nervosos, chamados também de potenciais de ação (figura 5). Estes sinais podem ser gravados através da utilização de eletrodos conectados a um amplificador para melhorar e filtrar os sinais capturados. Sinais estes, representados por uma forma de onda elétrica que percorre a membrana de uma célula transmitindo informações e estímulos entre os neurônios conectados. A sequência de potenciais de ação (disparos), denominadas de trens de pulsos - "*Spike trains*" - variam a frequência ou intervalo de tempo modulando assim as informações que são transmitidas.

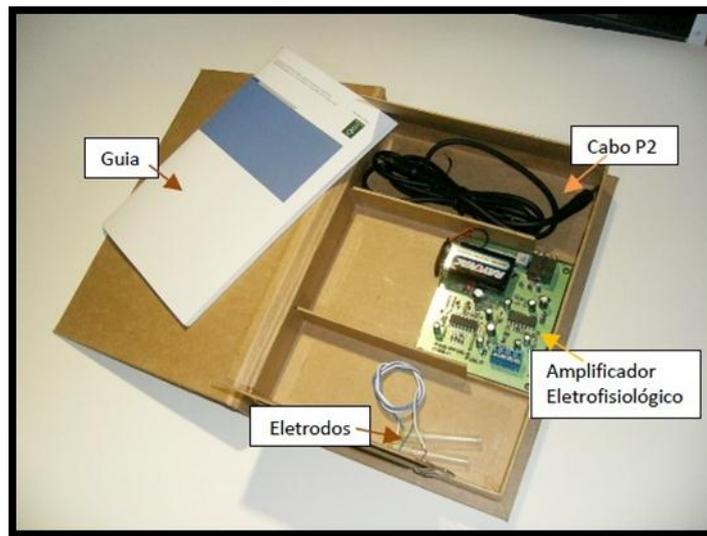
### 1.1 KIT DIDÁTICO

No intuito de repassar e obter os conhecimentos das principais funções do cérebro está sendo desenvolvido um projeto por professores da Universidade Federal do Rio Grande do Norte (UFRN), reunindo os departamentos de Biofísica, Farmacologia, Engenharia Biomédica e Instituto do Cérebro, juntamente com o departamento de Computação da Universidade do Estado do Rio Grande do Norte. Este projeto de ações integradas apresenta de maneira original e intuitiva a atividade elétrica do corpo aos alunos do Ensino Médio, da rede pública, realizando atividades práticas de forma a auxiliar a aquisição e apreciação do conhecimento científico nas áreas da Biologia, Eletrofisiologia e Computação.

O objetivo principal deste trabalho é desenvolver um *software* que será um componente do kit didático, (figura 26, página 50), de baixo custo e fácil utilização, capaz de registrar potenciais eletrofisiológicos em insetos, principalmente nas extremidades dos membros de locomoção dos mesmos. Este kit inicial ilustrado na figura 1 é composto por:

- Sistema de amplificação de sinais biológicos (*hardware*): estabelece conexão com o computador através da utilização de um cabo P2 (cabo com conector para transmissão de sinais analógicos) ligado a entrada de microfone do computador.
- Software: Usado na captura, processamento e visualização de sinais eletrofisiológicos.
- Eletrodos: Conector que transfere sinais elétricos para o amplificador (*hardware*).

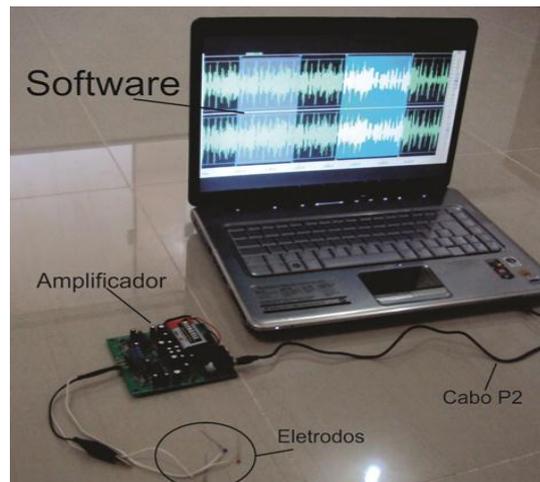
Figura 1. Kit didático e seus componentes (*hardware*).



Fonte: Elaborada pelo autor.

Este kit, mostrado na figura 1, tem como principal função registrar os potenciais bioelétricos (potenciais de ação) em aulas práticas, fazendo registros das atividades elétricas dos neurônios de insetos. Durante o uso do kit em oficinas práticas vinculadas ao estudo do sistema nervoso, os alunos interconectam diversos conhecimentos de Ciências Naturais, Exatas e Humanas, contribuindo com o desenvolvimento educacional dos mesmos e, partindo do método científico, associa-se aos conceitos em sala de aula recomendados pelas diretrizes dos Parâmetros Curriculares Nacionais (PCN's), e como, ainda, pode-se confirmar em Freire (1997, p.11) “para se compreender a teoria é preciso experiênciá-la”. Portanto, com uso de atividades práticas, será ampliado o conhecimento científico individual de alunos e professores tornando as aulas mais agradáveis e dinâmicas para ambos e com melhores resultados no que diz respeito ao aprendizado.

Figura 2. Cenário do kit didático e seus componentes de *hardware* e *software*.



Fonte: Elaborada pelo autor.

Diversos autores ressaltam “a importância das aulas práticas no ensino, sendo esta atividade altamente apreciada pelos alunos” (LEITE E COLS, 2000). Esta metodologia de trabalho sugerida por esse projeto ajudará os alunos a entenderem os fenômenos e mistérios envolvendo o cérebro, por isso para que alunos e docentes tenham uma melhor compreensão a respeito, seria de grande apoio utilizar um *software* capaz de visualizar os sinais enviados e recebidos pelo cérebro. Por esse motivo surgiu a ideia de desenvolver uma aplicação computacional livre (*open-source*) para captura, processamento e visualização dos sinais eletrofisiológicos (ver figura 2), monitorando o uso do método científico a fim de responder perguntas sobre a função do sistema nervoso de insetos. Ao mesmo tempo possibilitará visualizar, de uma maneira mais clara, a relação de conhecimentos de diversas áreas da neurociência de forma que alunos e professores do Ensino Médio, da rede pública, possam se apropriar dessas informações para um ensino-aprendizagem mais satisfatório, integrado e produtivo.

Hoje existem no mercado *softwares* que fazem a captura e visualização de sinais, como por exemplo, Audacity (AUDACITY, 2013), Pure Date (MILLER, 2014), Medplot (ORTIS, R. S. et al, 2013), dentre outros. Estes *softwares* não são específicos para captura, visualização e processamento de sinais eletrofisiológicos, daí, a ideia de desenvolver um *software* analisador de sinais eletrofisiológicos (*ASE*), específico para essas análises e processamentos de sinais visando o público alvo do projeto. O *ASE* tem como principal função repassar informações científicas mais compreensíveis, auxiliando os usuários a terem uma melhor interação com o sistema.

## 1.2 MOTIVAÇÃO E JUSTIFICATIVA

O rápido desenvolvimento da computação, em especial dos microprocessadores, permitiu uma evolução significativa das técnicas de captura, análise e processamento digital de sinais, influenciando diretamente os setores do conhecimento da neurociência.

Essa evolução trouxe maior flexibilidade e novas formas de pesquisas voltadas às áreas do conhecimento científico como a Biomédica, a Medicina e a Fisiologia e em especial à Engenharia Biomédica. Nessa área específica, é possível observar o avanço tecnológico dos equipamentos computacionais capaz de aumentar o processamento de sinais eletrofisiológicos.

Os equipamentos destinados à aquisição de sinais eletrofisiológicos existentes no mercado possuem elevado custo o que torna inviável a obtenção dos mesmos. Neste sentido, este trabalho propõe o desenvolvimento de um *software* capaz de visualizar sinais que são pré-amplificados através de um amplificador, cuja captura e controle seja feito via computador tornando, com isso, um sistema de custo reduzido e consequentemente mais acessível.

Pensando neste contexto, as atividades deste kit didático buscam inovar nessa área, inserindo a sociedade na *Iniciação Científica* através de um projeto de ação acadêmica buscando experimentos simples; mas com amplo conhecimento científico e, popularizando métodos quantitativos no estudo do sistema nervoso, apresentando ao público alvo - professores e alunos do Ensino Médio - como o sistema dos seres vivos se comporta através de experimentos e, ainda, descrevendo como determinados padrões de potenciais de ação produzem distintos padrões motores.

## 1.3 OBJETIVO

- Desenvolvimento de um software para captura, processamento e visualização de sinais eletrofisiológicos em insetos.

## 1.4 OBJETIVOS ESPECÍFICOS

- Desenvolver de uma interface (GUI) intuitiva.

- Capturar áudio através de um cabo P2 utilizado para estabelecer conexão entre o *Notebook* e o amplificador.
- Processar e visualizar o sinal com a utilização de funções básicas como *Zoon-in* e *Zoon-out*.
- Salvar os potenciais de ação (spikes) dos dois canais detectados em arquivos XML e o sinal contínuo em arquivos WAVE.
- Detectar potenciais de ação (spikes) neuronais determinado pelo valor do limiar, realizando contagem dos mesmos, calculando a determinada taxa de disparos em hertz.
- Validação e testes dos experimentos capturados pelo *software*.

## **1.5 ESTRUTURA DO TRABALHO**

Após o primeiro capítulo composto pela introdução, kit didático, justificativa e motivação e objetivo, os próximos capítulos serão abordados da seguinte forma.

No segundo capítulo é apresentado o referencial teórico sobre os principais conceitos e definições do sistema nervoso e os argumentos usados para o embasamento do resultado final desse trabalho. No terceiro é exposta a metodologia empregada na modelagem e desenvolvimento deste trabalho.

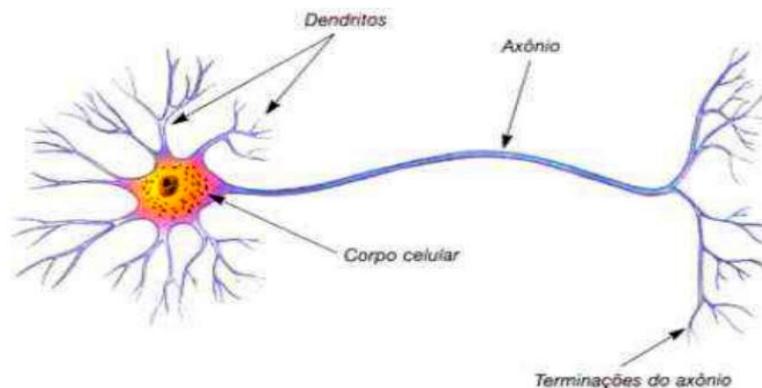
O quarto capítulo detalha os resultados finais obtidos a partir do desenvolvimento do ASE (analisador de sinais eletrofisiológicos), juntamente com o produto final (Kit Didático). O quinto capítulo é composto pelas considerações finais deste trabalho, e, por fim, o sexto capítulo traz a indicação para possíveis trabalhos futuros.

## 2. SISTEMA NERVOSO

O cérebro humano consiste de bilhões de elementos de processamento chamados neurônios. Esses neurônios são compostos por corpo celular, dendritos e axônio. Eles se comunicam por meio de uma rede de conexão formada por axônios e sinapses. Comandados pelo nosso sistema nervoso tem como principal função processar as informações que chegam através de suas unidades de entrada, transmiti-las pelas vias aferentes até o sistema nervoso central (SNC) e enviá-las, através das vias eferentes, para suas unidades de saída – os neurônios motores. Os neurônios sensoriais e motores constituem o elo entre SNC e o ambiente externo. Esta comunicação se processa essencialmente por impulsos elétricos e reações químicas (SOUZA, 2005).

Como citado anteriormente, o sistema nervoso central é capaz de processar informações e transformá-las em saídas. Para executar esse processamento conta com unidades que possuem a capacidade de captar os estímulos do ambiente e interpretá-los, podendo armazenar a informação resultante ou simplesmente gerar uma resposta na forma de impulsos nervosos (SOUZA, 2005). Tais unidades são chamadas de neurônios, como mostrado na figura 3.

Figura 3: Representação de um neurônio biológico.

