

**UNIVERSIDADE ESTADUAL DO NORTE DO  
PARANÁ**

***CAMPUS LUIZ MENEGHEL***

**HELLEN CHRISTINA FERRARI**

**DESENVOLVIMENTO DE SOFTWARE LIVRE PARA  
AUXÍLIO À PESQUISA EM NEUROCIÊNCIA -  
SLAPNeuro**

Bandeirantes

2011



**UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ**

**CAMPUS LUIZ MENEGHEL**

**HELLEN CHRISTINA FERRARI**

**DESENVOLVIMENTO DE SOFTWARE LIVRE PARA  
AUXÍLIO À PESQUISA EM NEUROCIÊNCIA -  
SLAPNeuro**

Bandeirantes

2011

**HELLEN CHRISTINA FERRARI**

**DESENVOLVIMENTO DE SOFTWARE LIVRE PARA  
AUXÍLIO À PESQUISA EM NEUROCIÊNCIA -  
SLAPNeuro**

Projeto Final apresentado à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito para aprovação no curso de Sistemas de Informação.

**COMISSÃO EXAMINADORA**

---

Prof. Me. Bruno Miguel Nogueira de Souza  
UENP - Campus Luiz Meneghel

---

Prof. Me. Glauco Carlos Silva  
UENP – Campus Luiz Meneghel

---

Prof. Me. Carlos Eduardo Ribeiro  
UENP – Campus Luiz Meneghel

Bandeirantes, \_\_\_ de \_\_\_\_\_ de 2011

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus sem o qual não chegaria a lugar algum.

A minha família, minha mãe Auzeni, meu pai Emílio, meus irmãos, Emerson, Marco, Cristiano e Ana Lúcia, que sempre estiveram ao meu lado acreditando e apostando em mim.

A meu orientador Bruno Miguel, que com sua sabedoria e paciência pôde me ajudar a concretizar este trabalho.

Aos meus amigos que estiveram ao meu lado, apoiando, auxiliando para a realização do mesmo, a Rosilei, por estar sempre pronta para as minhas viagens para Bandeirantes, a Bruna, por me ouvir desabafar sempre que precisei e mesmo não tendo noção do conteúdo se esforçou para me ajudar, Ana Carolina, Roberta por me apoiarem, a Gabriela que com seu conhecimento de usabilidade adquirido se prontificou para deixar o software mais fácil de aprender, Luiz Artur, Elizabete, Felipe, que me ajudaram e me aguentaram fazendo muitas perguntas e a todos da XIV turma de Sistemas de Informação.

Enfim, agradeço a todos que de alguma forma contribuíram para a realização deste trabalho.

*É graça divina começar bem.  
Graça maior é persistir na caminhada certa.  
Mas graça das graças é não desistir nunca.  
(Dom Hélder Câmara)*

## RESUMO

O presente trabalho tem como objetivo o desenvolvimento de software livre como um ambiente computacional de testes neuropsicológicos. Esse software visa auxiliar tanto a pesquisa, quanto a avaliação e, se possível, a reabilitação neuropsicológica de pacientes com síndromes neurológicas por meio de um ambiente de software integrado e de caráter livre. Tal software foi concebido com recursos da tecnologia Java em parceria com a Universidade Federal de São Paulo – UNIFESP firmado pelo convênio UNIFESP nº 101/2009, sendo que a criação do ambiente é composta por duas partes, a primeira que coletará informações dos sujeitos por meio de um questionário configurável e a segunda que agrupará testes neuropsicológicos. Foi desenvolvido o teste de avaliação da atenção, *Continuous Performance Test (CPT)* – Teste Contínuo de Desempenho. O software criado traz como resultado um relatório final no qual o avaliador poderá realizar o diagnóstico, e assim otimizar a tabulação e interpretação dos dados gerados pelos testes, podendo assim, melhorar na qualidade de pesquisa, uma vez que há redução do tempo para avaliação, além da eliminação de custos com licença e atividades que dizem respeito à avaliação dos sujeito.

**Palavras-chave:** Java, Neuropsicologia, Teste CPT, Netbeans, Banco de Dados Embarcado, Java Swing, Jasper Report e iReport, MVC, Hibernate

## **ABSTRACT**

This work aims to develop free software as a computing environment of neuropsychological tests. This software aims to help both the research and evaluation and, if possible, the neuropsychological rehabilitation of patients with neurological syndromes through an integrated software environment and freeware. This software is designed with features of Java technology in partnership with the Federal University of São Paulo - UNIFESP agreement signed by the UNIFESP No. 101/2009, and creating the environment is composed of two parts, the first to collect information from individuals through a questionnaire and the second configurable collate neuropsychological tests. The test was developed to assess the attention, the Continuous Performance Test (CPT). The software created brings as a result final report in which the assessor can make a diagnosis, and thus optimize the tabulation and interpretation of data generated by tests and could thus improve the quality of research, since there is a reduction of time for evaluation, beyond the elimination of license fees and activities concerning the assessment of the subject.

**Keywords:** Java, Neuropsychology, CPT Test, Netbeans, Embedded Database, Java Swing, Jasper Report and iReport, MVC, Hibernate

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	10
<b>1.1. OBJETIVO</b> .....	11
<b>1.2. JUSTIFICATIVA</b> .....	11
<b>1.3. MATERIAL E MÉTODOS</b> .....	12
<b>2. FUNDAMENTAÇÃO TEÓRICA</b> .....	14
<b>2.2 TESTE - <i>CONTINUOUS PERFORMANCE TEST</i> (CPT)</b> .....	15
<b>2.2.1 Medidas Utilizadas no Teste CPT</b> .....	16
<b>2.3 AMBIENTE INTEGRADO DE DESENVOLVIMENTO</b> .....	19
<b>2.4 SISTEMAS DE BANCO DE DADOS EMBARCADO</b> .....	20
<b>2.5 HIBERNATE ANNOTATIONS</b> .....	20
<b>2.6 ARQUITETURA MODELO VISÃO E CONTROLADOR (MVC)</b> .....	21
<b>2.7 JASPER REPORT E IREPORT</b> .....	23
<b>3 DESENVOLVIMENTO</b> .....	26
<b>3.1 LEVANTAMENTO DE REQUISITOS PARA A CONSTRUÇÃO DO SLAPNEURO</b> .....	26
<b>3.2 DESENVOLVIMENTO DO SISTEMA</b> .....	33
<b>3.2.1 TELAS DE EXECUÇÃO DA INFRAESTRUTURA DO SISTEMA</b> .....	38
<b>3.2.2 Execução do Questionário</b> .....	49
<b>3.3 EXECUÇÃO DO TESTE CPT</b> .....	52
<b>3.4 CONSIDERAÇÕES FINAIS</b> .....	54
<b>4 CONCLUSÃO</b> .....	556
<b>REFERÊNCIAS</b> .....	558

## LISTA DE FIGURAS

Figura 1 – Arquitetura MVC .....	22
Figura 2 – Diagrama processo Jasper Report.....	24
Figura 3 – Diagrama de Caso de Uso.....	27
Figura 4 – Diagrama de Máquina de Estados – Sujeito/Avaliador .....	29
Figura 5 – Diagrama de Máquina de Estados - Questionário .....	30
Figura 6 – Diagrama de Máquina de Estados – Gerenciar Teste e Relatório .....	31
Figura 7 – Modelo Relacional do Bando de Dados.....	32
Figura 8 – Diagrama de Pacotes .....	33
Figura 9 – Tela principal para chamadas das funções.....	34
Figura 10 – Tela aba arquivos.....	35
Figura 11 – Tela aba Cadastros.....	35
Figura 12 - Tela cadastro de questões e questionários.....	36
Figura 13 - Tela aba questionário.....	36
Figura 14 – Tela aba testes .....	37
Figura 15 – Tela aba relatório .....	37
Figura 16 - Tela aba ajuda.....	38
Figura 17 - Tela aba sobre.....	38
Figura 18 - Questões em XML .....	39
Figura 19 - Avaliadores em XML.....	39
Figura 20 - Sujeitos em XML .....	40
Figura 21 - Tela cadastros de sujeitos.....	41
Figura 22 - Tela cadastros de avaliadores .....	41
Figura 23 - Tela efetuado com sucesso.....	42
Figura 24 - Tela confirmação excluir .....	42
Figura 25 - Tela excluído com sucesso .....	42
Figura 26 - Cadastro de grupo de questões .....	43
Figura 27 - Tela exibição Múltiplas Respostas .....	43
Figura 28 - Tela exibição Múltipla Escolha.....	44
Figura 29 - Tela cadastro questão dissertativa.....	44
Figura 30 - Tela quantidade opções para questões. ....	45
Figura 31 - Tela exibição para questão com cinco opções .....	46
Figura 32 - Tela cadastro de questionários.....	47
Figura 33 – Tela dos tópicos de ajuda .....	48
Figura 34 – Tela sobre o sistema .....	49
Figura 35 – Tela Lista de Questionários.....	50
Figura 36 – Tela execução do questionário – Início .....	51
Figura 37 – Tela execução questionário – Questões .....	52
Figura 38 – Mensagem Final de Execução de Questionário.....	52
Figura 39 - Tela estímulo X do teste CPT. ....	53
Figura 40 – Tela estímulo teste CPT.....	54
Figura 41 – Relatório Final.....	55

# 1. INTRODUÇÃO

Com o avanço da psicologia e da neurociência foram criados testes sistematizados para detectar se um indivíduo sofre ou não de alguma síndrome neurológica. A partir da padronização de alguns testes ficou possível a criação de softwares que dessem suporte ao diagnóstico dos pacientes, porém, a maioria dos softwares existentes no mercado é de caráter proprietário, gerando assim, custo elevado para o desenvolvimento de pesquisas nesta área.

Com o intuito de minimizar gastos de pesquisa e sistematizar novos testes, o presente trabalho é proposto a criação de um software livre como um ambiente que dê suporte à inclusão de novos testes de avaliação neuropsicológica.

A pesquisa em neuropsicologia cognitiva tem como objetivo estudar e conhecer a relação das estruturas cerebrais e o comportamento (LURIA, *et al.* 1981). Sendo que as funções cerebrais como: Memória, Atenção, Linguagem, são subdivididas para uma melhor compreensão. Entretanto o acesso a essas funções não se dá de uma forma direta, sendo principalmente investigado através de testes especificamente delineados para avaliar uma função específica.

Com a era digital, às facilidades e à precisão que a computação trouxe, a tendência é que ocorra uma transição dos tradicionais testes de lápis e papel para testes computadorizados. Para isso se faz necessário uma interlocução entre pesquisadores das áreas de estudo de cognição e comportamento humano e os desenvolvedores de sistemas de informação, visando à interdisciplinaridade entre as áreas.

O sistema como um software livre e obtém a licença *opensource*, para que haja distribuição do sistema gratuita e sendo de código-fonte aberto para que seja possível a melhoria ou modificação do mesmo.

O software desenvolvido neste trabalho, por ser de caráter livre, tem o objetivo de facilitar o acesso aos testes e possibilitar a expansão e criação de novos testes por parte dos pesquisadores da área da neurociência. Por

utilizar a tecnologia de banco de dados embarcado não é necessário instalar um sistema gerenciador de banco de dados, desta forma o aplicativo é executado de forma direta e as informações são armazenadas em um banco de dados local. Para a persistência dos dados foi utilizado o framework *Hibernate Annotations*, auxiliando na portabilidade do banco de dados.