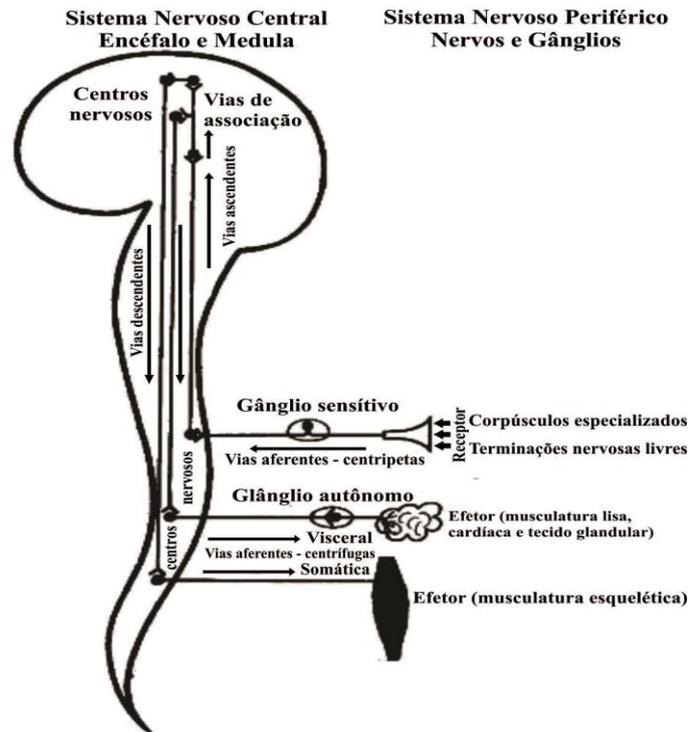


Fonte: FERNEDA, 2006

Como nos seres humanos, os seres vertebrados podem ser divididos em duas grandes partes: sistema nervoso central (SNC) e sistema nervoso periférico (SNP) (ERHART, 1974). Como é possível visualizar na figura 4, o SNC fica localizado na parte axial, sendo composto pelo encéfalo e medula espinal, já o SNP localiza-se na parte apendicular e é formado por nervos e gânglios.

O processamento das informações recebidas do SNC pelo SNP permite que funções importantes do organismo sejam coordenadas e executadas, tais como a respiração, os reflexos, as emoções, a memória entre tantas outras.

Figura 4: Representação esquemática do sistema nervoso de vertebrados.



Fonte: ERHART, 1974.

Portanto o sistema nervoso é capaz de transformar estas informações em saídas, reorganizando os movimentos do ser ao interagir com o ambiente, acarretando no preenchimento dos sentidos através de vários estímulos. Estes estímulos são responsáveis não somente pelo funcionamento normal do indivíduo, mas também pela aprendizagem do mesmo.

“A plasticidade neural ou neuroplasticidade é a capacidade de organização do sistema nervoso frente ao aprendizado e à lesão. Esta organização se relaciona com a modificação de algumas conexões sinápticas. A plasticidade nervosa não ocorre apenas em processos patológicos, mas assume também funções extremamente importantes no funcionamento normal do indivíduo. Os processos de modificação, em consequência da interação com o meio ambiente, e as conexões que se formam durante o aprendizado motor consciente (*memória*), são exemplos de como funcionam o cérebro dos indivíduos sem alteração do sistema nervoso central” (BASS, 2014, p.1).

Este funcionamento do cérebro pode ser visto, por exemplo, na percepção do campo visual, onde estrelas, alimentos, flores, caracterizados por forma, cor, posição e cheiro, em geral relacionam-se a outros e, mesmo, assim, lhes atribuímos nomes e conceitos distintos. Logo não existe um neurônio associado a cada objeto ou sequer um mapa composto de redes de neurônios para identificar cada objeto ou situação percebida. Pois existe um número enorme de neurônios, mas esse número é finito. O que se sabe, segundo Rezende (apud SILVA, 2003), é que existem conjuntos de neurônios especializados em analisar atributos distintos de cada objeto, localizados em áreas distintas do cérebro, razão pela qual deve haver um mecanismo que enlace temporalmente a atividade de todos eles de maneira que a percepção possa ser experimentada como uma unidade perceptual unificada. Essa ativação elétrica denominada de Spike, vinda de ligações de grupos de circuitos num padrão temporal e espacial específico produz a percepção, a emoção, a linguagem e a motricidade, caracterizando o comportamento e o fenômeno denominado consciência.

## **2.1 MEMBRANA NEURAL E POTENCIAL DE MEMBRANA**

A membrana neural é uma parte importante do neurônio, pois a partir dela é que se dá a transmissão das ondas ou sinais. Mais do que um simples delimitador, sua principal função é a filtragem de substâncias que entram ou saem do neurônio, criando assim, desequilíbrio na distribuição de íons e moléculas entre o interior e o exterior da membrana (SOUZA, 2005).

Esta transmissão de ondas, chamada de potencial de membrana, é a definição dada a essa diferença de carga elétrica que percorre a membrana neural. Os conceitos citados anteriormente sobre o potencial de membrana são muito importantes, pois são utilizados como principal meio de pesquisa visualizada pelo *software* proposto neste trabalho.

## **2.2 REDES NEURAIS**

Segundo Goldstein K. Paul Broca (1824-1880) tem-se amplos estudos no que diz respeito à afasia, uma espécie de deficiência da fala em pacientes com danos cerebrais e descobriu-se que o cérebro é responsável fundamental pelo trabalho de cada órgão.

Camilo Golgi, um dos precursores do estudo do sistema nervoso humano, em 1875, criou um método para análise e estudo de um neurônio, solitariamente, através da coloração do mesmo. Essa experiência deu um grande impulso na área da Biologia, mas ainda não revelaria o obscuro campo das muitas redes de estruturas neuronais (PARADISO, 2002).

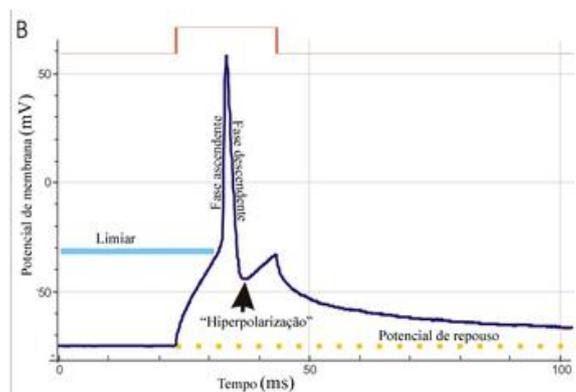
O estudioso Santiago Ramón y Cajal (p. 1054), seguindo a sombra de Golgi, adotou a noção de sistemas nervosos falando sobre a interligação das células por estímulos nervosos, descrevendo várias estruturas e clareando a idéia de que essas células trabalhavam em conjunto, cada uma com as suas especialidades, para uma estrutura global de propósitos mais gerais denominadas de redes neurais. A formação destas redes dependerá principalmente dos neurotransmissores modificando o comportamento da rede de acordo com as suas necessidades; a isso damos o nome de aprendizado de rede.

### 2.3 POTENCIAL DE AÇÃO

Existem dentro e fora das células diferentes concentrações, isso pode ocasionar variações muito rápidas de voltagem das células. Nos períodos em que o neurônio não está transmitindo informações seu potencial de membrana fica em estado de repouso.

Quando ocorre a perturbação da membrana é criado o fenômeno de diferença de potencial, ou seja, uma sucessão de eventos fisiológicos que ocorrem através da membrana. Tais fenômenos, importantes para o funcionamento de um neurônio, em conjunto, produzem aquilo que se chama potencial de ação, como mostra a forma de onda da figura 5.

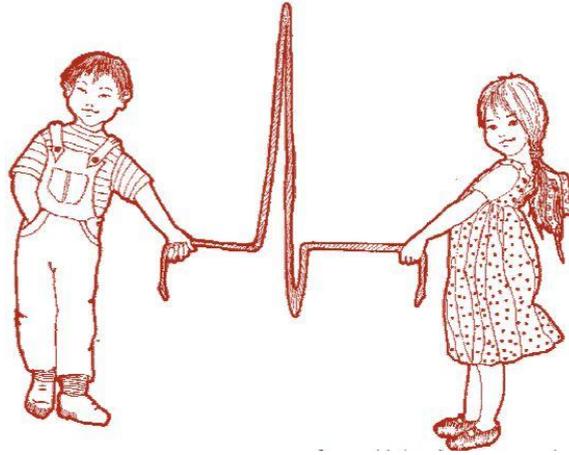
Figura 5. Gráfico da forma de onda de um potencial de ação.



Fonte: (SPECTROGRAMS, 2013)

Este potencial pode ser provocado por uma estimulação química ou elétrica, originada em uma transmissão como mostra a figura 6.

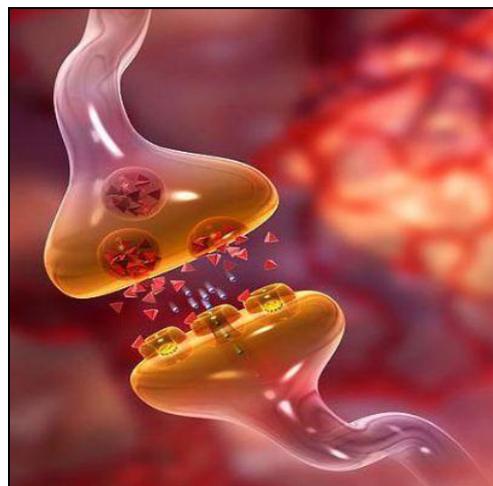
Figura 6: Representação transmissão de uma forma de onda.



Fonte: KRANTS, 2013.

O potencial a ser gerado age inicialmente nas extremidades e se propaga em uma única direção, não retornando pelo caminho já percorrido. Consequentemente, os potenciais de ação são unidirecionais, o que se pode chamar de condução. Com essa propagação a sinapse é gerada para a liberação de moléculas e neurotransmissores como é mostrado na figura 7. Esses neurotransmissores se ligam a receptores específicos do próximo neurônio desencadeando o mesmo processo e dando continuidade aos impulsos nervosos.

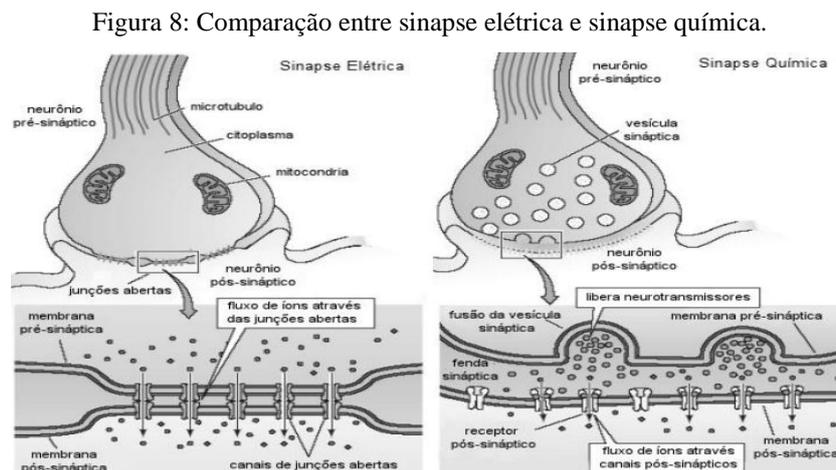
Figura 7: Representação da sinapse.



Fonte: NEUROMED, 2011.

Conforme Souza (2005), sinapse é o ponto de “contato” entre um neurônio e outra célula, seja ele músculo ou glândula, para a transmissão de informações. Logo o neurônio que envia a informação é chamado neurônio pré-sináptico; enquanto o neurônio que recebe esta transmissão e denominado de pós-sináptico. Na maioria das sinapses a transformação sináptica é feita através de um espaço, a fenda sináptica, onde são lançadas as substâncias mensageiras, chamadas neurotransmissores. Além deste papel, as sinapses determinam a direção que o impulso trafega pelo sistema nervoso, bloqueando os mais fracos e permitindo a passagem dos mais fortes, fazendo com que se tornem repetitivos, ou seja, integrados, para gerar padrões de potencial de ação, os *spikes trains*.

Existem dois tipos de sinapses: sinapses elétricas e sinapses químicas. Na figura 8 é possível observar a diferença entre elas.



Fonte: PURVES et al, 2004.

As sinapses elétricas têm como principal função permitir a passagem direta de sinais elétricos entre os neurônios pré e pós-sináptico. Esta transmissão é feita através de canais chamados junção aberta que, por serem canais largos, permitem a passagem da maioria dos íons celulares. Além disso, permitem também a passagem de pequenas moléculas orgânicas (SOUZA, 2005). Já as sinapses químicas são as mais comuns no SNC, e o maior benefício desse tipo de sinapse é a capacidade de controlar a passagem de informações entre neurônios.

## 2.4 SINAIS BIOLÓGICOS E ELETROFISIOLOGIA

A eletrofisiologia consiste no estudo das propriedades elétricas em células e tecidos. Envolve medições de diferenças de potencial elétrico numa vasta variedade de escalas, desde simples proteínas de canais até órgãos completos, como por exemplo, o coração e o cérebro. Na Neurociência inclui medidas das atividades elétricas de neurônios e, particularmente, da atividade do potencial de ação na condução de estímulos nervosos que ocorrem nas fibras neuromusculares excitáveis.

O cientista Leão (2007), descobriu esses fenômenos quando pesquisava a epilepsia experimental, nos Estados Unidos. Em um experimento, em 1943, colocou uma fileira de eletrodos na superfície do cérebro de um coelho, como ilustrado na figura 9. O primeiro eletrodo estimularia a região enquanto os outros registrariam as ondas elétricas. Dessa forma, ele constatou que, ao ser estimulada, a região sofria uma depressão intensa e duradora com oscilação elétrica espontânea que se expandia lentamente e, ao fim de cerca de 15 minutos, retornava à normalidade.

Figura 9: Experimento em um cérebro de um coelho.



Fonte: LEÃO, 2007.

Com o desenvolvimento dessas pesquisas ele comprovou que esses fenômenos não se restringem apenas ao córtex do cérebro, mas pode ser identificado em outras estruturas neurais. Já outros pesquisadores se dedicaram as pesquisas em rotinas isoladas de insetos, sobretudo sapos, pintos, camundongos, grilos e em especial a barata.

Na Engenharia, é entendido como sinal qualquer evento que carregue informação, e é nessa direção que os sinais biológicos passaram a ser definidos. Alguns variam no tempo de forma lenta, outros de forma rápida, como por exemplo, *SPIKE* (*potencial de ação*), que por sua vez dura aproximadamente 1ms. Além disso, a energia física

presente em um sinal biológico pode ser de natureza elétrica como *EEG* (*eletroencefalograma*), *ECG* (*eletrocardiograma*), *EMG* (*eletromiograma*) e *SPIKE* (*potencial de ação*).

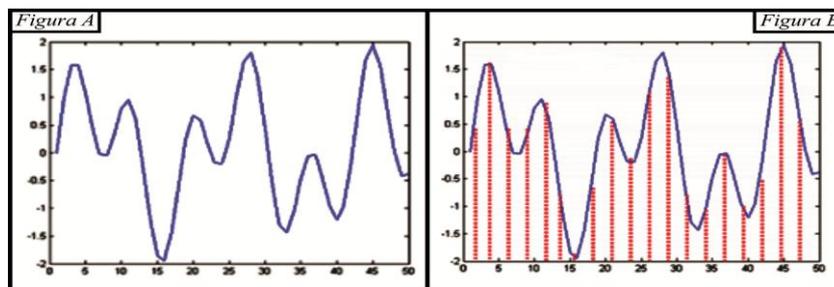
Com a evolução tecnológica a análise desses sinais biológicos passou a ser realizada por computadores. Para que isso aconteça, entretanto, é necessário capturar e visualizar os sinais de maneira contínua, tomando medidas de amplitude do sinal em intervalos regulares. Este processo é chamado de amostragem de sinal no tempo.

A visualização dos sinais é um processo que deve ser executado simultaneamente, porém o *software* para a plotagem dos sinais é um sistema em tempo crítico, e o atraso na plotagem poderá prejudicar a sua análise. No caso de sinais onde a taxa de amostragem é mais baixa, a atualização na tela não afeta a visualização.

#### 2.4.1 TEORIA DA AMOSTRAGEM

Sinais discretos no tempo podem ser gerados de diferentes formas, o mais comum é sendo uma representação de sinais contínuos no tempo. Isso é devido ao fato de que o processamento de sinais é feito através do processo discreto. Pode-se ver um exemplo de sinal contínuo no tempo ilustrado na figura 10.

Figura 10 – Sinal original (*Figura A*) e amostragem deste sinal (*Figura B*).



Fonte: ALEXANDRE, 2013.

A melhor forma de obter uma representação discreta no tempo de um sinal é através de uma amostragem periódica, quando a sequência é obtida de um sinal contínuo no tempo. Na prática, a operação de amostragem é executada por um conversor analógico para digital (A/D). No Kit apresentado, este circuito foi desenvolvido pelo professor George Nascimento do Departamento de Engenharia Biomédica da UFRN. Para a implantação ou escolha de um conversor foi considerado a

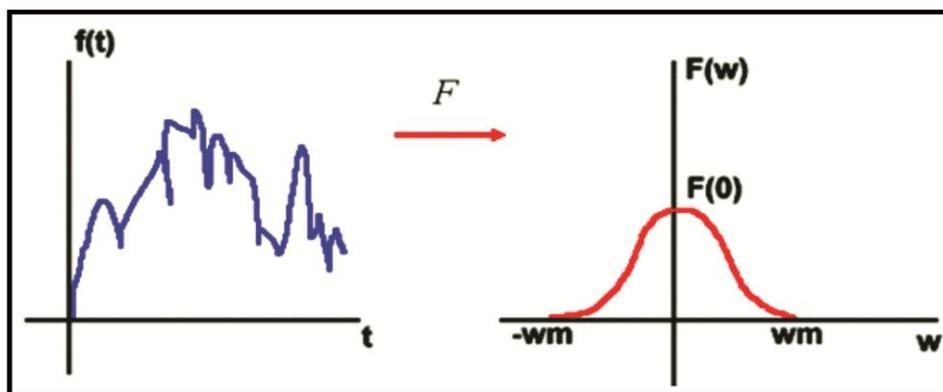
quantização da saída, a linearidade e a necessidade de circuitos *sample-and-hold* (ALEXANDRE, 2013).

Na conversão A/D é necessário colher um número discreto de amostras de um sinal contínuo para que o problema crucial contido no número de amostras por segundo possa ser colhido na amostragem. Entretanto, um pequeno número de amostras pode resultar em uma representação demasiadamente pobre do sinal. A análise quantitativa acerca desse problema é estudada pelo teorema de Nyquist-Shannon. (MALHEIROS, 2011).

A princípio, pode-se imaginar que no processo de amostragem de um sinal analógico há sempre perda de informação e que essa perda é tanto menor quanto maior a taxa de amostragem utilizada. Entretanto, o teorema de Shannon (ALEXANDRE, 2013), mostra que isto nem sempre é verdade, pois o mesmo estabelece que sob certas condições, as amostras de um sinal podem conter, precisamente, toda a informação a ele associada. Isto significa que o sinal pode ser perfeitamente recuperado a partir de amostras colhidas sem nenhuma aproximação.

O estudo sobre o teorema da amostragem é aplicado a sinais, sendo que aqueles que não possuem componentes espectrais em sua amostragem passam para uma frequência acima, como no exemplo da figura 11.

Figura 11 – Exemplo de um sinal banda limitada.



Fonte: ALEXANDRE, 2013.

Embora essa condição seja verificada de outras formas, ela é útil em termos práticos para a implantação de sistemas que utilizem a captura de muitos dados de amostragem para processamento dos mesmos.

## 2.4.2 TEOREMA DE NYQUIST-SHANNON

O teorema da amostragem de Nyquist-Shannon é fundamental para a área de processamento de sinais, em que o sinal tem como característica uma banda limitada por  $F_m$  Hz, e está unicamente determinado por amostras e são tomadas, pelo menos,  $2.f_m$  amostras equidistantes por segundo. A prova disso são as amostras obtidas a cada  $T_s$  segundos, considera-se então um trem de impulsos.

Segundo Alexandre (2013), amostrar é o processo no qual se converte um sinal em uma sequência numérica. Sendo assim, uma frequência de amostragem de um sinal analógico, para que possa, posteriormente, ser reconstituída a informação, deve ser igual ou maior a duas vezes a maior frequência do espectro desse sinal  $B < \frac{f_s}{2}$ . Ou seja, a condição suficiente para uma reconstrução a partir das amostras em uma taxa de amostragem  $f_s > 2B$ , em que  $2B$  é chamado de Taxa de Nyquist e é uma propriedade do sinal, enquanto que  $f_s/2$  é chamado de Frequência de Nyquist sendo uma propriedade do sistema de amostragem.

O intervalo de tempo entre sucessivas amostras é chamado de intervalo de amostragem  $T \stackrel{\text{def}}{=} \frac{1}{f_s}$ , e as amostras de  $x(t)$  são:  $x(nT)$   $n \in \mathbb{Z}$  (inteiros). O teorema da amostragem nos conduz a um procedimento para a reconstrução do  $x(t)$  original a partir de amostras e, respeitando-se as condições iniciais, tornando-a, assim, possível (MALHEIROS, 2011).

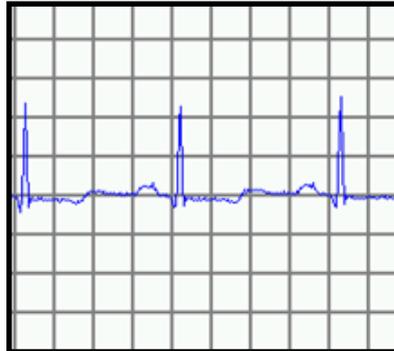
Essa reconstrução do sinal original é um processo de interpolação que matematicamente define um sinal contínuo no tempo  $x(t)$ , a partir de amostras discretas  $x[n]$  e às vezes, entre os instantes de amostragem  $nT$ .

## 2.4.3 CARACTERÍSTICAS DOS SINAIS

Foram escolhidos inicialmente sinais para serem visualizados: *SPIKE*, *ECG* e *EMG*. O analisador de sinais eletrofisiológicos (ASE) foi criado de forma a possibilitar a inclusão de outros sinais. Logo, sua interface consiste em uma tela inicial na qual o usuário pode, facilmente, adaptar e visualizar outros sinais. Em todos os casos, o sinal é deslizado pela tela, da esquerda para a direita.

O sinal visualizado pelo software Medplot no Departamento de Medicina da Universidade de Brasília (ORTIS, R. S. et al, 2013) corresponde a dois (2) segundos do *ECG* no modo retrato e está ilustrado, para uma melhor compreensão, na figura 12.

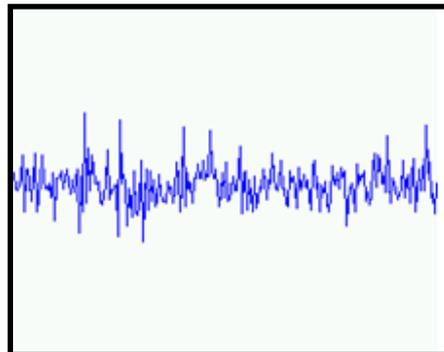
Figura 12 – Plot do ECG.



Fonte: ORTIS, R. S. et al, 2013.

O sinal do EMG é trabalhado de maneira que o usuário visualize o mesmo de forma contínua e seja mostrado na tela. Para tanto, pode ser necessário uma atenção e conhecimento específico na área da Medicina Biomédica para realizar experimentos com o sinal. A figura 13 mostra a captura, na tela, do sinal de EMG.

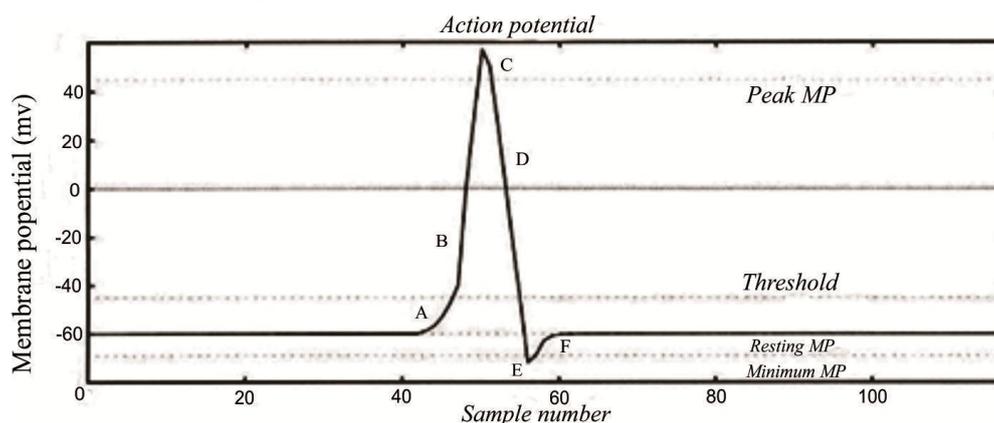
Figura 13 – Plot do EMG.



Fonte: ORTIS, R. S. et al, 2013.

Os *spikes* têm como principal característica o disparo do neurônio formando, assim, uma saída de potencial de ação, tais como: média temporal dos disparos, intervalos entre pulsos, sequência dos disparos e forma de onda específica, como ilustra a figura 14.

Figura 14: Potencial de ação SPIKE de um neurônio.



Fonte: DAMPER et al.

A forma de onda mostrada na figura 14 define o estado do potencial da membrana em relação a um dos valores fixos do neurônio, conforme apresentado na tabela 1. Estes valores são baseados nos potenciais de membrana de um neurônio biológico do tipo motor.

Tabela 1. Parâmetros fixos em relação ao potencial de membrana do neurônio.

Parâmetro	Valor em mV
Potencial mínimo (E)	-69
Potencial de repouso (F)	-60
Limiar de disparo (Entre A e B)	-45
Potencial de pico (C)	45

Os estados apresentados da forma de onda do *spike* são definidos como:

- **Estado A:** Potencial da membrana se encontra acima do potencial de repouso e abaixo do limiar.
- **Estado B:** Potencial da membrana se encontra acima do limiar e abaixo do pico.
- **Estado C:** Potencial da membrana se encontra no pico.
- **Estado D:** Potencial da membrana está no pós-disparo.
- **Estado E:** Potencial da membrana se encontra no mínimo.

- **Estado F:** Potencial da membrana em repouso.

Foram realizadas pesquisas com os sinais citados anteriormente, que serviram de base para o desenvolvimento de um *software* capaz de capturar, visualizar e processar esses sinais.

O desenvolvimento do *software*, acima citado, e suas especificações serão abordadas de forma mais completa no próximo capítulo.

### 3. METODOLOGIA

Análise de Sinais Eletrofisiológicos (ASE), é um sistema específico para captura, armazenamento, processamento e visualização de sinais eletrofisiológicos. Este *software* foi desenvolvido em conjunto com o kit didático, mas não será obrigatoriamente vinculado a este kit.

O ASE será utilizado principalmente em experimentos utilizando insetos. Os dados capturados pelo sistema servirão como material de estudo para o público alvo do projeto - alunos, professores e pesquisadores da rede pública de ensino.