Foi criada uma infraestrutura básica para pesquisa, visando a coleta de informações dos sujeitos que realizarão os testes e um teste sistematizado para avaliação da atenção do indivíduo o Teste Contínuo de Desempenho (*Continuous Performance Test - CPT*). Extraindo relatórios dos testes e questionários do banco de dados da aplicação.

### **1.1. OBJETIVO**

Desenvolver software que dê suporte computacional para aplicação de testes que visam avaliação neuropsicológica, como o CPT, sendo que, tal software seja de caráter livre e possua uma infraestrutura básica para o armazenamento de informações dos sujeitos a serem avaliados. Por ser de caráter livre, o objetivo da ferramenta é minimizar restrições financeiras para pesquisa.

#### **Objetivos específicos:**

- Desenvolvimento de uma infraestrutura básica de testes neuropsicológicos que visa coletar informações necessárias através da criação de questionários configuráveis para as pesquisas e avaliações na área em questão;
- Desenvolvimento do teste de avaliação da atenção *Continuous Performance Test* (CPT); e
- Criação de relatório de resultados dos testes, para otimizar a análise dos resultados nas pesquisas neuropsicológicas;

### **1.2. JUSTIFICATIVA**

Dados da Organização Mundial de Saúde (2005) apontam que 6,29% da população mundial sofrem de alguma síndrome neurológica. Neste

sentido, pelo respaldo da neurociência em conjunto com o suporte que a ciência da computação e a tecnologia fornecem, os resultados esperados darão um suporte à Avaliação Neuropsicológica, aumentando a precisão e tornando os resultados da avaliação mais facilmente interpretáveis, bem como elaborar possíveis softwares que ajudem nos processos de Reabilitação Cognitiva.

Atualmente para realizar uma avaliação de funções cognitivas são utilizados diversos testes e paradigmas que em sua grande maioria são desenvolvidos nos grandes centros de excelência do exterior, e isto acarreta alguns problemas para os pesquisadores brasileiros como, por exemplo, as dificuldades financeiras na aquisição dessas tecnologias. Quando se consegue comprar um software, este só pode ser instalado em um computador, sendo usado por poucos pesquisadores ao mesmo tempo. Esses fatores justificam a necessidade do desenvolvimento do software livre que foi especificado. O ambiente de testes neuropsicológicos é de distribuição gratuita e foi desenvolvido utilizando a tecnologia Java.

### **1.3. MATERIAL E MÉTODOS**

Foi utilizado como ferramenta de desenvolvimento do software livre a IDE Netbeans que é um ambiente de desenvolvimento integrado que possibilita a criação de sistemas complexos de computadores baseado na tecnologia Java. Para o desenvolvimento das interfaces gráficas, foi utilizado o Framework JAVA Swing. E para o armazenamento das informações utilizou-se o sistema de gerenciamento de banco de dados embarcado desenvolvido pela Sun, o Derby. Comentou Richard Hillegas (2007).

Para resolver o problema de integração entre o modelo de Orientação a Objetos e o modelo Entidade Relacional (banco de dados) foi utilizado o *Hibernate Annotations*, justificado por sua portabilidade e independência de sistema de banco de dados e baixo acoplamento.

Para a estrutura do sistema optamos pela arquitetura *Model, View e Control* (MVC) para as classes e pacotes ficarem mais bem estruturados. A obtenção de relatórios foi criado com as ferramentas *Jasper Report* e *iReport* para dar estrutura, design e clareza as informações finais.

Definidos a plataforma de desenvolvimento e o sistema de gerenciador de banco de dados, foi realizado um mapeamento das necessidades do software a ser criado, sendo que, o ambiente é composto por uma infraestrutura, no qual são realizados os cadastros: de sujeitos, de avaliadores, que são os pesquisadores profissionais na área de neuropsicologia, e dos grupos para avaliação. A infraestrutura fornece uma interface que dá acesso aos resultados e relatórios que serão elaborados. Também foi criado um mecanismo de exportação e importação de arquivos XML, visando a portabilidade das informações. Os testes neuropsicológicos foram desenvolvidos de forma independente e então acoplados ao ambiente de testes para a elaboração dos resultados. Deste modo, o ambiente possibilita que outros tipos de testes poderão ser vinculados. Para a validação dos testes neuropsicológicos, os participantes da UNIFESP vinculados ao convênio UENP/UNIFESP aplicaram os testes e seguiram os princípios sugeridos na Resolução nº 196, de 10 de outubro de 1996, do Conselho Nacional de Saúde, que determina as normas de experimentação em sujeitos humanos.

Assim se tem o que será abordado nas próximas seções, na fundamentação teórica serão abordados os seguintes tópicos: a neurobiologia da atenção, o funcionamento do CPT com suas medidas utilizadas, o ambiente integrado de desenvolvimento, sistemas de banco de dados embarcado, o que é um SGBD e o funcionamento do Derby, sistema utilizado, a forma de mapeamento de *Hibernate Annotations*, *Jasper Report* e *iReport* para a extração de relatórios.

## 2. FUNDAMENTAÇÃO TEÓRICA

Para melhor entendimento sobre os testes que foram transformados em linguagem computacional, a seguir serão abordados temas como: neurobiologia da atenção, *Continuous Performance Test (CPT)*, ambiente integrado de desenvolvimento, sistemas de Banco de Dados embarcado, hibernate annotations, arquitetura modelo visão e controlador, e jasper report e ireport.

### 2.1 Neurobiologia da Atenção

De acordo com EYSENCK, *et al.* (2007) a atenção pode ser denominada como:

[...] um processo cognitivo pelo qual o intelecto focaliza e seleciona estímulos, estabelecendo relação entre eles. A todo instante recebemos estímulos, provenientes das mais diversas fontes, porém só atendemos a alguns deles, pois não seria possível e necessário responder a todos. É um processo de extrema importância em determinadas áreas, como na educação, já que se exige, por exemplo, a um aluno que preste atenção às matérias lecionadas pelo professor, ignorando outros estímulos visuais, sonoros ou outros, como o que se está a passar fora da sala de aula (estando, neste caso, relacionado também com o problema da disciplina). Além da atenção concentrada, em que se seleciona e processa apenas um estímulo, também pode existir atenção dividida, em que são selecionados e processados diversos estímulos simultaneamente - como quando se conduz um automóvel e se ouvem as notícias do rádio simultaneamente.

Numa revisão de RIZZUTI, *et al.* (2008) dois sistemas de atenção principais relacionam-se às principais funções da atenção. O primeiro é chamado sistema de atenção anterior (SAA), de neurotransmissão dopaminérgica e controlada principalmente pelas áreas pré-frontais, relaciona-se com as habilidades executivas e de inibição de estímulos considerados irrelevantes. O segundo sistema é o sistema de atenção posterior (SAP), controlado pelas áreas parietais do cérebro e de neurotransmissão múltipla (noradrenérgica, serotoninérgica), e está envolvido principalmente com a atenção seletiva. Disfunções do sistema de atenção anterior causam principalmente impulsividade e dificuldade de freio inibitório (auto-regulação

motora e cognitiva) enquanto as disfunções do sistema posterior estão mais relacionadas à dificuldade de atenção seletiva, atenção dividida e concentrada, diz RIVERO (2010) *apud* RIZZUTI (2008).

## **2.2 TESTE - *CONTINUOUS PERFORMANCE TEST (CPT)***

Segundo RIVERO (2010), o Teste de Desempenho Contínuo é uma tarefa de vigilância desenvolvida inicialmente por Rosvold et al. (1956) para discriminar déficits de atenção sustentada em indivíduos com lesões cerebrais. A versão original do CPT era uma tarefa visual de vigilância, incorporando elementos de seleção de estímulos, atenção sustentada e inibição de respostas irrelevantes. De forma geral, a tarefa original requer que o sujeito mantenha a vigilância através da rápida apresentação de diversos estímulos diferentes, por exemplo: letras, com um alvo determinado, sendo sensível para discriminar déficits de atenção em indivíduos com diversos distúrbios do Sistema Nervoso Central (Rosvold *et al.* 1956). Por essa sensibilidade, tornou-se um dos mais populares instrumentos de pesquisa e diagnóstico de déficits de atenção (Egeland & Kovalik-Gran, 2008; Edwards *et al.* 2007).

De forma geral, os paradigmas CPT requerem que o sujeito mantenha a vigilância e responda rapidamente quando a detecção da presença, apertando a barra de espaço do teclado sempre que aparecer letra na tela, 324 vezes e ausência (não apertando tecla nenhuma) de um estímulo específico num conjunto de estímulos distratores apresentados continuamente por um tempo prolongado. Este estímulo pode ser simples, por exemplo, uma letra “X” ou uma sequência de estímulos, que aparecerão 36 vezes. Em geral os índices gerados nessas versões correspondem a número de erros por omissão (*omissions*), quando a resposta não é dada após aparecer o alvo, e considerado como medida de inatencão, número de erros por co-omissão ou falsos alarmes (*commissions*), quando o sujeito responde ao estímulo não-alvo, que pode refletir impulsividade; (Corkum & Siegel, 1993; Klee & Garfinkel, 1983; Levy, 1980). As letras serão apresentadas por 250 milissegundos.

Várias versões do CPT foram desenvolvidas ao longo dos estudos (CPT-X; CPTAX; CPT-XX; CPT-IP, 1998) que variam em relação à modalidade (auditiva ou visual), o tipo de estímulo apresentado (letras, números,

cores ou formas geométricas), a natureza da tarefa (responder a estímulos simples, a uma sequência de estímulos ou a cada estímulo com exceção de um).

### 2.2.1 Medidas Utilizadas no Teste CPT

São utilizadas quinze medidas para avaliação no teste CPT, segue-as:

- nº de omissões;
- % de omissões;
- nº de comissões;
- % de comissões;
- TR (ms) – (Tempo de Reação em milissegundos);
- EP do TR – (Erro Padrão do Tempo de Reação);
- Variabilidade;
- Detectabilidade ( $d'$ );
- Estilo de resposta ( $\beta$ );
- nº de perseverações;
- % de perseverações;
- Mudança de TR por bloco;
- EP das Mudanças de TR por bloco;
- Mudanças de TR por IIE; e
- EP das Mudanças de TR por IIE.

O teste tem como resultado as quinze medidas de avaliação, tendo abaixo suas respectivas definições separadas por blocos, como: Erros, Tempo de reação, Teoria da detecção de sinais, Resultados por blocos e Resultados por ISIs (intervalos inter-estímulos – *interstimulus intervals*).

**Erros** - Medidas de exatidão da resposta:

São dados em duas categorias:

a) Omissão (*omission*): são os números de alvos para o qual o indivíduo não respondeu, isto é, quando a resposta não é dada após aparecer uma letra (não-X) na tela.

b) Comissão (*commission*): são os números de vezes que o indivíduo respondeu ao não-alvo, isto é, quando a resposta é dada após um X aparecer na tela.