Para desenvolver o *software* didático, de fácil utilização, multiplataforma e com um alto desempenho que repassasse informações específicas de forma intuitiva para os usuários, foi utilizado o paradigma de Programação Orientada a Objeto e Linguagem Unificada de Modelagem (UML), para definir e modelar suas classes.

Seguindo esse raciocínio de criar um sistema multiplataforma com uma interface intuitiva robusta sem perda nos dados, confiável e com um alto desempenho gráfico, foi utilizada a linguagem de programação C++ e as bibliotecas QT e QWT que fornecem funções para capturar, processar e visualizar o sinal. A seguir serão detalhadas as ferramentas e as linguagens utilizadas para um maior entendimento e agilidade no desenvolvimento.

#### 3.1 LINGUAGENS E TECNOLOGIAS UTILIZADAS

Lista das linguagens e ferramentas utilizadas durante o desenvolvimento do *software*:

- **QT CREATOR 5.0.2:** IDE (*integrated development environment*) desenvolvido pela empresa Digia, para desenvolvimento de *software* multiplataforma, é uma estrutura de interface do usuário para os desenvolvedores que usam a linguagem de programação C++, com o auxílio da biblioteca QT e QWT. Ferramenta que ajuda muito durante a construção de projetos de um porte mais elevado, oferecendo recursos *back-end* conectados às aplicações, seja ela para computadores ou sistemas embarcados.

- **C++:** Linguagem de programação multiparadigma com *POO* (programação orientada a objetos), utilizada junto à plataforma *QT* para desenvolvimento do *software*. Linguagem bastante poderosa, com suporte as melhores práticas de programação. Contando com suas bibliotecas e ferramentas de programação como: *containers* (vetores, listas, entre outros).
- **Biblioteca QWT:** Juntamente com seus pacotes para a produção de gráficos dos mais diversos tipos. Contendo componentes da *GUI* e desenho gráfico *2D* fornecendo escala, permitindo a implementação de funções como *zoom* para melhor visualização da interface. Bastante versátil a *QWT* e seus componentes permitem o desenho de curvas complicadas e muitos pontos, com muita facilidade e rapidez.
- **Compilador para Windows 32-bits MinGW 4.7:** É um ambiente de execução completo. Portátil, suportam binários nativos para qualquer sistema operacional Windows 32 ou 64 bits.
- **UML:** Linguagem Unificada de Modelagem auxilia a visualização da comunicação entre classes e os objetos, documentando a estrutura do sistema utilizando diagramas padronizados, junto à notação gráfica.
- **Astha Community:** É uma ferramenta para uso da criação e aprimoramento de UML, *software* muito robusto e eficaz no aperfeiçoamento da documentação, facilita bastante a abordagem de modelagem orientada a objetos, pois utiliza diagramas dinâmicos facilitando a visualização lógica do desenvolvimento completo do sistema.

### 3.2 MODELAGEM

A importância da modelagem dentro do desenvolvimento de um sistema é indiscutível. “A modelagem é a parte central de todas as atividades que levam à implantação de um bom sistema” (JACOBSON 2000, p.15).

Somente com o auxílio da modelagem é possível visualizar e controlar o desenvolvimento do sistema de maneira eficaz, identificando e gerenciando riscos. Partindo deste princípio foi formulada a estrutura do sistema e através dela, foram definidas as técnicas e formas escolhidas com o auxílio da modelagem UML.

A UML é uma linguagem - no sentido de permitir a comunicação - para visualização, especialização, construção e documentação de um sistema de informações (FRANCE, 2001), para captura, processamento e visualização de sinais eletrofisiológicos. Foi utilizando o paradigma de Programação Orientada a Objeto com UML para definir e modelar suas classes como são mostrados nos tópicos seguintes.

### **3.2.1 MÁQUINA DE ESTADOS**

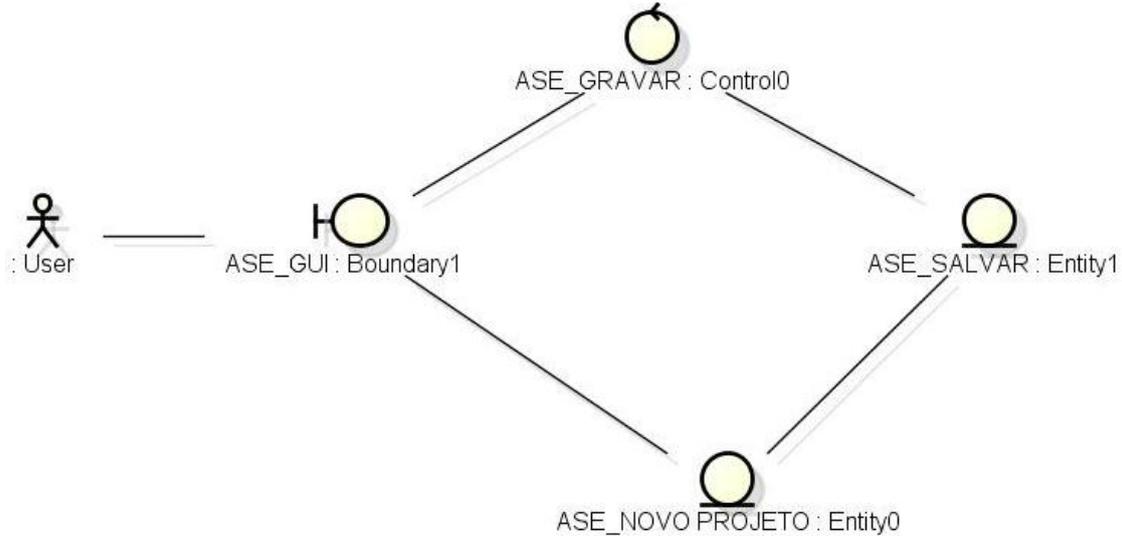
Uma máquina de estados representa a sequência de estados que um objeto assume durante o seu tempo de vida, em resposta a eventos. Desta forma, uma máquina de estados modela o comportamento de um objeto individual.

O processo completo de visualização de estados do sistema foi modelado como uma máquina de estados finitos, como mostra a figura 15. O estado principal, denominado de *ASE\_GUI*, possui relações com outros estados ortogonais. Estados esses que podem ser caracterizados de forma paralela.

O uso de uma máquina de estados para mostrar o funcionamento do sistema possibilitou, entre outros:

- Identificar e dividir os pontos de entrada e saída de cada estado.
- A modelagem da estrutura dos dados que são trocadas entre subsistemas do sistema de controle. Por isso, é de fundamental importância identificar as ações para que se possa entender melhor como os dados fluem no sistema e influencia na captura e visualização dos dados.

Figura 15: Máquina de estados do sistema.



Fonte: Elaborada pelo autor.

As propriedades de cada estado definem o comportamento do mesmo em resposta aos eventos. Estes estão programados para aceitar esta restrição de aceitação e impor regras que minimizem os erros de fluxo dos dados, isto é, as circunstâncias nas quais a máquina pode passar de um estado para outro e que ações respostas precisam ser realizadas quando isso ocorrer. A seguir serão detalhados os estados do diagrama mostrados na figura 15:

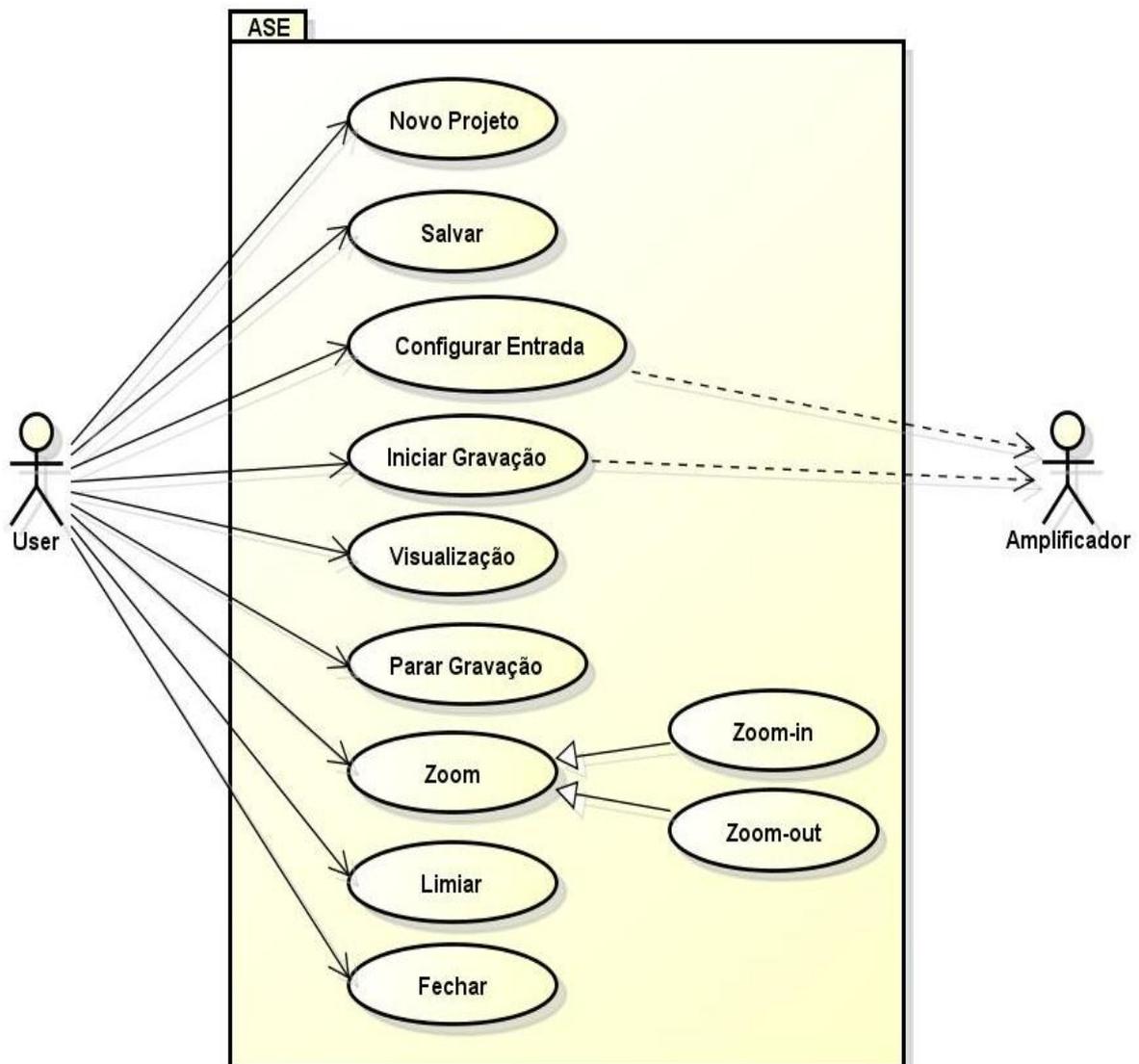
- **User:** (usuário) ator principal que interage com o sistema.
- **ASE\_GUI:** Estado principal que o objeto ou entidade única, representa a interface do sistema onde o usuário interage e visualiza os dados.
- **ASE\_NOVO PROJETO:** Cria um novo projeto para manipulação dos dados.
- **ASE\_SALVAR:** Armazena os dados e cria um arquivo com um diretório para salvar os dados do projeto.
- **ASE\_GRAVAR:** Controla a captura dos dados, executando ações que demandam o processamento e visualização, executando a transferência das informações para a interface do sistema.

### 3.2.2 DIAGRAMA DE CASO DE USO

É utilizado para visualização dos casos de uso que especificam a dinâmica de um sistema, um subsistema ou até mesmo uma classe (LARMAN, 2000; FOWLER, 2004), possibilitando a comunicação entre analistas, desenvolvedores e usuários finais. Um diagrama de casos de uso mostra casos de usos, atores, e os relacionamentos entre seus diversos elementos.

O diagrama de caso de uso foi utilizado para estruturar o comportamento do sistema, além de envolver a interação do ator (usuário) com o *software*. Pode-se ver isso, mais claramente, na figura 16.

Figura 16: Diagrama de caso de uso do sistema.



Fonte: Elaborada pelo autor.

Descrição dos casos de uso:

- **User:** Usuário.
- **Amplificador:** Placa Amplificadora do sinal.
- **Novo Projeto:** Este caso de uso é inicializado quando o usuário pressiona um botão correspondente a Novo Projeto nomeando o projeto, executando assim a criação de um novo projeto.
- **Salvar:** Esse caso de uso o usuário escolhe um diretório (pasta) para salvar os arquivos do projeto.
- **Configurar Entrada:** É iniciado quando o usuário pressiona o botão correspondente a configuração de entrada da placa de áudio do computador, tendo assim a opção de escolher qual entrada vai utilizar para entrada dos dados.
- **Iniciar Gravação:** Sua inicialização ocorre quando o Usuário pressiona o botão Iniciar Gravação, logo será iniciado a captura dos dados oriundos do amplificador e armazena na memória (*RAM*) do computador.
- **Visualização:** Neste caso de uso acontece a visualização das informações capturadas no estado anterior, onde é possível mostrar graficamente um determinado sinal no tempo.
- **Parar Gravação:** Acontece quando o usuário esta visualizando as informações e pressiona o botão Parar Gravação, ocorrendo parada da captura dos dados.
- **Zoom:** Pode ser uma opção realizada no ato da execução do programa, ou após a parada da captura dos dados.
- **Zoom-in:** Este caso de uso é inicializado quando o usuário pressiona o botão esquerdo do mouse.