**Tempo de reação (*Hit Reaction Time*)** - é a velocidade e consistência de reação medida em milissegundos (ms):

São fornecidas as seguintes medidas:

a) Tempo de reação (*Hit Reaction Time - Hit RT*) é definido como a média de tempo de resposta (em ms) para todas as respostas-alvo e para todos os seis blocos.

b) Erro-padrão do tempo de reação (*Hit RT Std. Error*): Consistência do tempo de resposta, para respostas ao alvo. Escores elevados indicam maior variabilidade.

c) Variabilidade – erro-padrão (*Variability - Std Error*): mede a variabilidade do respondente (variabilidade que o indivíduo mostra em relação ao seu próprio erro-padrão geral).

**Teoria da Detecção de Sinais (*Signal Detections Theory*):**

a) Estilo de resposta (*Response Style*) ou Índice Beta ( $\beta$ ): indica o critério de resposta do sujeito.

b) Índice de detectabilidade -  $d'$  - (*Detectability*): é o índice de capacidade de discriminação dos sinais. É uma medida do grau de discriminação do sujeito entre o alvo (não-X) e o não-alvo (X).

c) Perseverações (*Perseverations*): definida como a resposta que ocorre menos de 100 ms após a apresentação do estímulo.

**Resultados por blocos:**

O CPT é apresentado em seis blocos e isso permite analisar os resultados por blocos, avaliando assim mudanças na velocidade e no tempo de

reação depois de decorrido o tempo e também a vigilância (concentração), e ainda a consistência de respostas conforme o teste progride. Duas medidas são geradas:

a) Mudanças de tempo de reação (*Hit RT Block Change*): indicam a mudança no tempo de reação com o decorrer dos seis blocos.

b) Erro-padrão das mudanças de tempo de reação (*Hit SE Block Change*): valores positivos indicam menos consistência no tempo de reação e valores negativos de maior consistência.

**Resultados por ISIs** - o teste é apresentado em diferentes intervalos inter-estímulos (ISIs – interstimulus intervals). As medidas geradas avaliam a habilidade do sujeito para se ajustar a mudanças de tempo para responder à demanda da tarefa, isto é, as mudanças na taxa de apresentação dos estímulos:

O teste é apresentado em diferentes intervalos inter-estímulos (*ISIs – interstimulus intervals*). As medidas geradas avaliam a habilidade do sujeito para se ajustar a mudanças de tempo para responder à demanda da tarefa, isto é, as mudanças na taxa de apresentação dos estímulos. São duas as medidas:

a) Mudanças de tempo de reação por ISI (*ISI Hit RT ISI Change*): valores positivos indicam lentificação no tempo de reação conforme aumenta o ISI, e negativos indicam tempo de reação mais rápido conforme aumenta o ISI.

b) Erro-padrão (*Hit SE ISI Change*): positivo indica menor consistência no tempo de reação durante ISIs longos, e negativo maior consistência com ISIs longos.

As 15 medidas do CPT são utilizadas para avaliar um transtorno específico como o Transtorno de Déficit de Atenção e Hiperatividade (TDAH). O TDAH é uma entidade nosológica mental de início precoce que surge na infância, com evolução crônica e que repercute em vários contextos e é

caracterizado por desatenção, atividade motora excessiva, hiperatividade e impulsividade (BIEDERMAN, *et al.* 1998).

## 2.3 AMBIENTE INTEGRADO DE DESENVOLVIMENTO

Como esclarece no site oficial, a IDE NetBeans auxilia programadores a escrever, compilar, debugar e instalar aplicações, e foi arquitetada em forma de uma estrutura reutilizável que visa simplificar o desenvolvimento e aumentar a produtividade, pois reúne em uma única aplicação todas estas funcionalidades. Totalmente escrita em Java, mas que pode suportar qualquer outra linguagem de programação que desenvolva com Swing, como C, C++, Ruby e PHP. Também suporta linguagens de marcação como XML e HTML.

O NetBeans fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e APIs (*Application Program Interface*, um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de software) além de uma documentação vasta, inclusive em português, bem organizada. Tais recursos auxiliam o desenvolvedor a escrever seu software de maneira mais rápida. A distribuição da ferramenta é realizada sob as condições da SPL (*Sun Public License*), uma variação da MPL (*Mozilla Public License*). Esta licença tem como objetivo garantir a redistribuição de conhecimento à comunidade de desenvolvedores quando novas funcionalidades forem incorporadas à ferramenta.

Atualmente está distribuído em diversos idiomas e isto tem o tornado cada vez mais popular, facilitando o acesso a iniciantes em programação e possibilitado o desenvolvimento de aplicativos multilíngüe. Como o NetBeans é escrito em Java, é independente de plataforma, funciona em qualquer sistema operacional que suporte a máquina virtual Java (JVM).

Como encontrado no documento JDK 6 Swing (*Java Foundation Classes (JFC)*), feito pelo criados Swing é um *widget toolkit* para uso com o Java. Ele é compatível com o *Abstract Window Toolkit (AWT)*, mas trabalha de uma maneira totalmente diferente. A API Swing procura desenhar por conta

própria todos os componentes, ao invés de delegar essa tarefa ao sistema operacional, como a maioria das outras APIs de interface gráfica trabalham.

Por ser uma API de mais alto nível, ou seja, mais abstração, menor aproximação das APIs do sistema operacional, ela tem bem menos desempenho que outras APIs gráficas e consome mais memória RAM em geral. Porém, ela é bem mais completa, e os programas que usam Swing têm uma aparência muito parecida, independente do Sistema Operacional utilizado.

## **2.4 SISTEMAS DE BANCO DE DADOS EMBARCADO**

Zikopoulos (2005) destaca que banco de dados é um conjunto de registros dispostos em estrutura regular que possibilita a reorganização dos mesmos e produção de informação. Um banco de dados normalmente agrupa registros utilizáveis para um mesmo fim.

Um banco de dados é usualmente mantido e acessado por meio de um software conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Normalmente um SGBD adota um modelo de dados, de forma pura, reduzida ou estendida. Muitas vezes o termo banco de dados é usado, de forma errônea, como sinônimo de SGBD. E um banco de dados embarcado é caracterizado por não haver necessidade de instalação de um sistema de gerenciamento para executar o software.

Apache Derby é um banco de dados relacional Java que pode ser embutido em programas Java e usado para processamento de transações online. Consome apenas 2MB de espaço em disco. O Apache Derby é desenvolvido como um projeto open source sob a licença Apache 2.0. E era anteriormente distribuído como IBM *Cloudscape*. Atualmente é distribuído como Sun Java DB.

O motor de banco de dados do Derby é banco de dados relacional com funcionalidades completas. O JDBC e SQL são as APIs utilizadas, cita Zikopoulos (2005).

## **2.5 HIBERNATE ANNOTATIONS**

A comunidade do *Hibernate, Documentation – Hibernate – Jboss Community*, explica que *Hibernate* é um *Framework* para mapeamento Objeto/Relacional em Java e possibilita desenvolver classes persistentes usando Java convencional: associação, composição, herança, polimorfismo e coleções Java. Permite implementar mecanismos de mapeamento: Classes Java, tabelas em SGBDs relacionais, tipos Java, tipos SQL. Permite também implementar mecanismos convenientes para consulta e recuperação de dados.

O *Hibernate* persiste objetos Java comuns (POJO), usa reflexão para acessar as propriedades persistentes de um objeto. As classes persistentes são definidas em documentos de mapeamento e arquivos XML são usados para descrever os campos, associações e subclasses persistentes. Os Mapeamentos são "compilados" na inicialização da aplicação Podem ser usados também para operações de suporte como: Geração de esquemas do banco de dados Geração de código-fonte Java. (Persistência usando Hibernate).

O objetivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional (presente na maioria dos SGBDs). Em especial, no desenvolvimento de consultas e atualizações dos dados.

Sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL). O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual da conversão dos dados resultante, mantendo o programa portátil para quaisquer bancos de dados SQL, porém causando um pequeno aumento no tempo de execução. (*Documentation – Hibernate – Jboss Community*).

## 2.6 ARQUITETURA MODELO VISÃO E CONTROLADOR (MVC)

**Model-view-controller (MVC)** é um padrão de arquitetura de software que visa a separar a lógica de negócio da lógica de apresentação, permitindo o desenvolvimento, teste e manutenção isolado de ambos.

Abaixo segue uma figura do ciclo da arquitetura MVC.

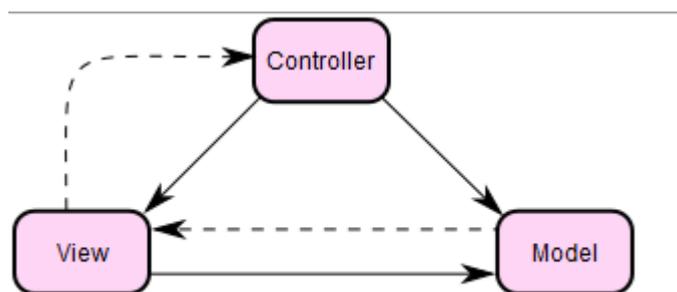


Figura 1 – Arquitetura MVC

## Componentes

O modelo (*model*) é usado para definir e gerenciar o domínio da informação e notificar observadores sobre mudanças nos dados. Ele é uma representação detalhada da informação que a aplicação opera. A lógica de negócio adiciona valor semântico aos dados, e quando há mudança de estado o modelo notifica seus observadores. Por exemplo, *aluno*, *professor* e *turma* fazem parte do domínio de um sistema acadêmico. Operações como calcular a média final do aluno ou o índice de faltas da turma fazem parte da lógica de domínio. A forma como o dado é armazenado ou acessado não é de interesse do MVC, assume-se que é de responsabilidade do modelo. Xerox Parc (1978-79)

A visão (*view*) apresenta o modelo num formato adequado ao utilizador, na saída de dados, e diferentes visões podem existir para um mesmo modelo, para diferentes propósitos.

O controlador (*controller*) recebe a entrada de dados e inicia a resposta ao utilizador ao invocar objetos do modelo, e por fim uma visão baseada na entrada. Ele também é responsável pela validação e filtragem da entrada de dados.

Ainda explica Xerox Parc (1978-79), um caso prático é uma aplicação web em que a visão é um documento HTML (ou derivado) gerado pela aplicação. O controlador recebe uma entrada GET ou POST após um estímulo do utilizador e decide como processá-la, invocando objetos do domínio para

tratar a lógica de negócio, e por fim invocando uma visão para apresentar a saída.

Allan (1997) justifica o porquê utilizar este tipo de arquitetura, com o aumento da complexidade das aplicações desenvolvidas, torna-se relevante a separação entre os dados e a apresentação das aplicações. Desta forma, alterações feitas no *layout* não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o *layout*.

Esse padrão resolve este problema por meio da separação das tarefas de acesso aos dados e lógica de negócio, continua Allan, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois, o controlador.