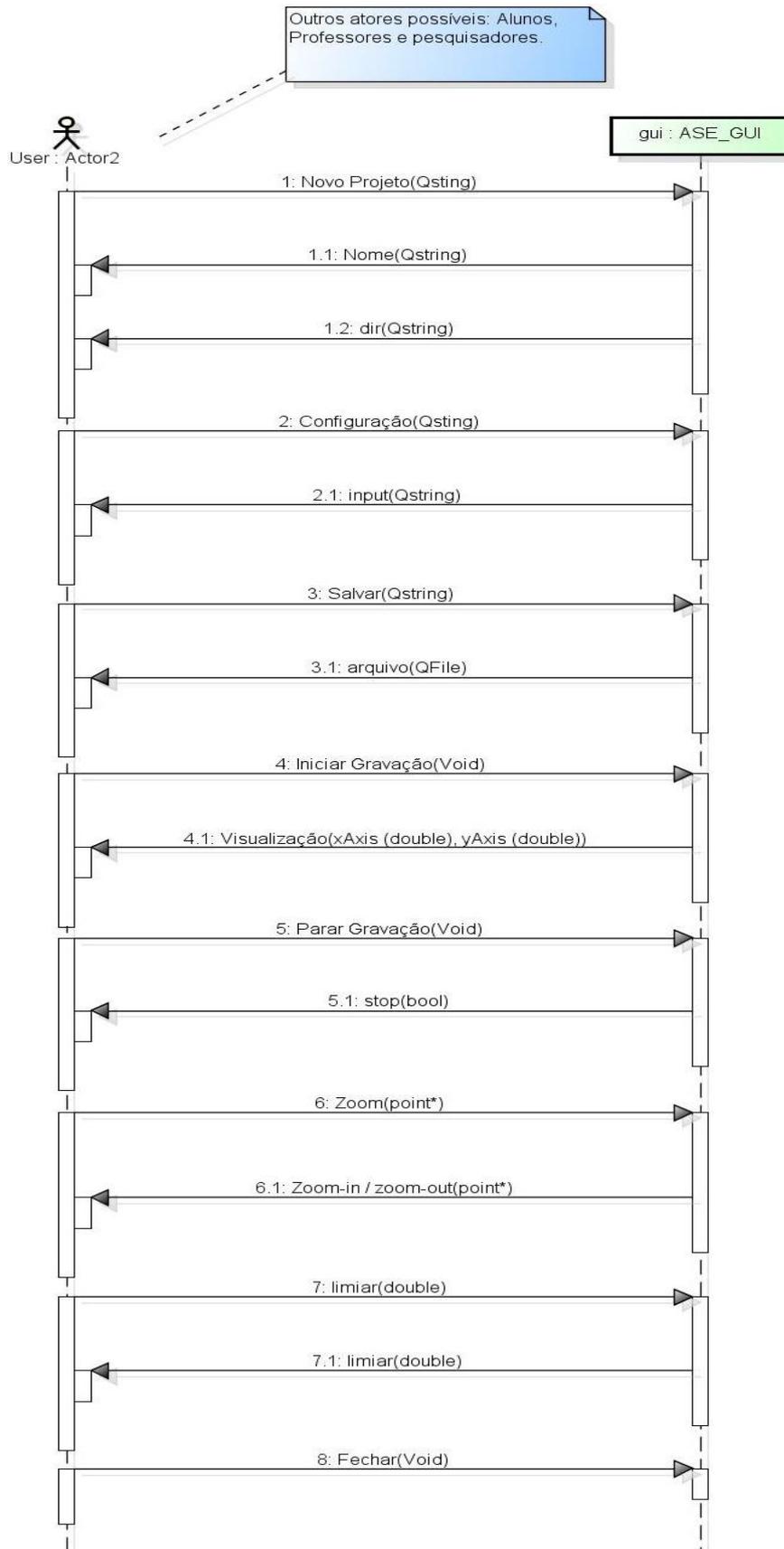
- **Zoom-out:** É inicializado quando o usuário está no estado anterior Zoom-in, e pressiona o botão direito do mouse, voltando a visualizar o sinal em seu estado original no tempo.
- **Limiar:** Neste caso de uso o usuário poderá selecionar um valor de amplitude do sinal (limiar) em uma determinada escala do gráfico para ser usado na detecção e visualização de *spikes*.
- **Fechar:** O caso de uso é inicializado quando o usuário clica na caixa de fechamento do programa, na janela principal do aplicativo.

### 3.2.3 DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência deste sistema permitem, além da representação sequencial das mensagens que trafegam entre os objetos, a representação de mensagens condicionais.

O uso interessante dos diagramas de sequência é uma utilização para representar as interações entre os usuários e o sistema (LARMAN, 2000). Para isso, foram construídos os diagramas de sequência para visualização destas interações mostradas na figura 17.

Figura 17: Diagrama de sequência do usuário.

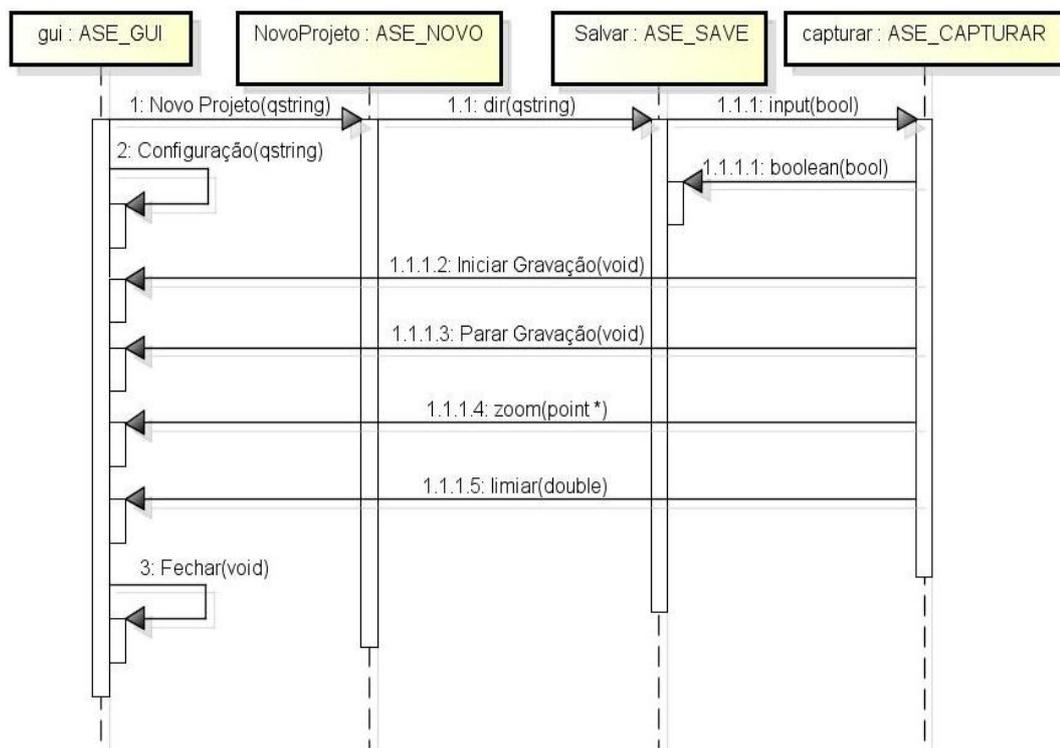


Fonte: Elaborada pelo autor.

A figura 17 mostra o diagrama de sequência do ASE representando a sequência de interações existentes entre o usuário e o sistema, iniciando a criação de um novo projeto. A partir daí o usuário poderá selecionar a placa de áudio do computador que será a porta de entrada dos dados, escolhendo um diretório para salvar os arquivos.

Além destes procedimentos o usuário poderá iniciar uma captura de dados e simultaneamente visualizar as informações. Para a captura e em qualquer destes processos poderá, também, fazer uso da função *Zoom* para aumentar ou diminuir o seu campo de visualização. Utilizando o *limiar* (linha horizontal, ver figura 24) poderá selecionar um determinado ponto do gráfico de detecção de *spikes*, permitindo a visualização dos mesmos. Por fim permite ao usuário sair (fechar) do sistema (figura 17).

Figura 18: Diagrama de sequência do sistema.



Fonte: Elaborada pelo autor.

Nesta figura 18 as mensagens foram colocadas no diagrama conforme sua sequência temporal, informando assim, as interações do sistema com a demanda de funções de uma determinada operação realizada pelo usuário e ao mesmo tempo, focando no diagrama centrado nos comandos e mensagens enviados pelo agente

(usuário) para o sistema, ilustrando, dessa forma, como ele trata as interações ocorridas em cada uma das suas solicitações.

### 3.2.4 DIAGRAMA DE CLASSES

Na UML, os aspectos estruturais são descritos com diagramas de classes, que é a melhor ferramenta oferecida pela UML pra a exibição de um retrato de modelos orientados a objetos, pois descrevem graficamente como as classes e interfaces se relacionam, especificando também as propriedades de cada classe e quais são os métodos oferecidos a implementar respectivamente. (FOWLER, 2004).

Em Engenharia de Software, diagramas de classes apresentam e caracterizam objetos com seus relacionamentos. Este tipo de documentação do software de baixo nível de abstração guia as suas atividades. Para o diagrama de classes deste trabalho, foram definidas três interfaces para implantação: *GUI*, *Novo Projeto e Selecionar diretório*. Interface é um conceito de Orientação a Objetos que define como um componente que irá interagir com outros elementos.

Em termos de funcionalidades, as classes deste sistema foram organizadas seguindo o mesmo argumento proposto nos diagramas dos tópicos anteriores. Os requisitos funcionais foram traduzidos em métodos, utilizados para executar as tarefas dos usuários (figura 19).



Apresentação das classes e seus relacionamentos mostrados na figura 19 e também no apêndice A.

- **mainWindow:** Classe principal do programa, onde o controle do programa inicia e termina, sendo responsável por chamar as classes. É composta por algumas funções como: botões e gravação do áudio.
- **processaSinal:** Classe responsável por receber os pacotes de áudio (buffer) da placa amplificadora, executar o processamento de detecção de *Spikes* dos dois canais (esquerdo e direito), calculando a frequência de disparo e a realização do *downsample* do sinal.
- **file:** Classe responsável por criar um arquivo a cada experimento com os *Spikes* detectados do projeto em XML.
- **dialogNew:** Classe responsável pela janela de novo projeto, onde, nesta janela, o usuário pode definir o nome e diretório onde será salvo o projeto.
- **levelWidget:** Classe responsável por gerar as barras verticais que mostram a intensidade do sinal de cada canal.
- **spikeWidget:** Responsável por conter os eventos de pintura que desenham o *limiar* para selecionar o valor dos *Spikes* a serem detectados pela classe *processaSinal* nos dois canais.
- **contínuoWidget:** Responsável por conter os eventos de pintura que desenham o sinal contínuo dos dois canais (esquerdo e direito), além de conter métodos que trabalham com funções de *zoom* para interagir com o usuário.
- **canvas:** Responsável por gerar as paletas do plano de fundo do gráfico e configurar os *Widgets* da biblioteca *QWT*.

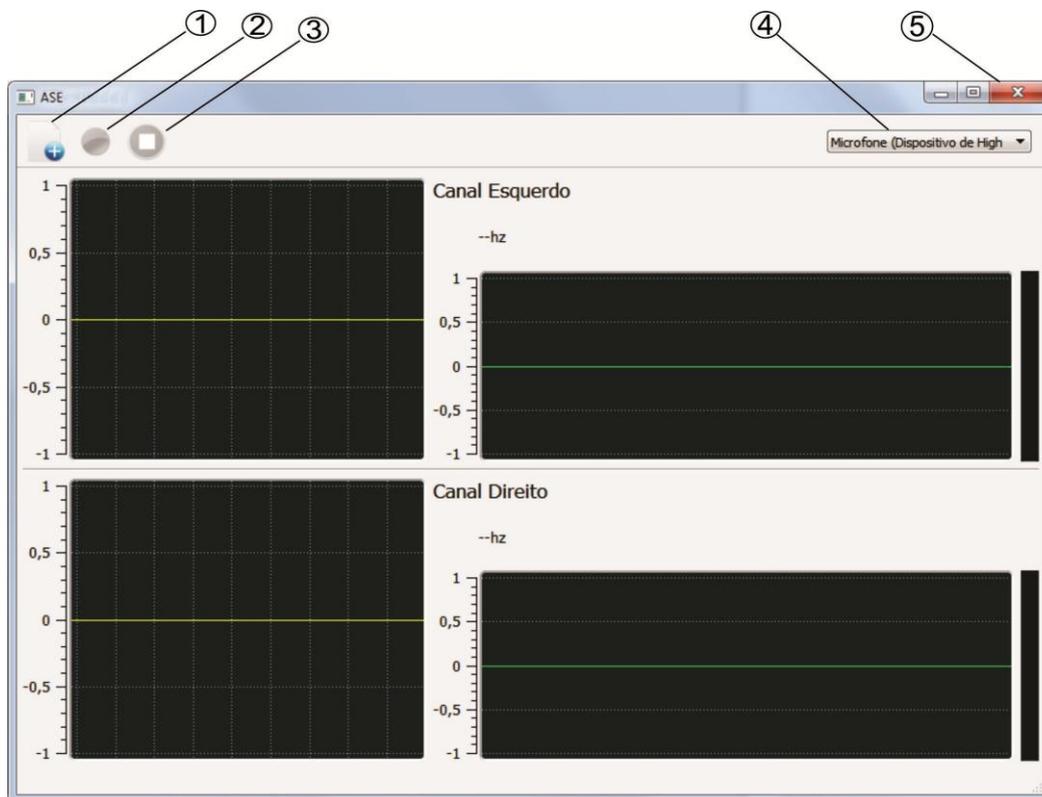
## 4. RESULTADOS

Foram desenvolvidas todas as classes que estruturam a arquitetura proposta do sistema mostrado no capítulo três, onde foi modelado e documentado o *software*, com todos os objetivos gerais, específicos e requisitos exigidos detalhados no primeiro capítulo como a *captura, processamento e visualização de sinais eletrofisiológicos*. Além de funções do sistema: *zoom-in, zoom-out, limiar, detecção de spikes, cálculo da taxa de disparos e salvar os arquivos em XML e WAVE*. Os resultados serão apresentados de acordo com a estrutura modelada obtida em todo o trabalho.

Com a utilização da placa amplificadora é possível estabelecer conexão entre a placa e o computador, realizando uma amplificação do sinal.

Através dessa amplificação, foi desenvolvido um *software (open-source)* didático, de fácil utilização, em português, específico para captura, processamento e visualização de sinais eletrofisiológicos, repassando as informações em uma linguagem científica simples e intuitiva para o público alvo do projeto, como é visualizado na figura 20.

Figura 20: GUI do software ASE.



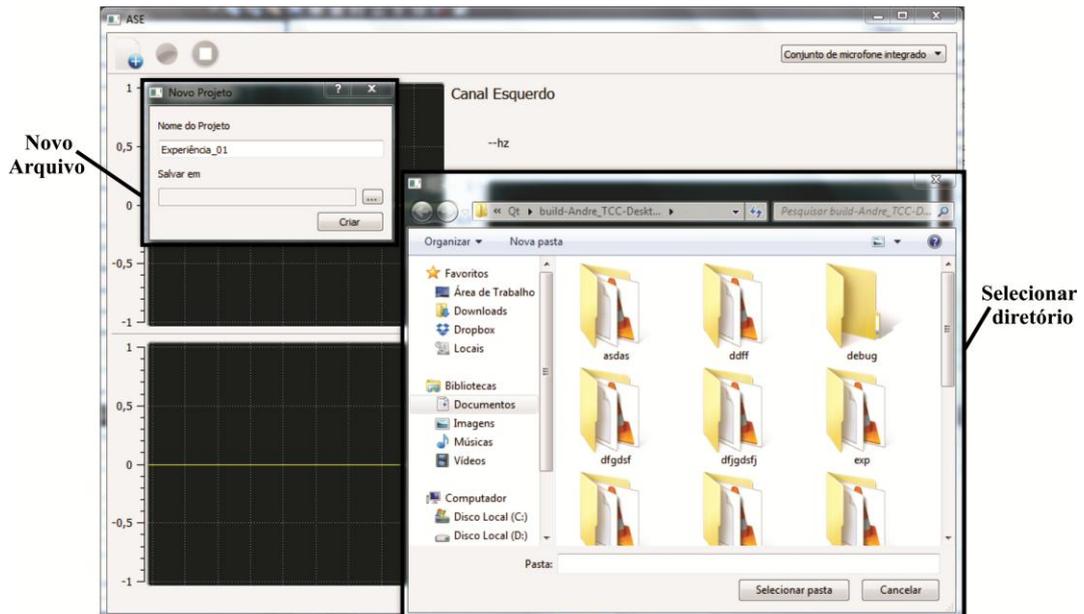
Fonte: Elaborada pelo autor.

A figura 20 mostra a GUI do sistema ASE responsável por gerar a interface gráfica do projeto juntamente com todos os botões enumerados e suas funcionalidades, fornecidas pela barra de ferramenta localizada a esquerda e a direita da janela principal do usuário. A seguir serão detalhadas as funções de cada botão enumerado:

1. **Novo Projeto** : Responsável por criar um novo experimento, nomeando-o e selecionando o diretório (pasta) para armazenar os arquivos capturados do projeto.
2. **Iniciar Gravação** : Utilizado para iniciar a captura e visualização dos dados, ao ser pressionado este botão gera um arquivo WAVE para gravar o áudio contínuo e também dois arquivos XML para salvar os Spikes detectados pelos dois canais.
3. **Parar Gravação** : Utilizado para parar a captura dos dados, este botão finaliza a gravação do áudio, encerrando os arquivos WAVE e XML.
4. **Configuração** : Permite que o usuário selecione a placa de áudio do seu computador que será a entrada de áudio a ser utilizada no experimento.
5. **Fechar** : Permite que o usuário saia do sistema.

Após a inicialização do sistema o usuário poderá interagir com o mesmo, utilizando-se das funções que compõem o *software* através dos botões do sistema, criar um novo projeto e escolher o diretório que irá salvar os dados da experiência científica (ver figura 21).

Figura 21: Janela de Novo Projeto.



Fonte: Elaborada pelo autor.

Ao pressionar o botão Novo Projeto, será gerada uma janela na interface do ASE, ilustrado na figura 21, onde o usuário poderá nomear sua experiência e determinar qual o diretório irá salvar seus dados experimentais.

Os dados gerados pelo sistema serão armazenados em uma pasta em arquivos WAVE que grava o sinal contínuo e dois arquivos XML para os *spikes* detectados nos dois canais (esquerdo e direito) de cada captura.

Além destas funções, mencionadas anteriormente, o usuário pode utilizar o botão configuração ilustrado na figura 20, para gerar lista de dispositivos de entrada do seu computador, ou seja, qual placa de áudio irá utilizar para captura dos dados.

Se o sistema operacional do seu computador estiver configurado, como padrão, a placa de áudio de sua *webcam*, o usuário terá que selecionar como entrada a placa de áudio integrada à placa mãe, onde encontrará a entrada do microfone no qual será conectado o cabo P2 ligado ao amplificador eletrofisiológico.

Logo em seguida, o utilizador do sistema pode efetuar seus experimentos que serão capturados e visualizados através de desenhos gráficos gerados pelos algoritmos implantados no desenvolvimento do sistema.

A seguir será mostrada, na figura 22, a interface do ASE visualizando uma experiência.

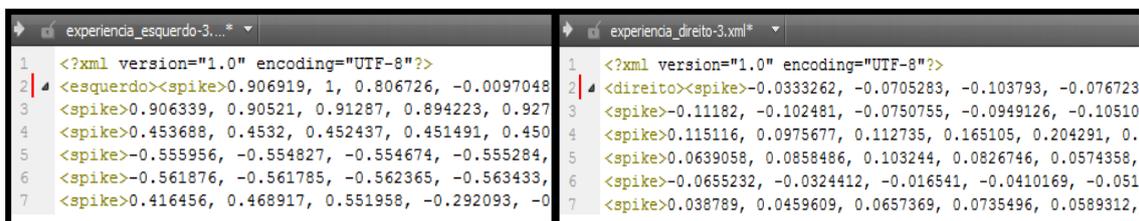
Figura 22: Visualização dos dados na GUI do ASE.



Fonte: Elaborada pelo autor.

A figura 22 mostra a *GUI* do sistema ASE visualizando o sinal contínuo e os *spikes* capturados e processados em dois canais (esquerdo e direito). Nesta interface o usuário pode iniciar a captura e visualização de suas experiências e também optar por parar a captura utilizando o botão Parar Gravação e, posteriormente retornar a captura, guardando os seus *spikes* detectados em arquivos XML separados para cada canal como é visualizado na figura 23.

Figura 23: Visualização dos Spikes gravados em XML dos canais esquerdo e direito.



Fonte: Elaborada pelo autor.

Após a geração do arquivo XML o usuário pode acessar os dados capturados salvos anteriormente, visualizando estas informações em qualquer editor de texto. Estes arquivos XML guardam as informações dos dois canais (esquerdo e direito) em linhas contínuas, pois nelas estão todos os *spikes* detectados pelo *software*.

Além de salvar estes *spikes*, o *software* também armazena o sinal contínuo dos dois canais em um arquivo WAVE de alta qualidade, formato onde o áudio digital é armazenado sem perdas por sequências numéricas convertidas em dados (amostras), que são gravadas bit por bit.

Diante do experimento o *software* fornece funções primordiais para que o usuário tenha uma ampla interação com suas cobaias, pois o sistema é composto por ferramentas complementares que o tornam mais robusto e eficaz em sua visualização.

Para que essa interação aconteça foram desenvolvidas ferramentas de *Zoom-in*, *Zoom-out* para visualizar, e *Limiar* para realizar contagem e detecção de *spikes* nas experiências, mostradas na figura 24.

Figura 24: Funções de Zoom-in, Zoom-out e Limiar do ASE.



Fonte: Elaborada pelo autor.

As funções da ferramenta *Zoom-in* e *Zoom-out* da figura 24 trazem uma maior funcionalidade para os usuários do sistema, sendo possível utilizar tanto no momento da captura quanto na visualização dos dados, pois torna possível a ampliação e diminuição do gráfico mostrado na GUI.

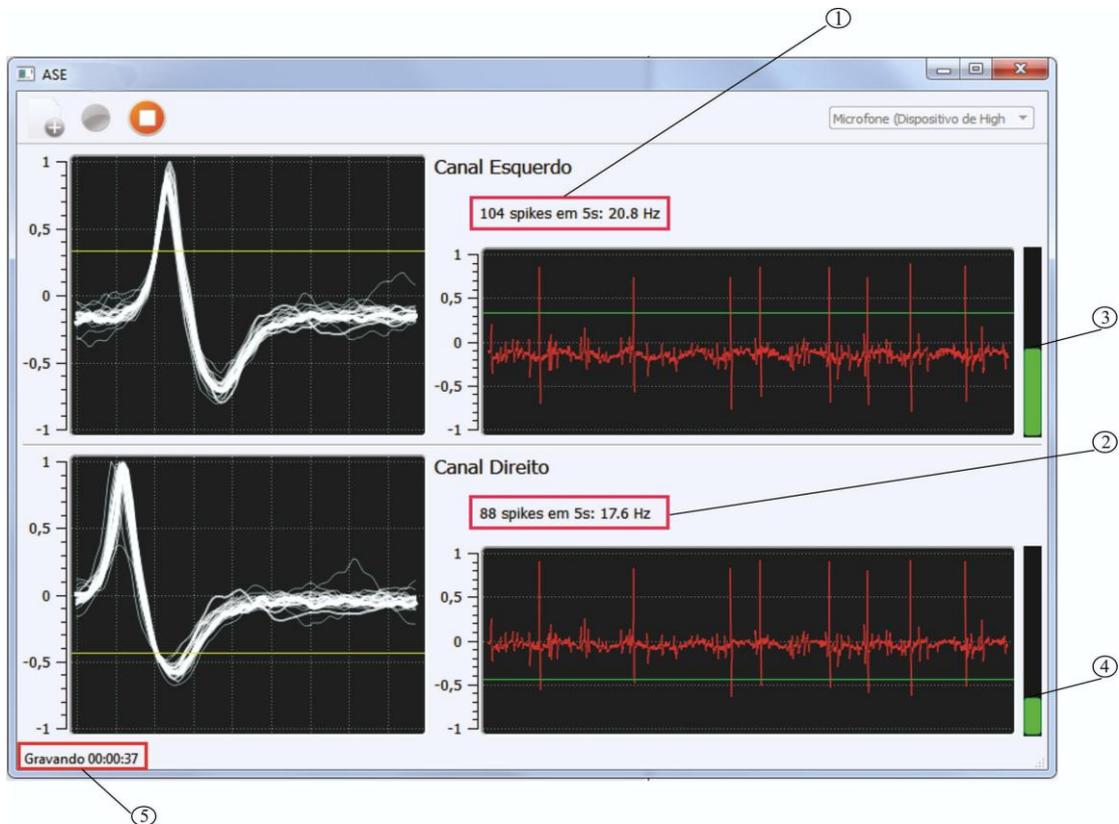
Para que os usuários fossem capazes de realizar a contagem dos *spikes* de um determinado experimento, foi criado um algoritmo que conta os disparos (picos da onda

que ultrapassa a linha horizontal ou *limiar* selecionado) de uma determinada amostra, em um ponto específico definido pelos utilizadores do *software* (figura 24). Essa função é de suma importância, pois torna o sistema mais robusto e com maior funcionalidade e através dela os pesquisadores terão maior flexibilidade e auxílio na detecção dos potenciais de ação, apresentação e detalhamento de um experimento capturado mostrando seus resultados como a detecção de *spikes*, a visualização de sinal contínuo, a taxa de disparo e o temporizador dos dois canais.