## 2.7 JASPER REPORT E IREPORT

### 2.7.1 Jasper Report

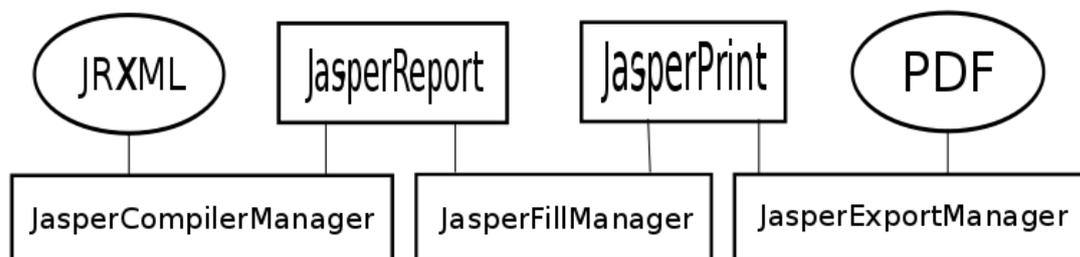
O *JasperReports* é um framework para a geração de relatórios. É uma ferramenta totalmente *open source* e gratuita, e a mais utilizada com esse propósito atualmente. Entre as funcionalidades do *JasperReports* podemos destacar:

- é capaz de exportar relatórios para diversos formatos diferentes, tais como PDF, HTML, XML, XLS, etc.
- aceita diversas formas de entrada de dados, tais como um arquivo XML ou CSV, conexão com o banco de dados, uma sessão do Hibernate, uma coleção de objetos em memória, etc.
- permite o uso de diagramas, gráficos, e até códigos de barras.

Um aspecto importante do *JasperReports* é que o *layout* do relatório é definido em um arquivo XML, geralmente com a extensão *.jrxml*. Este XML possui todas as informações de formatação do relatório, e além disso, possui os campos que serão preenchidos posteriormente, de acordo com a fonte de dados utilizada (*data source*). Como dito anteriormente, a fonte de dados

pode variar, e ser uma tabela em uma base de dados, ou ser um arquivo CSV, porém a formatação do relatório será a mesma em ambos os casos.

Os passos para gerar um relatório são bem simples. O primeiro passo é compilar o relatório em XML. Depois da compilação, o resultado é um objeto do tipo *JasperReport*. O próximo passo é preencher o relatório com os dados, e o resultado dessa etapa fica armazenado em um objeto do tipo *JasperPrint*. Esse objeto já representa o relatório finalizado, a partir dele podemos enviar para impressão diretamente, ou podemos exportar para um outro formato, tal como PDF por exemplo. Veja um diagrama ilustrando o processo completo:



F

figura 2 – Diagrama processo Jasper Report

### 2.7.2 iReport

O *iReport* é uma ferramenta desenvolvida pela mesma empresa do *JasperReports*, a *JasperForge*, e por isso é muito comum ver os dois sendo usados em conjunto. Uma das dificuldades ao trabalhar com os relatórios está na definição do *layout*. É complicado escrever o layout totalmente em XML, sem ter que se aprofundar em todas as *tags* e atributos possíveis, e além disso posicionar todos os elementos corretamente. Na prática, é muito raro alguém editar o JRXML manualmente, e sim apenas para fazer alguns pequenos ajustes quando necessários. O processo normal é utilizar alguma ferramenta para gerar o JRXML automaticamente, e o *iReport* é utilizado com esse propósito.

O *iReport* é um aplicativo gráfico, que permite que você “desenhe” um relatório, utilizando uma palheta, e arrastando e soltando componentes, de forma bem parecida com a criação de interfaces e janelas para programas. Ao salvar, automaticamente será gerado um JRXML que você

poderá utilizar na aplicação que estiver desenvolvendo. A vantagem é que não é necessário que você conheça a fundo o XML a ser editado, economizando tempo de desenvolvimento. Ele também traz um conjunto pronto de *templates* que você já pode utilizar diretamente, ou então, escrever seus próprios *templates* e reaproveitá-los sempre que precisar criar um novo tipo de relatório.

Neste capítulo foram abordados o funcionamento do teste CPT e as descrições das ferramentas e estruturas que foram utilizadas para a construção do sistema. No próximo capítulo será apresentado o desenvolvimento e mostrando como as mesmas foram empregadas.

### 3 DESENVOLVIMENTO

Para o desenvolvimento foi necessário a construção de diagramas *Unified Modeling Language* (UML) para o levantamento de requisitos e para melhor estruturar a execução do sistema como temos nos tópicos a seguir: no tópico 3.1 temos o levantamento de requisitos para a construção da aplicação, contendo os seguintes diagramas: diagrama de casos de uso, diagrama de máquina de estados, modelo relacionado do banco de dados, diagrama de pacotes, no tópico 3.2 será mostrado como o sistema foi desenvolvido, ilustrando o desenvolvimento por meio de das telas principais do sistema, mostrando as abas e funções existentes. Já o item 3.3 é mostrado a execução da infraestrutura, como telas de cadastros. Por fim no item 3.4 apresenta como foi implementado o teste CPT, mostrando a tela de execução do referido teste.

#### 3.1 LEVANTAMENTO DE REQUISITOS PARA A CONSTRUÇÃO DO SLAPNEURO

O levantamento de requisitos é uma das etapas mais importantes do desenvolvimento de um software. Para a criação do SLAPNeuro, foram consultados pesquisadores da UNIFESP que elencaram os requisitos para a construção do software. A partir do levantamento inicial, foram criados diagramas para confrontar com os pesquisadores se o entendimento do sistema estava correto.

Dessa forma o diagrama de casos de uso corresponde a uma visão externa do sistema e representa graficamente os atores, os casos de uso, e os relacionamentos entre estes elementos e tem como objetivo ilustrar em um nível alto de abstração quais elementos externos interagem com que funcionalidades do sistema. A finalidade de um diagrama de caso de uso é apresentar um tipo de diagrama de contexto que apresenta os elementos externos de um sistema e as maneiras segundo as quais eles as utilizam. Segue ilustração. (Figura 3).

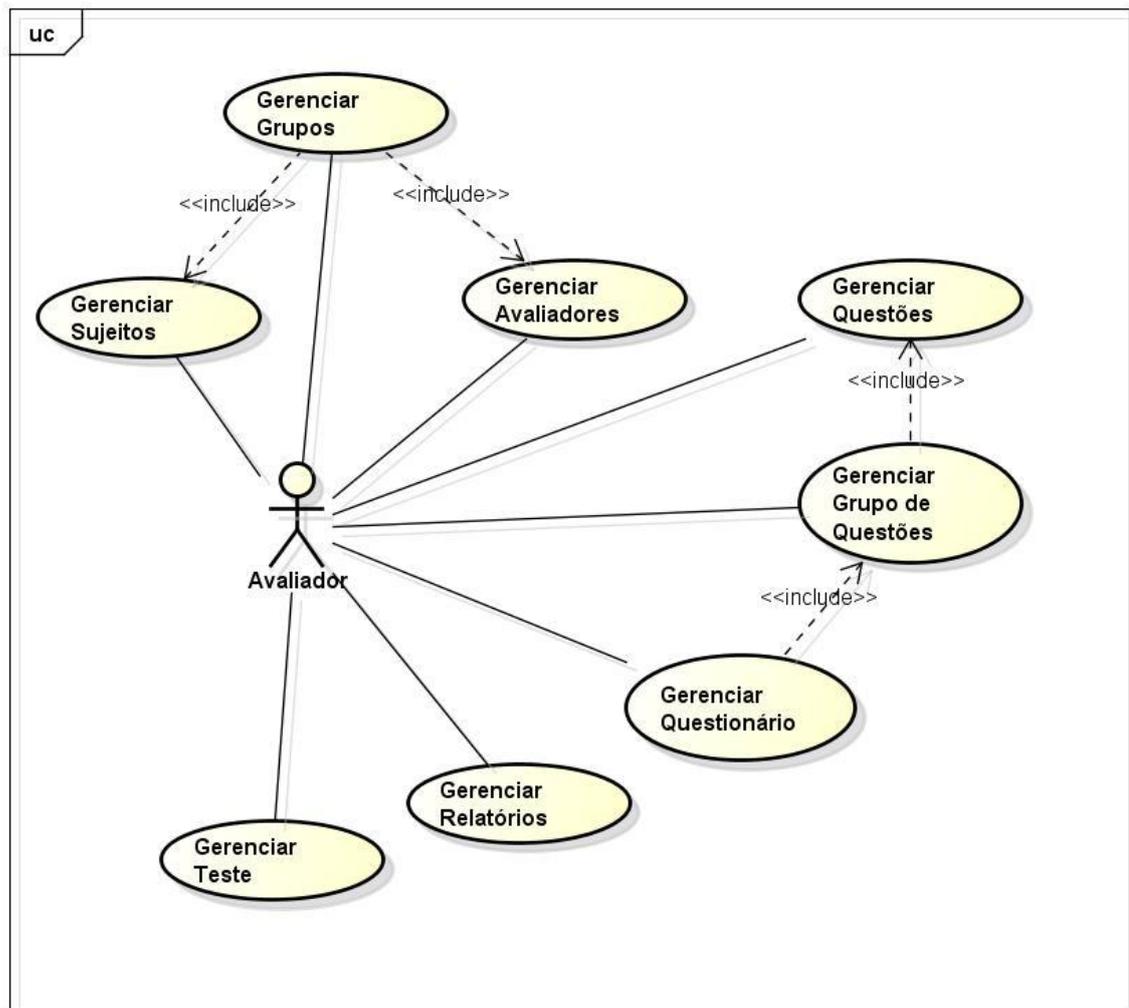


Figura 3 – Diagrama de Caso de Uso

A seguir o cenário principal e alternativo do caso de uso.

### **Cenário Principal**

- O avaliador solicita cadastros de avaliadores, sujeitos, questões ou questionários.
- O avaliador preenche os campos pedidos.
- O sistema grava esses dados.
- O avaliador solicita a execução do questionário, escolhendo qual sujeito irá responder.
- O sistema armazena os dados em tempo de execução.
- O avaliador solicita execução do teste CPT.

- O sistema armazena os dados após a execução.
- O avaliador solicita o relatório final do sujeito indicado.

### **Cenário Alternativo**

- No passo 1 do cenário principal o avaliador solicita um cadastro de questionário sem questões ou grupos existentes. Inviabilizando o cadastro de um novo questionário.

- No passo 4 do cenário principal o avaliador solicita a execução do questionário não havendo questionário ou sujeito cadastrado. Inviabilizando a execução de questionário.

- No passo 8 do cenário principal o avaliador solicita o relatório final não contendo questionário respondido ou teste realizado. Havendo ausência de informações no relatório.

Como levantamentos de requisitos também contem o diagrama de máquina de estados separados por casos de uso.

Para representar a situação em que um objeto pode se encontrar no decorrer da execução é ilustrado pela Figura 4 logo abaixo, o diagrama para cadastro de sujeito ou avaliador no sistema.