A fim de validar os resultados obtidos pelo *software*, foi realizado a captura do sinal, detecção do *SPIKE* e *taxa de disparo*, através de experimentos testes utilizando a captura dos dados da pata de uma barata para validação dos dados capturados. O ASE se mostrou apto a ser utilizado, pois capturou o sinal contínuo e detectou os *SPIKES* determinado pelo *limiar*. Além de salvar os disparos detectados e o sinal contínuo de forma eficaz, também foi possível utilizar da ferramenta de zoom para obter uma melhor visualização do sinal.

A seguir serão detalhados os resultados enumerados na figura 25 que mostra uma experiência com a pata de uma barata:

Figura 25: Frequência de disparos, nível do sinal e Relógio.



Fonte: Elaborada pelo autor.

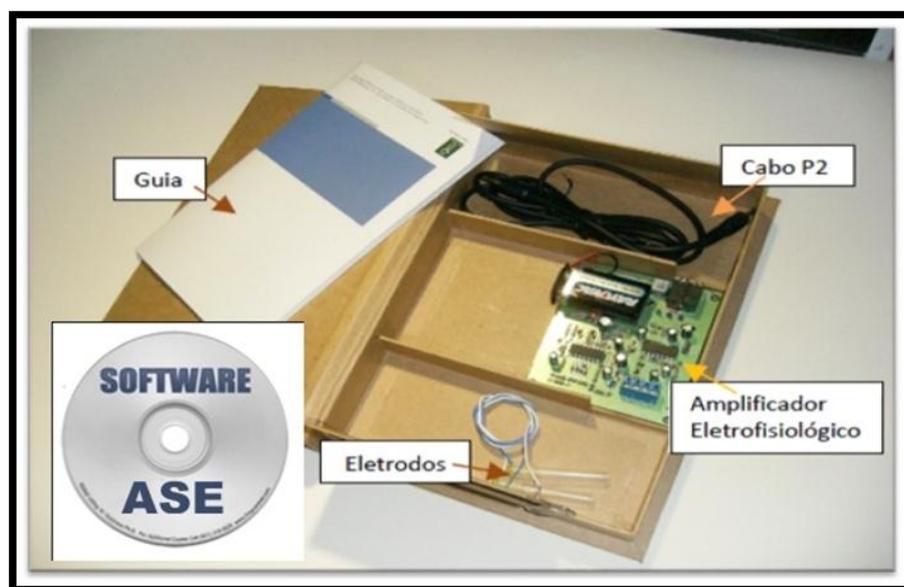
1. Quantidade de *spikes* detectados pelo canal esquerdo e sua frequência de disparos em hertz.
2. Quantidade de *spikes* detectados pelo canal esquerdo e sua frequência de disparos em hertz.
3. Nível do sinal contínuo do canal esquerdo.
4. Nível do sinal contínuo do canal direito.
5. Relógio temporizador.

A taxa de disparo dos canais esquerdo e direito é calculada diferente para cada canal de acordo com a sua amostragem, pois estes capturam dados diferentes pelo fato da captura do áudio ser Estéreo. Logo seus comportamentos serão distintos, consequentemente seus *spikes* detectados e sua taxa de disparos também.

#### 4.1 PRODUTO FINAL

O objetivo deste trabalho foi desenvolver um *software* didático e intuitivo para captura, processamento e visualização de sinais eletrofisiológicos em insetos. O desenvolvimento deste *software* complementou o kit didático que é composto por um sistema de amplificação de sinais biológicos - hardware - eletrodos, cabo P2 e manual como ilustrado da figura 26.

Figura 26: Produto Final Kit Didático.



Fonte: Elaborada pelo autor.

Com o término do desenvolvimento do sistema foi possível ter um produto final, como ilustra a figura 26, que tem como principal função registrar os potenciais de ação bioelétricos dos neurônios de insetos.

Estes registros servirão como material de estudo para os usuários do kit didático, contribuindo com o desenvolvimento educacional dos alunos do Ensino Médio, da rede pública, por meio da oferta prática de oficinas e aulas vinculadas ao estudo do sistema nervoso. A apresentação deste método de ensino científico se associa aos conceitos transmitidos em sala de aula recomendados pelas diretrizes dos Parâmetros Curriculares Nacionais (PCN's).

Essas atividades práticas possibilitarão ampliar o conhecimento científico individual de alunos e professores ajudando a entender os fenômenos e enigmas envolvendo o cérebro, e a produção de novos saberes na área da neurociência. Esta produção estimulará o desenvolvimento de novas tecnologias e produtos, que por sua vez, possibilitarão novas descobertas científicas.

A prática pedagógica nas aulas de Biologia poderá ser mais dinâmica, através de experimentos realizados em sala, uma vez que este kit didático favorecerá a troca de experiências entre os alunos durante a prática. Este material didático, em conjunto com os saberes escolares, pode contribuir na experiência cotidiana do ensino-aprendizagem, permitindo que os saberes construídos em sala de aula, articulados a vivência do aluno e aos conteúdos didáticos, criem melhores oportunidades de compreensão dos conceitos científicos e conteúdos pedagógicos, possibilitando assim, a construção de novos saberes por meio do diálogo constante entre a teoria e a prática.

Neste raciocínio, torna-se importante pensar na introdução do kit didático na Educação, em especial no Ensino Médio, de modo que ele possa melhorar a relação ensino-aprendizagem, no qual o educando possa deixar de ser apenas um mero expectador para tornar-se um sujeito ativo do/no processo de ensino- aprendizagem, pois permitirá ao professor interagir com este em sala de aula, transformando o ambiente de aprendizagem tradicional num espaço motivador de ensino e gerador de experiências inovadoras valiosas.

Acredita-se que a utilização deste kit didático, além de construir o conhecimento científico, terá uma grande importância para a educação dos jovens brasileiros, pois promoverá um programa de inclusão do conhecimento do Ensino Médio, reciclando os professores das áreas de Física, Matemática e Biologia dentro da interdisciplinaridade que caracteriza a Neurociência.

Este produto final terá um grande impacto na formação de recursos humanos, pois trará conhecimento a novos membros da sociedade e servirá como ponte de aproximação entre duas Instituições de Educação: a Universidade e a Escola de Ensino Fundamental, da rede pública.

Vale ressaltar, também, que essas atividades práticas no dia-a-dia dos alunos contribuem de maneira decisiva para a motivação e a melhoria do aprendizado.

Por tanto, é de suma importância relatar o desafio proposto pelo kit didático, não somente pela tentativa de fazer Ciência fora do ambiente universitário, mas, também pela grande resistência encontrada por professores em ministrar aulas práticas, devido à falta de infra-estrutura de laboratórios e dificuldade em elaborar um material interessante com qualidade que possa tornar o espaço escolar mais prazeroso e propício a construção do conhecimento científico.

## 5. CONSIDERAÇÕES FINAIS

Neste trabalho, em atendimento ao seu objetivo principal, foi desenvolvido um *software* livre e robusto tendo como principais funcionalidades a captura, análise e processamento de sinais, tornando uma ferramenta intuitiva para que os usuários utilizem em suas pesquisas, facilitando a visualização de fenômenos do sistema nervoso dos seres vivos como o *spike*. É importante ressaltar, entretanto, que a área de processamento de sinais biomédicos é imensa, e para se obter um domínio mínimo da mesma é necessária, em geral, a participação de especialistas na área e pesquisas complexas relacionadas ao tema.

Grande parte das principais aplicações que envolvem as modernas tecnologias empregadas e técnicas de processamento, de alguma forma, permite o entendimento dos princípios básicos dessas tecnologias que são tão importantes para se ter uma noção das características dos equipamentos e de suas limitações. A partir deste conhecimento obtido, os utilizadores poderão tornar-se independentes, passando a realizar seus próprios experimentos.

As pesquisas obtidas, difundidas nas práticas experimentais em eletrofisiologia, promovem as orientações apropriadas à melhoria na qualidade do ensino de alunos do Ensino Médio das escolas públicas, estabelecendo as diretrizes para direcionar o conhecimento a uma população que não usufrui de uma estrutura tecnológica ideal para o aprendizado.

O presente trabalho proporcionará aos professores métodos de ensino lúdico, tornando as aulas mais agradáveis, como também facilitará a aprendizagem dos alunos utilizadores do sistema, que por sua vez foi desenvolvido por algoritmos compostos por seus atributos e métodos. As técnicas utilizadas neste trabalho se mostraram bastante satisfatórias, pois tornaram possíveis a captura, processamento e visualização de sinais eletrofisiológicos, além da construção de ferramentas como *Zoom* e detecção de *spikes* visualizando estes potenciais de ação.

Finalmente, é importante ressaltar a necessidade de um treinamento prévio do usuário na utilização do *software* desenvolvido, e um conhecimento básico na área de Neurociência para que possa ter um maior grau de entendimento e aprendizagem.

## 6. TRABALHOS FUTUROS

No processo de desenvolvimento desse trabalho pode-se concluir que algumas áreas da Neurociência ainda são muito deficitárias para o desenvolvimento real de plataformas móveis. Por isso optou-se por começar pela definição de uma arquitetura pra fins de viabilidade tecnológica. Ainda foi sentida a necessidade de estudos nas áreas de educação sobre o conceito para estimular à adesão a mesma, a definição, adaptação ou estudo de linguagens padrões para tratar da construção do sistema. Recomenda-se que sejam realizadas novas avaliações deste meio de aplicação, para que além da realização de melhorias contínuas, seja possível realizar comparativos entre aplicações futuras e existentes no mercado. Considerando que o projeto está em sua primeira versão, acredita-se que por meio da avaliação seja possível um refinamento, permitindo que este esteja em processo de constante melhoria. Com a utilização da biblioteca *QT* em seu desenvolvimento, sendo ela apta a construção de *softwares* para dispositivos móveis, geridos pelo sistema operacional *Android*, uma sugestão para o futuro seria uma migração e adaptação do *software* para estes dispositivos, tendo em vista o grande avanço e disseminação dos mesmos, ocasionando na popularização e divulgação do sistema móvel, tornando capaz a realização de pesquisas realizadas por pessoas de diversas classes sociais, vivenciando experiências em ambientes surpreendentes e inimagináveis.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALEXANDRE, Carlos. **Processamento Digital de Sinais**. Centro de Informática, UFPE. Recife, p.1 – p.331, 2013.

AUDACITY, **Audacity**. Disponível em: <<http://audacity.sourceforge.net/?lang=ptBR>>. Acesso em: 09 de março de 2013, 16:20.

BASS, Bianca Laufer. **Plasticidade Neural**. Disponível em: <<http://www.profala.com/artfisi36.htm>>. Acesso em: 18 de outubro de 2014.

C++, C. E. **C & C++ Brasil**. Disponível em: < <http://www.ccppbrasil.org/wiki/Main> >. Acesso em: 13 de maio de 2013, 13:15.

CPLUSPLUS. **Other Data Types - C++ Documentation**. Disponível em: < [http://www.cplusplus.com/doc/tutorial/other\\_data\\_types/](http://www.cplusplus.com/doc/tutorial/other_data_types/) >. Acesso em: 15 de maio de 2013.

DAMPER, Robert et al. **An autonomous robot based on inspirations from biology**. Robotics and Autonomous Systems, v. 31 n. 4, p. 247 – 274, 2000.

DIGIA. **Qt**. Disponível em: < <http://qt.digia.com/> >. Acesso em: 13 de maio de 2013, 9:00.

ERHART, Eros Abrantes. **Neuranatomia**. 5. ed. São Paulo: Atheneu, p. 1- p. 420, 1974.

FEMEDA, Edberto. **Redes Neurais e sua aplicação em sistemas de recuperação de informação**. Ciência da Informação, Brasília, v. 35, n. 1, p.25-30, jan/abr. 2006.

FRANCE, R.: KOBRYN, C. **UML for Software Engineer**. In: Internacional Conference on Software Engineering, 23, 2001, Toronto. Proceedings. Toronto: IEEE Computer Society Press. P. 705 – 706, 2001.

FREIRE, P. **Pedagogia da autonomia-Ed**. Paz e Terra-São Paulo, 1997.

**GAPMINDER: Unveiling the beauty of statistics for a fact based world view.** Disponível em: < <http://www.gapminder.org/> >. Acesso em: 08 de março de 2013, 11:20.

**GCC. GCC, the GNU Compiler Collection** - GNU Project - Free Software Foundation (FSF). Disponível em: < <http://gcc.gnu.org/> >. Acesso em: 16 de maio de 2013, 18:40.

**GEPHI, an open source graph visualization and manipulation software.** Disponível em: < <https://gephi.org/> >. Acesso em: 06 de março de 2013, 22:40.

**GOLDSTEIN K. PAUL BROCA (1824-1880). In Haymaker W. The founders of neurology.** Springfield: Charles C. Thomas, 1953:259-263.

**ISO. ISO/IEC JTC1/SC22/WG21 - The C++ Standards Committee.** Disponível em: < <http://www.open-std.org/jtc1/sc22/wg21/> >. Acesso em: 15 de maio de 2013, 23:10.

JACOBSON, Ivar et al. **UML Guia do Usuário.** Rio de Janeiro: Campus, p. 472, 2000.