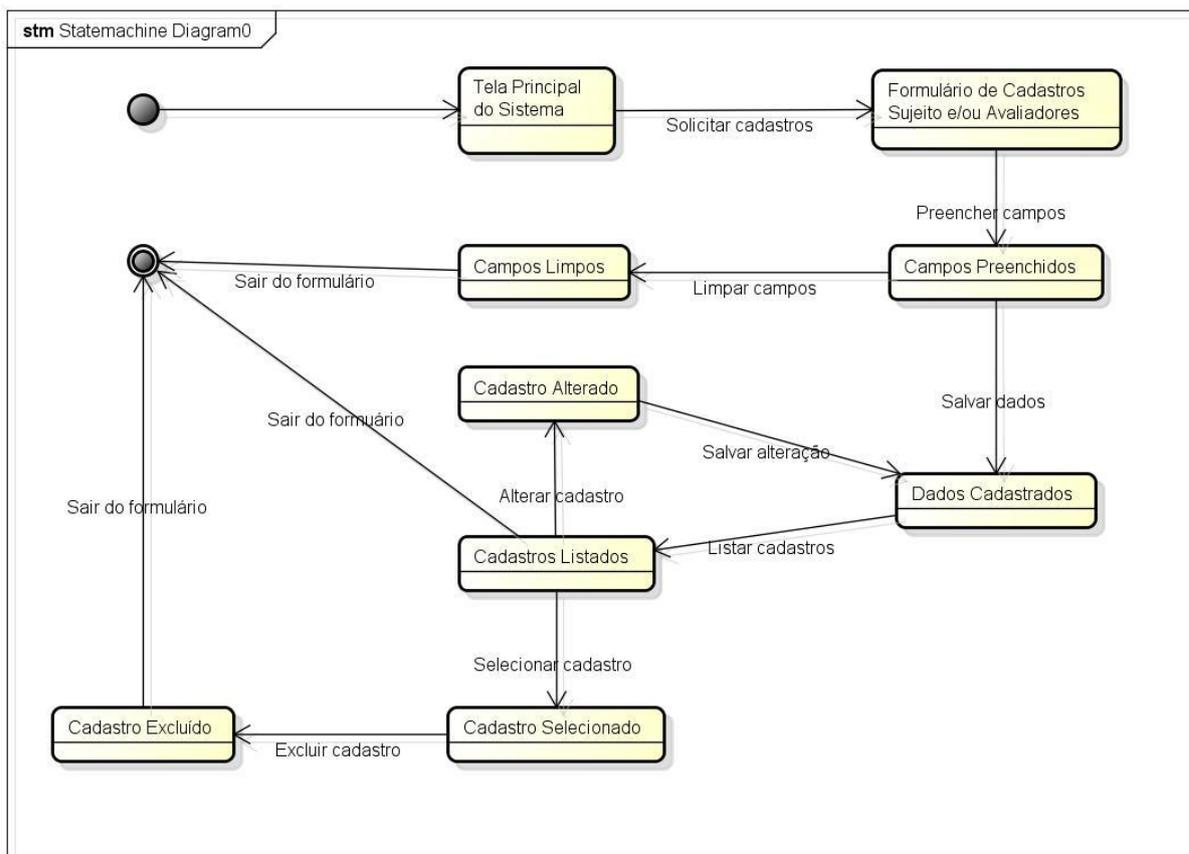


Figura 4 – Diagrama de Máquina de Estados – Sujeito/Avaliador

Na Figura 5 é ilustrado o diagrama de estados para o caso de uso manter questionário, no qual são as etapas que se deve seguir.

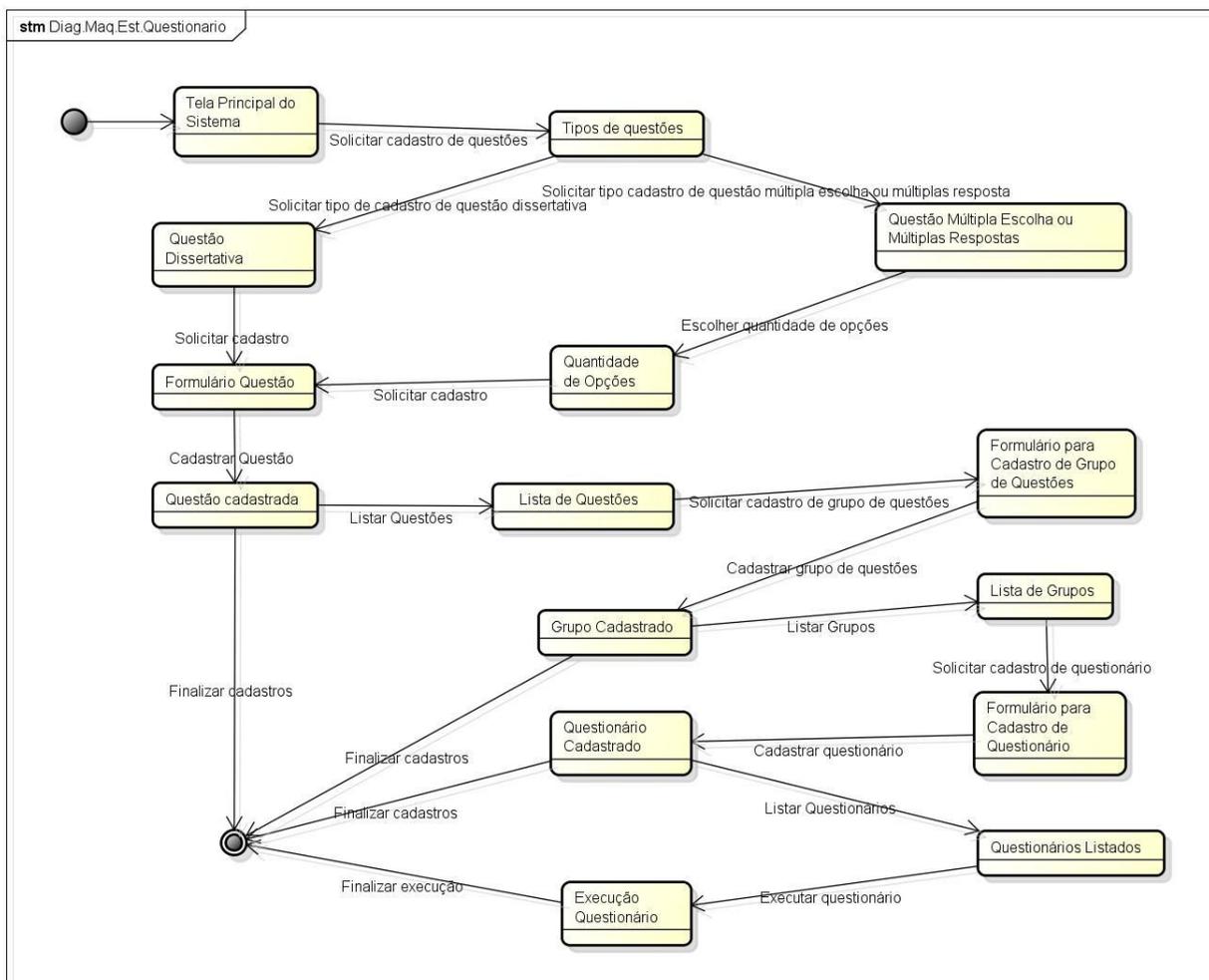


Figura 5 – Diagrama de Máquina de Estados - Questionário

Os estados que estarão para os teste e emissão de relatório são mostrados no diagrama a seguir. Figura 6.

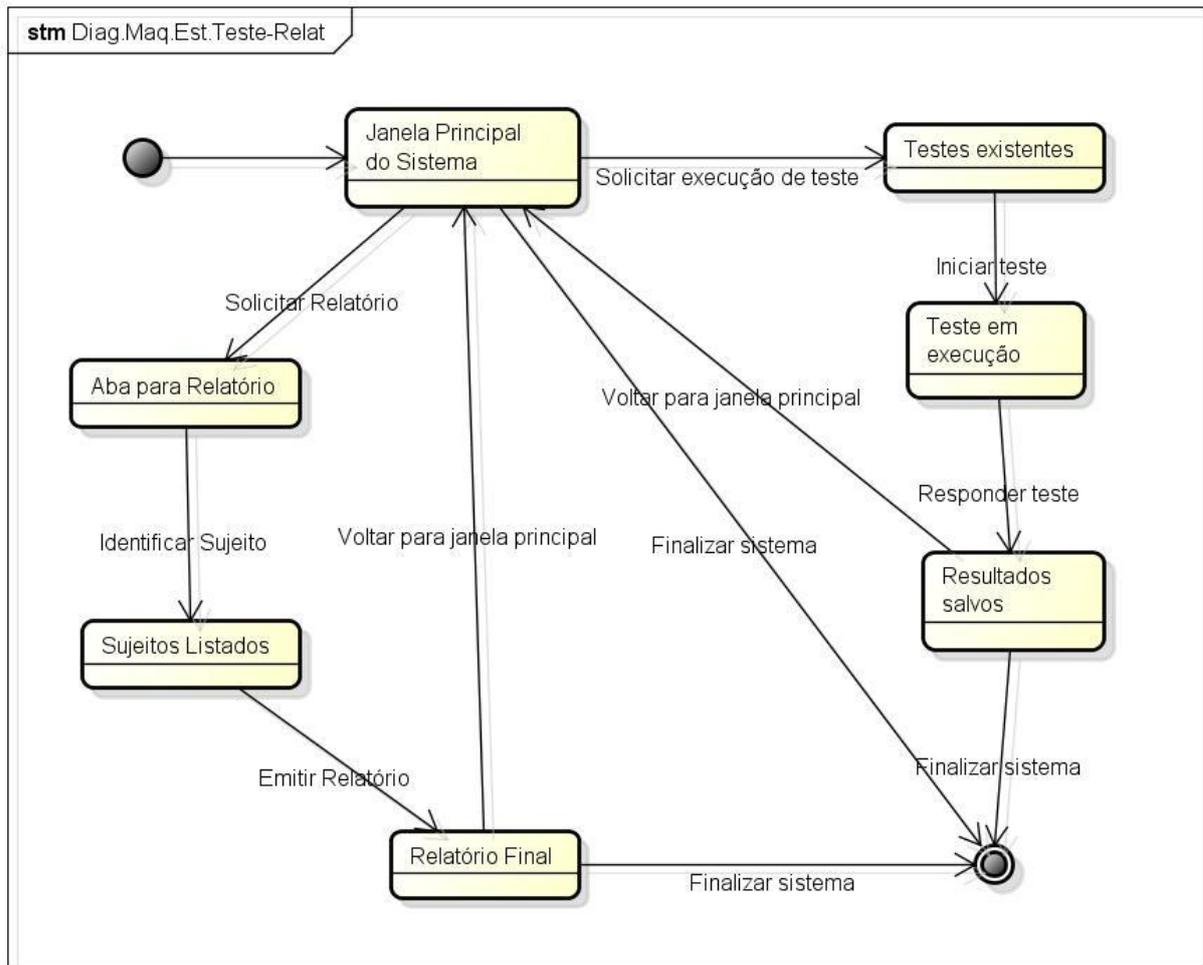


Figura 6 – Diagrama de Máquina de Estados – Gerenciar Teste e Relatório

Além dos diagramas da fase de requisitos, foi desenvolvido o Modelo Relacional para auxiliar na construção do banco de dados. Este Modelo é um projeto estrutural do que será o Banco de Dados, composto de tabelas, atributos e seus relacionamentos. (Figura 7).

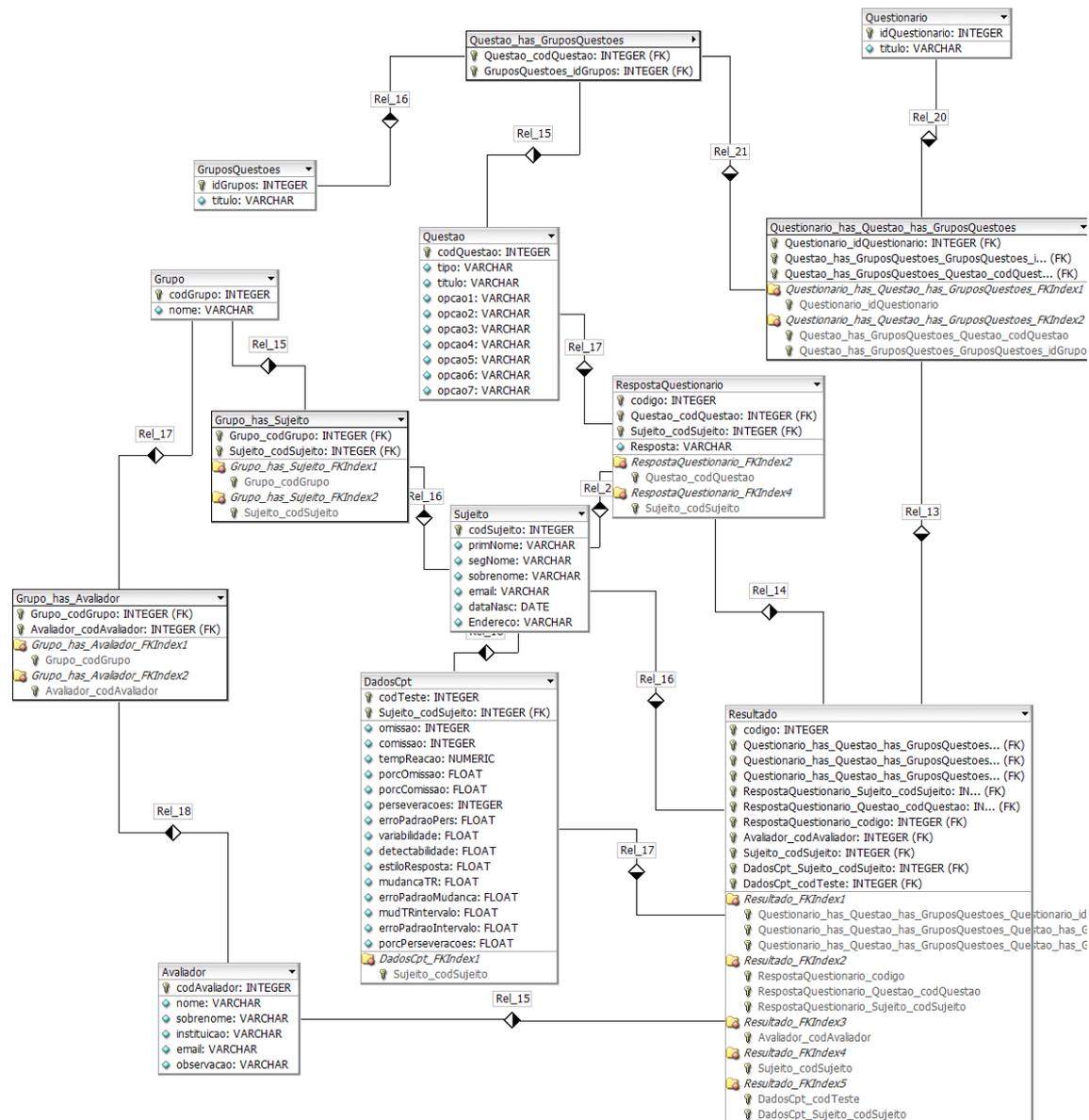


Figura 7 – Modelo Relacional do Bando de Dados

No modelo relacional da Figura 7, tem-se como está estruturado o banco de dados, sendo que este possui um total de sete tabelas, no qual a tabela grupo é composta por sujeito e avaliador que relacionam com a resposta, dados do teste e resultado, tendo a tabela de resposta que também é composta por questões e questionários e relacionando também com a tabela resultado, de onde serão extraídas as informações para a geração do relatório final.

Para definir a estrutura do sistema, foi criado o diagrama de pacotes, que é definido pela UML e descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos mostrando as dependências entre estes.

Representando assim, um grupo de classes ou outros elementos que se relaciona com outros pacotes através de uma relação de dependência. Na Figura 8 apresenta o diagrama de pacotes do software SLAPNeuro.

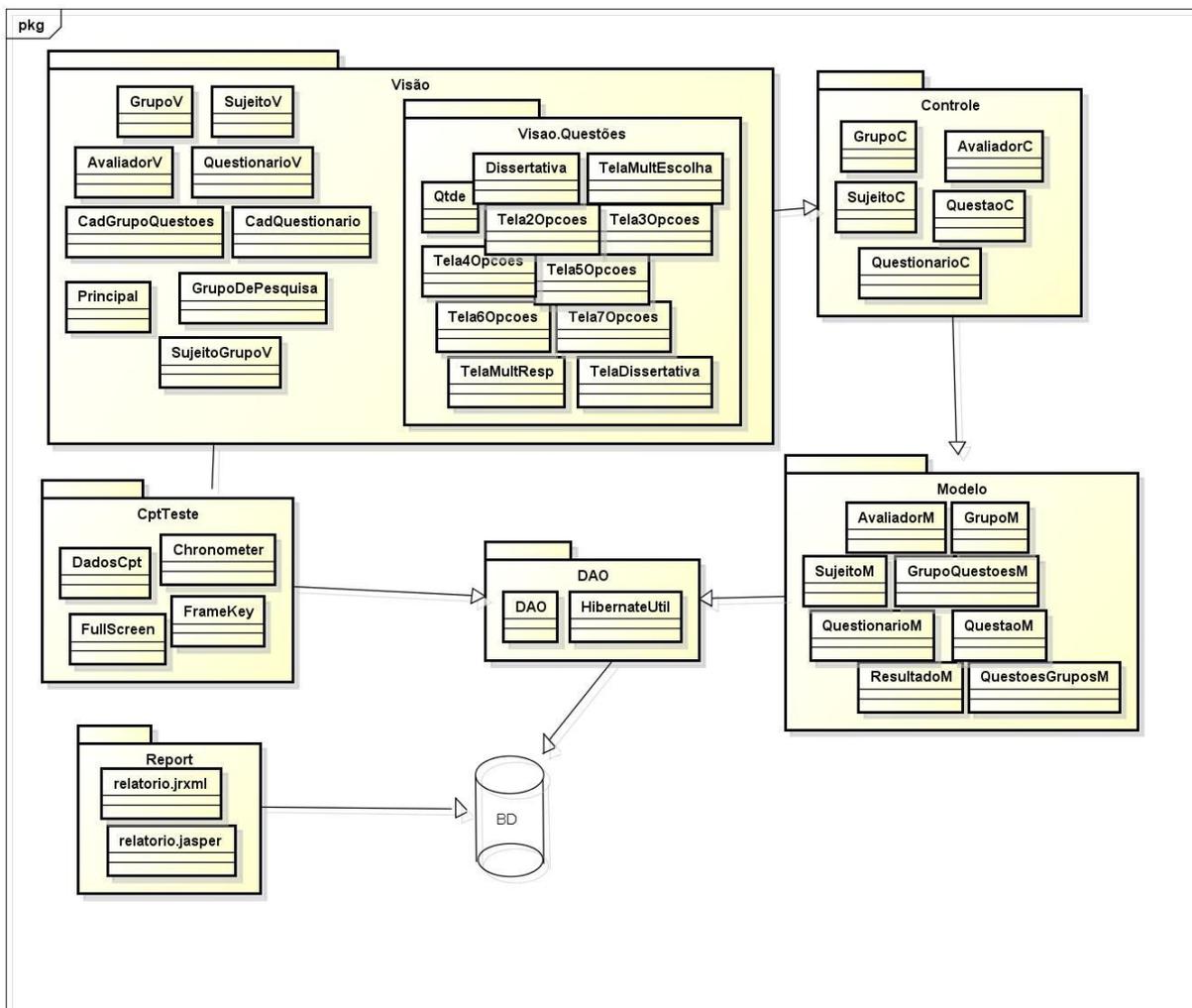


Figura 8 – Diagrama de Pacotes

Após as análises e a criação do projeto utilizando as notações da UML, iniciou-se o processo de implementação, com as tais ferramentas especificadas na seção 1.3, que será tratado na próxima seção.

### 3.2 DESENVOLVIMENTO DO SISTEMA

Na fase de desenvolvimento, foram criados diversos formulários para compor a infraestrutura do software, como: manter avaliadores, sujeitos, grupos de pesquisa e início dos testes. E chamar a função para cadastro de questionários.

No menu arquivo é possível exportar e importar arquivos XML para serem manipulados. Em relatórios poderá exibir todos os resultados já avaliados com o programa. Na figura a seguir é apresentada a tela inicial do software (Figura 9).