KRANTS, John. H. Hanover College. **Cognition Laboratory Experiments.** Disponível em: < <http://psych.hanover.edu/JavaTest/CLE/Cognition/Cognition.html> >. Acesso em: 15 de abril de 2014, 8:35.

LARMAN, G. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientado a objetos.** Porto Alegre: Bookman, 2000.

LEÃO, Aristides, **Ciência Hoje.** Disponível em: <<http://cienciahoje.uol.com.br/controlPanel/materia/view/730>> Acesso em: 18 de janeiro de 2014, 21:45.

LEITE S. D., **As atividades laboratoriais e a avaliação das aprendizagens dos alunos.** Trabalho prático e experimental na Educação em Ciências, p. 91-108, 2000.

MALHEIROS, R. **Detector de campos eletromagnéticos de frequência ultra baixas utilizando a plataforma arduino duemilanove.** Engenharia da computação. Universidade São Francisco. Itatiba p.1 – p.57, 2011.

MILLER, S. **Puckette's "vanilla" distribution of Pd.** Disponível em: <<http://puredata.info/downloads/pure-data>>. Acesso em: 02 de fevereiro de 2014, 10:22.

NEUROENGENHARIA: **Neuroengenharia UFRN**. Disponível em: <<https://sites.google.com/site/neuroengenharia20112ufrn/home/kit-neuroengenharia>> Acesso em: 15 de junho de 2013, 8:00.

NETMAGAZINE. **Netmagazine**. The top 20 data visualisation tools | Feature | .net magazine. Disponível em: <<http://www.netmagazine.com/features/top-20-data-visualisation-tools> >. Acesso em: 13 de março de 2013.

NEUROMED, Fernando Pontes. **Neuromed**. Disponível em: <[http://neuromed94.blogspot.com.br/2011\\_12\\_01\\_archive.html](http://neuromed94.blogspot.com.br/2011_12_01_archive.html)>. Acesso em: 11 de março de 2013, 20:10.

ORTIS, R. S. et al. **Monitoração de Sinais Biomédicos em Assistentes Pessoais Digitais**. Universidade de Brasília. Brasília, n.4, p.1-6, 2013.

PEGEA. **Pós-Graduação em Eng. de Sistemas Eletrônicos e de Automação / Dep. de Eng. Elétrica / UnB - Principal**. Disponível em: < <http://www.pgea.unb.br/> >. Acesso em: 14 de março de 2013, 15:50.

PURVES, Dale et al. **Neuroscience**. 3. ed. Massachusetts: Sinauer Associates Inc, 773p, 2004.

QT BLOG. **QT**. Disponível em: < <http://blog.qt.digia.com/> >. Acesso em: 28 de maio de 2013, 14:00.

RAMON Y CAJAL, Santiago. **Neuronismo o Reticularismo. Las pruebas objetivas de La unidad anatômica de lãs células nerviosas** [1933]. Madrid: Intituto Cajal, 1952.

RESERARCHGATE. **Otimização de desempenho de algoritmos de compressão de sinais biológicos utilizando redes neurais artificiais**. Disponível em: < [http://www.researchgate.net/publication/38976716\\_Otimizacao\\_de\\_desempenho\\_de\\_algoritmos\\_de\\_compresso\\_de\\_sinais\\_biologicos\\_utilizando\\_redes\\_neurais\\_artificiais](http://www.researchgate.net/publication/38976716_Otimizacao_de_desempenho_de_algoritmos_de_compresso_de_sinais_biologicos_utilizando_redes_neurais_artificiais) >. Acesso em: 15 de março de 2013, 16:10.

REVISTA NEUROCIÊNCIAS. **Revista neurociências**. Disponível em: < <http://www.revistaneurociencias.com.br/> >. Acesso em: 10 de março de 2013.

REZENDE, Magali et al. **Categorização e Classificação de documentos a partir de suas citações: Uma proposta baseada em Redes Neurais Artificiais**. Revista de Informação – Minas Gerais v.12 n.5, outubro de 2011.

SILVA, Maurício Marx et al. **A consciência: algumas concepções atuais sobre sua natureza, função e base neuroanatômica.** Rio Grande do Sul, 25 (suplemento 1), p. 52-64, abril 2003.

SOUZA, Gisele. **Um Simulador Neural para Visualização de Aspectos de Aprendizado utilizando Neurônios Spiking.** Centro de Informática, UFSC. Florianópolis, p.1-109, 2005.

**SPECTROGRAMS, SPECTROGRAPHS AND SPECTROGRAM SOFTWARE.**  
Disponível em: < <http://www.visualizationsoftware.com/spectrogram> >. Acesso em: 07 de março de 2013, 13:40.

UFSM. **UFSM.** Disponível em: <<http://www-usr.inf.ufsm.br/~rose/curso3/cafe/XML-Cap1-Linguagem.pdf>>. Acesso em: 15 de junho de 2013, 14:30.

UNB. **UNB. Departamento de Engenharia Elétrica.** Disponível em: < <http://www.ene.unb.br/> >. Acesso em: 09 de março de 2013, 16:00.

## APÊNDICE A – Detalhamento do Diagrama de Classes

Apresentação das classes em que seus requisitos funcionais foram traduzidos em métodos, utilizados para realizar chamadas e executar as tarefas do usuário.

Detalhamento das classes com seus principais atributos e métodos ilustrados na figura 19 (ver página 41):

- **mainWindow:** Classe principal do programa no qual o controle do mesmo inicia e termina, sendo responsável por chamar as classes. É composta por algumas funções como: botões, gravação do áudio, dentre outras.

Atributos:

- ui: Objeto que gera o *layout* (interface) para o usuário.
- arquivo\_e: Gerencia os arquivos do canal esquerdo.
- arquivo\_d: Gerencia os arquivos do canal direito.
- player: Objeto que gera o *player*.
- gravador: Objeto responsável por gravar o áudio.
- escuta: Objeto responsável por monitorar o áudio.
- theadDireito: Responsável por processar o sinal do canal direito.
- theadEsquerdo: Responsável por processar o sinal do canal esquerdo.
- spikeEsquerda: Gera o gráfico de *Spike* do canal esquerdo.
- spikeDireita: Gera o gráfico de *Spike* do canal direito.
- continuoEsquerda: Gera o gráfico do sinal contínuo do canal esquerdo.
- continuoDireita: Gera o gráfico do sinal contínuo do canal direito.
- sampleLeft: Responsável por guardar as amostras do canal esquerdo.
- sampleRigth: Responsável por guardar as amostras do canal direito.
- limiarE: Linha de marcação do *Spike* canal esquerdo.
- limiarD: Linha de marcação do *Spike* canal direito.
- contadorExp: Responsável por contar os *Spikes*.

#### Métodos:

MainWindow: Construtor da classe MainWindow.  
~MainWindow: Destrutor da classe MainWindow.  
program\_estado: Define o estado da aplicação, ativa e desativa os botões.  
slot\_NovoProjeto: Responsável por criar um novo projeto.  
slot\_GravacaoInicial: Responsável por iniciar a captura dos dados.  
slot\_GravacaoParar: Responsável por parar a captura dos dados.  
slot\_GravacaoDuracao: Responsável cronometrar o tempo de captura.  
processBuffer: Processa o *buffer* que armazena as amostras.  
audioDispositivoMudar: Gera a lista de dispositivos de entrada.  
atualizaHzE: Realiza o cálculo da taxa de amostragem do canal esquerdo.  
atualizaHzD: Realiza o cálculo da taxa de amostragem do canal direito.  
sig\_inicialProcessoE: Processa o sinal contínuo do canal esquerdo.  
sig\_inicialProcessoD: Processa o sinal contínuo do canal direito.  
levelEsquerdo: Gera o nível do sinal do canal esquerdo.  
levelDireito: Gera o nível do sinal do canal direito.

- **processaSinal:** Classe responsável por receber os pacotes de áudio (*buffer*) da placa amplificadora, executar o processamento de detecção de *Spikes* dos dois canais (esquerdo e direito), calculando a frequência de disparo e a realização do *downsample* do sinal.

#### Atributos:

mythead: Realiza os processos da classe processaSinal.  
timer\_id: Recebe o id do timer.  
limiar: Recebe o ponto limiar escolhido pelo usuário.  
downSamples: Armazena o áudio com redução de pontos.  
bufferSpike: Responsável por armazenar os *Spikes*.  
timerCount: Realiza a contagem do tempo.  
isSpike: Recebe o *Spike* detectado para realizar a comparação.  
wait: Recebe a posição do último parâmetro.  
contadorSpikes: Realiza a contagem dos *Spikes* detectados.

#### Métodos:

processaSinal: Construtor da classe.

~processaSinal: Destrutor da classe.

carregaBuffer: Armazena a captura dos dados.

timerEvent: Realiza o cálculo da taxa de disparo.

sinalContínuo: Responsável por enviar o *downsample* para plotar o sinal contínuo nos dois canais (esquerdo e direito).

sinalSpike: Responsável por enviar o *buffer* para plotar os *Spikes* detectados nos dois canais (esquerdo e direito).

hzReady: Responsável por emitir a taxa de disparo em hertz.

- **file:** Classe responsável por criar um arquivo a cada experimento com os *Spikes* detectados do projeto em XML.

#### Atributos:

dir: Caminho para diretório do projeto.

dirExperiencia: Direciona o caminho para gravar o arquivo xml.

myThread: Realiza os processos da classe file.

canal: Gera o arquivo de cada canal no arquivo.

arquivo: Gera o arquivo no caminho da variável dirCapture.

xml: Manipula e descreve os dados.

#### Métodos:

file: Construtor da classe.

~file: Destrutor da classe.

novoProjeto: Gera um novo projeto, salvando o caminho da pasta para guardar as informações do projeto em XML.

novaExperiencia: Responsável por gerar os arquivos com as amostras.

escreverExperiencia: Escreve no arquivo gerado pela função newCapture.

fecharExperiencia: Fecha o arquivo gerado pela função newCapture.

- **dialogNew:** Classe responsável pela janela de novo projeto onde, nesta janela, o usuário pode definir o nome e diretório onde será salvo o projeto.

Atributos:

ui: Gera a interface de novo projeto para o usuário.

dir: Local escolhido para diretório do projeto.

Métodos:

dialogNew: Construtor da classe.

~dialogNew: Destrutor da classe.

getPath: Retorna o caminho do projeto.

saveAs: Define o diretório onde vai ser salvo o arquivo.

check: Checa se todos os campos foram preenchidos.

- **levelWidget:** Classe responsável por gerar as barras verticais que mostram a intensidade do sinal de cada cana distintos.

Atributos:

level: Visualiza a intensidade do sinal nos canais esquerdo e direito.

timer\_id: Recebe o id do timer.

altura: Recebe a altura do nível do sinal.

largura: Recebe a largura do nível do sinal.

Métodos:

levelWidget: Construtor da classe.

~levelWidget: Destrutor da classe.

loadLevel: Carrega os níveis do sinal dos canais.

resizeEvent: Redesenha os níveis quando a janela e redimensionada.

timerEvent: Realiza o cálculo do nível do sinal nos canais.

paintEvent: Responsável por desenhar os níveis do sinal.

- **spikeWidget:** Responsável por conter os eventos de pintura que desenharam o *limiar* para selecionar o valor dos *Spikes* a serem detectados pela classe *processaSinal* nos dois canais.

Atributos:

- \*curvas: Carrega as curvas do *plot*.
- \*grid: Gera o eixo do gráfico.
- \*limiarObject: Limite definido, marcado pela linha.
- limiar: Valor da *limiar* no gráfico atual.
- xAxis: Ponto x do gráfico.
- yAxis: Ponto y do gráfico.
- count: Zera o detector de *Spike* após a visualização da *taxa de disparo*.
- flag: Envia a mensagem para desenhar o *Spike*.

Métodos:

- spikeWidget: Construtor da classe.
  - ~spikeWidget: Destrutor da classe.
  - desenhar: Desenha o gráfico.
  - mouseMoveEvent: Evento que captura as coordenadas do mouse.
  - mousePressEvent: Evento que captura a posição no *click* do mouse.
- **continuoWidget:** Responsável por conter os eventos de pintura que desenham o sinal contínuo dos dois canais (esquerdo e direito), além de conter métodos que trabalham com funções de *zoom* interagir com o usuário.

Atributos:

- \*curvas: Carrega as curvas do *plot*.
- \*grid: Gera o eixo do gráfico.
- \*limiarObject: Limite definido, marcado pela linha.
- \*marker: Desenha o *limiar* nos canais.
- \*limiar: Valor do *limiar* no gráfico atual.
- xAxis: Ponto x do gráfico.
- yAxis: Ponto y do gráfico.
- contador: Contador de pixel da tela dos canais contínuos.
- xTimer: Armazena o número de ponto do eixo x no tempo.

Métodos:

continuoWidget: Construtor da classe.

~continuoWidget: Destrutor da classe.

desenhar: Desenha o gráfico contínuo do sinal nos canais.

- **canvas:** Responsável por gerar as paletas do plano de fundo do gráfico e configurar os *Widgets* da biblioteca *QWT*.

Métodos:

Canvas: Permite desenhar customizando os gráficos.

setupPalette: Permite selecionar a cor para pintar o gráfico.