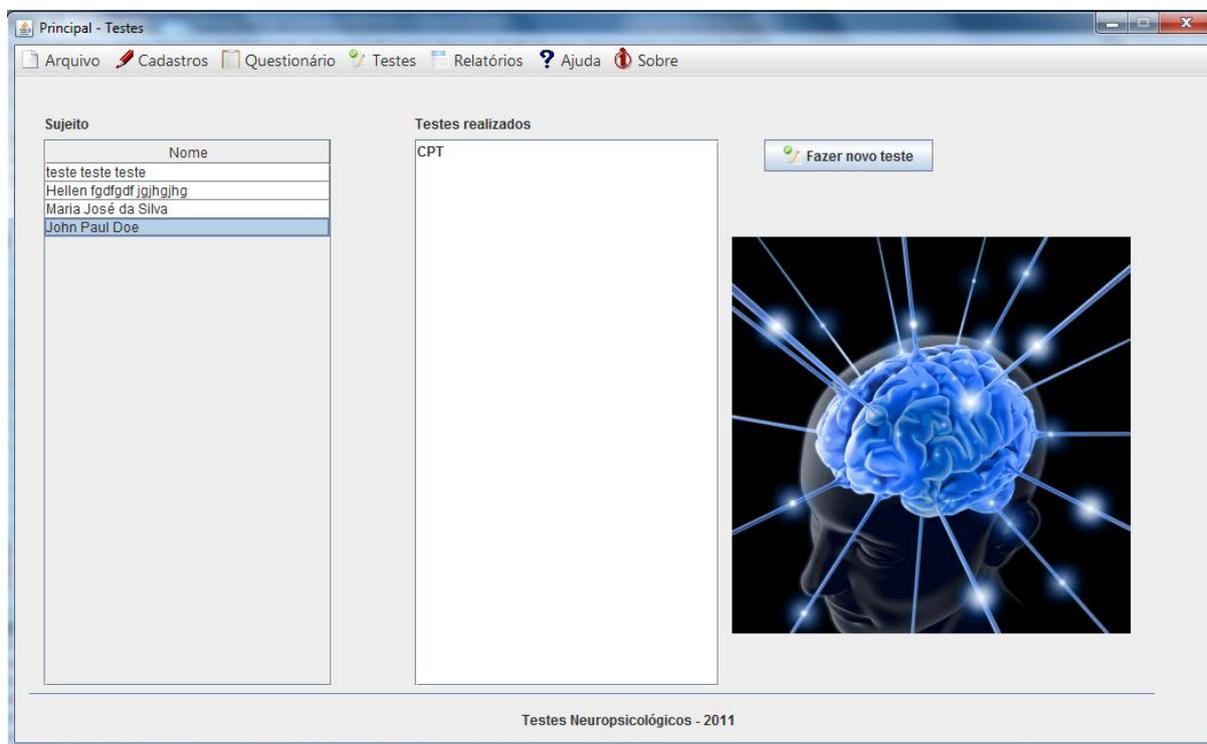


Figura 9 – Tela principal para chamadas das funções

Na Figura 10 apresenta a possibilidade de importar e exportar arquivos XML

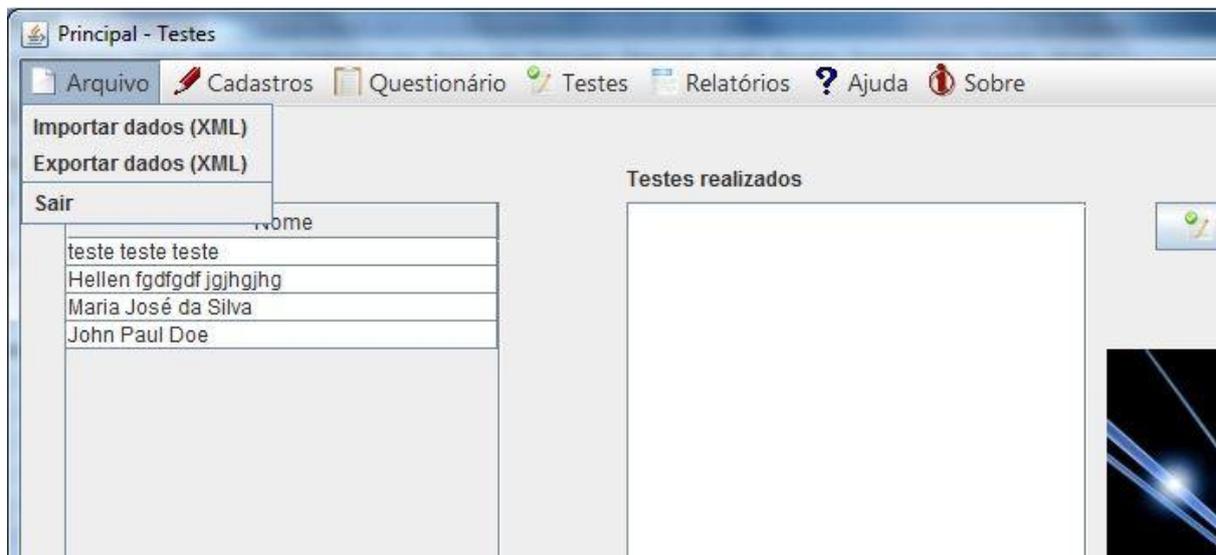


Figura 10 – Tela aba arquivos

Exibir tipos de cadastros, sendo eles, de sujeitos, avaliadores ou grupos de pesquisa e um novo questionário, segue na Figura 11.

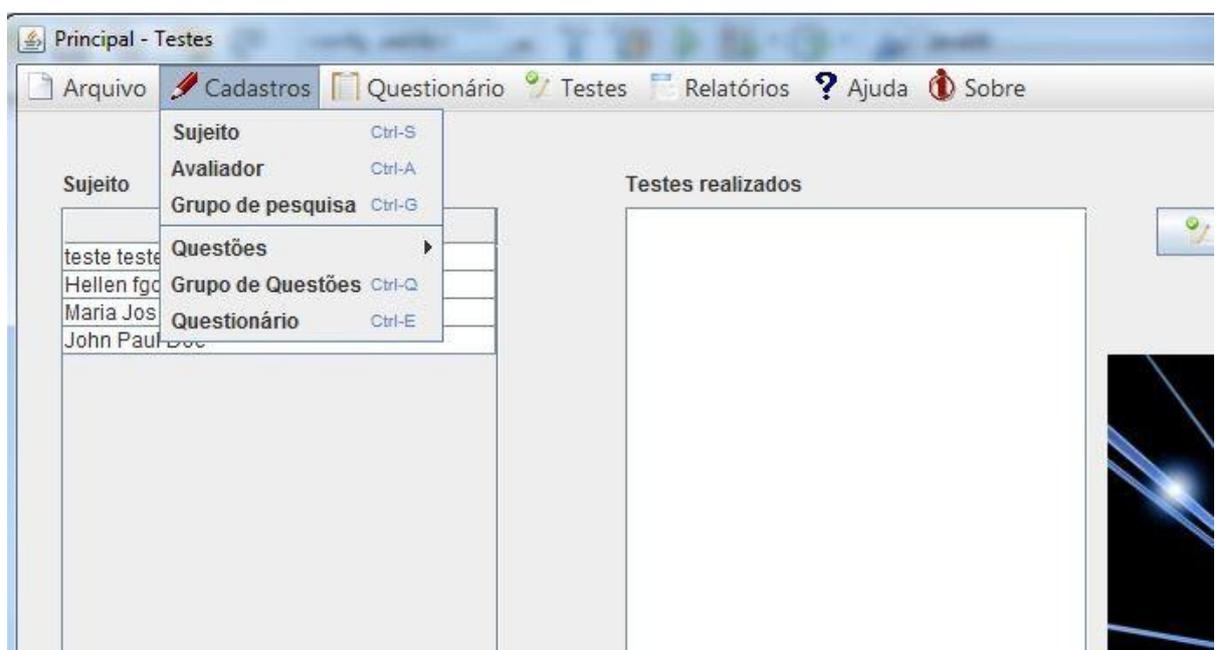


Figura 11 – Tela aba Cadastros

A seguir, tela de tipos cadastro de questões, sendo elas, dissertativa, múltipla escolha ou múltiplas respostas. (Figura 12).

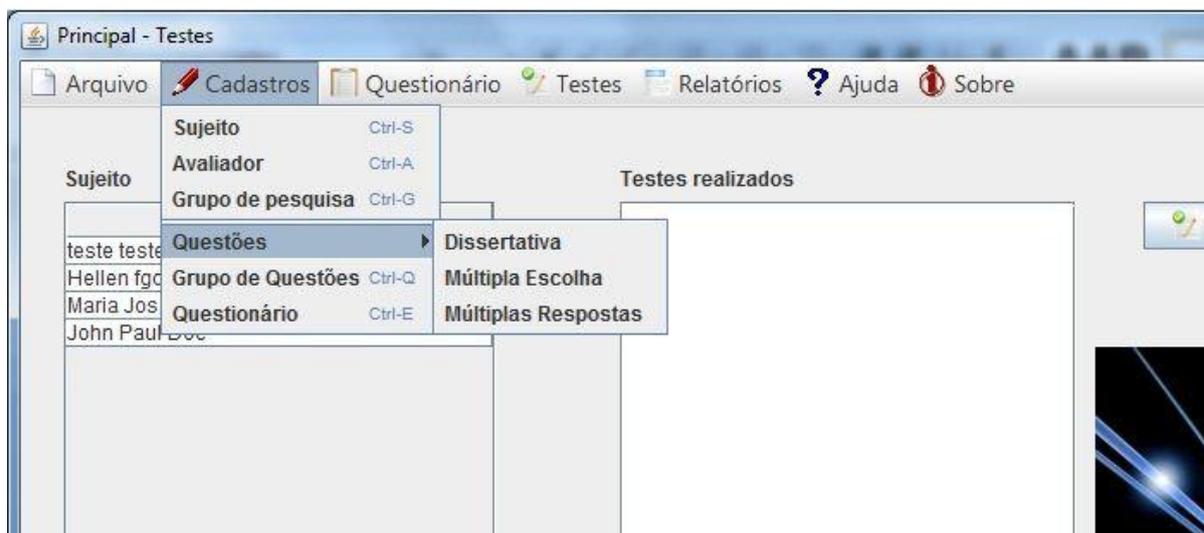


Figura 12 - Tela cadastro de questões e questionários

Para executar o questionário basta clicar na aba Questionário e selecionar Executar Questionário como mostrado na imagem abaixo. (Figura 13)

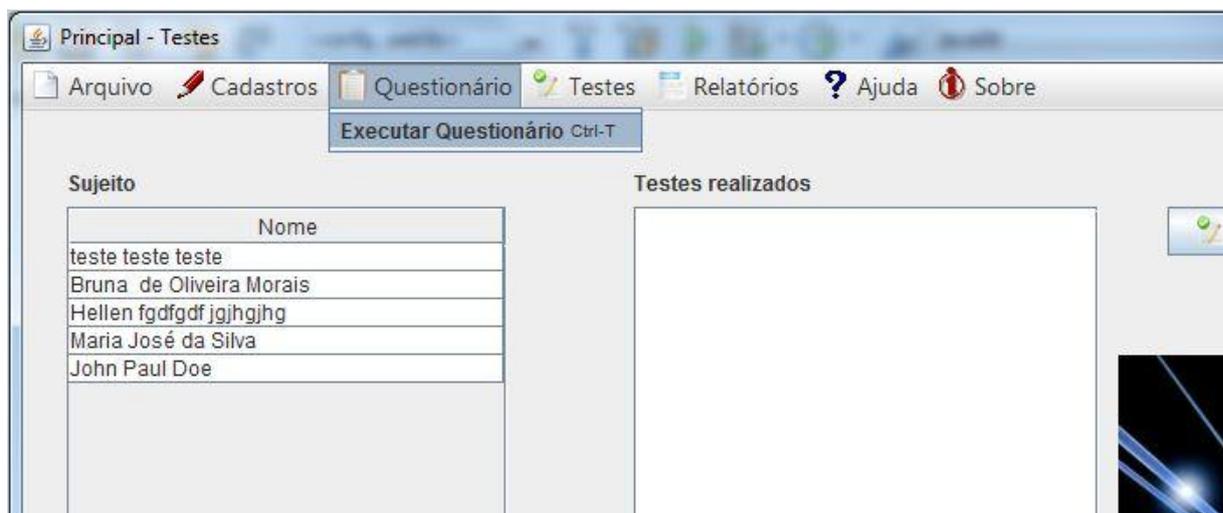


Figura 13 - Tela aba questionário

Ao escolher o tipo de teste a ser aplicado deverá ir até a aba Teste e optar pelo teste desejado, no caso o CPT é o válido para este trabalho. O caminho é mostrado na Figura 14.

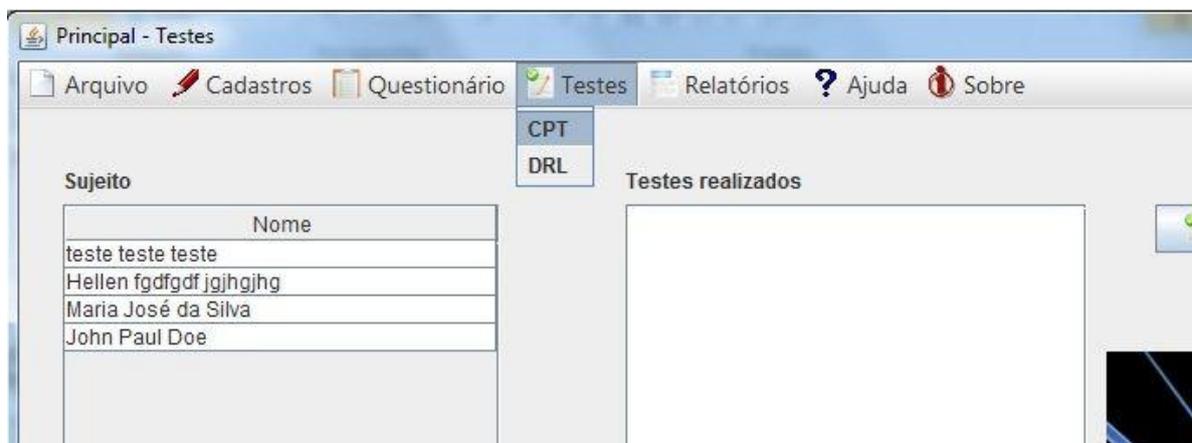


Figura 14 – Tela aba testes

Assim que sujeitos e avaliadores cadastrados, questionários respondidos e teste realizado poderão ser apresentados no relatório final contendo todas essas informações, para emitir basta clicar na aba Relatório e em Relatório Final como é possível visualizar na Figura 15.

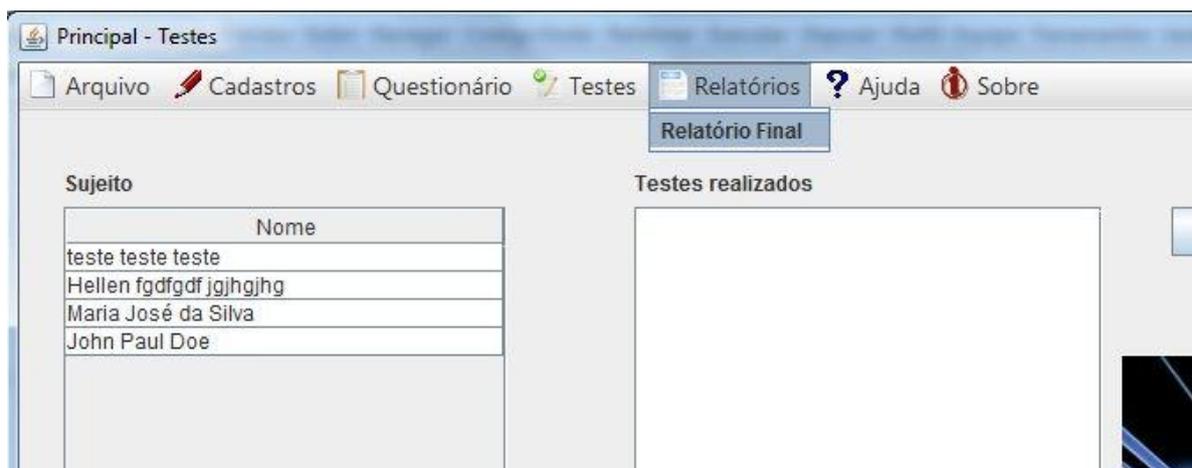


Figura 15 – Tela aba relatório

Foram criados tópicos para ajuda com o software em casos de dúvidas, a tela a seguir (Figura 16) ilustra como obter essa ajuda, que se encontra no caminho Ajuda, Tópicos de Ajuda.

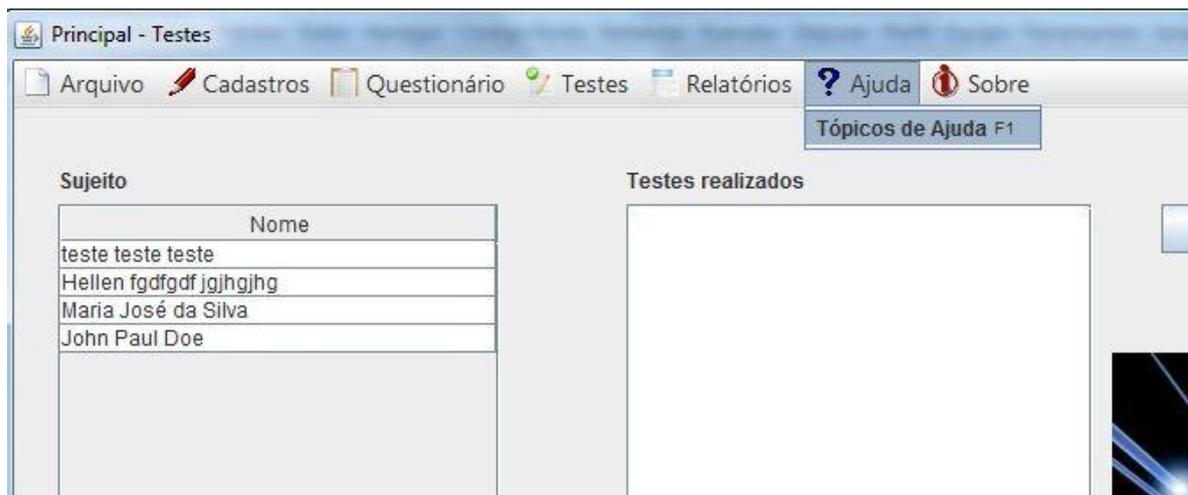


Figura 16 - Tela aba ajuda

Informações sobre o software são encontradas na aba Sobre. (Figura 17).

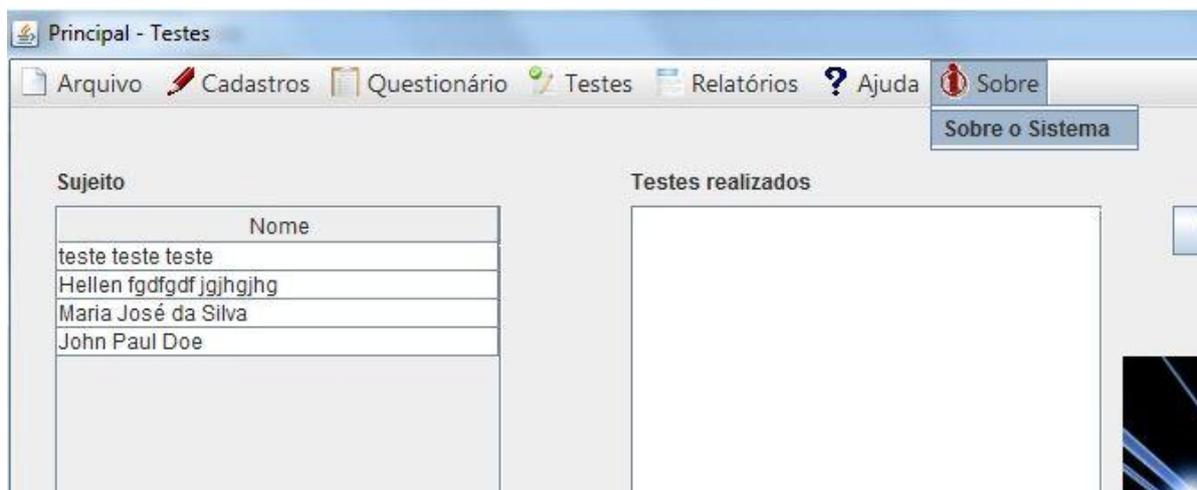


Figura 17 - Tela aba sobre

### 3.2.1 TELAS DE EXECUÇÃO DA INFRAESTRUTURA DO SISTEMA

Nesta seção serão ilustradas figuras da execução da infraestrutura para facilitar a compreensão do sistema, como serão exibidas as exportações em XML, os formulários para cadastros e execução do questionário e do teste CPT.

Será permitido exportar três tipos de dados no formato XML, para: sujeitos, avaliadores e questões, seguem abaixo exemplos extraídos do sistema.

Após ir à aba Arquivo, Extrair dados, é possível optar por quais dados, entre, Questão, Avaliador e Sujeito, na Figura 18 tem-se uma demonstração de como será se Questões for gerada em XML.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Questoes>
3    <Questao>
4      <titulo>Quais suas preferências musicais?</titulo>
5      <tipo>Múltiplas Respostas</tipo>
6      <opcao1>Rock</opcao1>
7      <opcao2>Clássico</opcao2>
8      <opcao3>Pop</opcao3>
9      <opcao4>Pagode</opcao4>
10     <opcao5>Axé</opcao5>
11     <opcao6>Gospel/Religioso</opcao6>
12     <opcao7>Reggae</opcao7>
13   </Questao>
14 </Questoes>
15

```

Figura 18 - Questões em XML

Se no caso a opção for extrair Avaliadores a Figura 19 exemplifica avaliadores gerados em XML.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Avaliadores>
3    <Avaliador>
4      <nome>John</nome>
5      <sobrenome>Doe</sobrenome>
6      <email>john@email</email>
7      <instituicao>UFJD</instituicao>
8      <observacao>Avaliador psicológico</observacao>
9    </Avaliador>
10 </Avaliadores>
11

```

Figura 19 - Avaliadores em XML

Para a extração de dados de Sujeitos em XML segue abaixo na Figura 20 um exemplo de um sujeito em XML.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Sujeitos>
3   <Sujeito>
4     <primeiroNome>John</primeiroNome>
5     <segundoNome>Paul</segundoNome>
6     <sobrenome>Doe</sobrenome>
7     <dataNasc>12/12/1080</dataNasc>
8     <sexo>Masculino</sexo>
9     <endereco>Rua dos Johns</endereco>
10    <cidade>São Paulo</cidade>
11    <estado>SP</estado>
12    <cep>987345-098</cep>
13    <foneResidencial>(23) 4567-2345</foneResidencial>
14    <foneTrabalho>(23) 4567-8907</foneTrabalho>
15    <celular>(23) 98070987</celular>
16    <email>john@email</email>
17    <austimo>não</austimo>
18    <epilepsia>não</epilepsia>
19    <depressao>não</depressao>
20  </Sujeito>
21 </Sujeitos>
```

Figura 20 - Sujeitos em XML

Na Figura 21, é possível visualizar o formulário de cadastro de sujeitos, que é útil para o armazenamento das informações do sujeito, assim como atrelar os testes de software ao cadastro.

Nome	Sobrenome	E-mail
teste	teste	teste@teste.com
Hellen	jjhgjhjg	
Hellen	jjhgjhjg	teste@uenp.edu.br
Maria	da Silva	mariajose@email.com

Figura 21 - Tela cadastros de sujeitos

Logo abaixo na Figura 22 pode-se visualizar a tela do formulário para cadastrar avaliadores, no qual são preenchidos para cadastrado e listados os existentes.

Nome	E-Mail
HellenF	hellen@email.com
João do Pulo	joao@teste
Teste	teste@ueno.edu.br
John	johndoe@email.com
Maria	maria@email.com
José	zesauo@email.com
Creuza	creuza@email.com

Figura 22 - Tela cadastros de avaliadores

Para qualquer alteração ou efetuado um novo cadastro recebe-se a mensagem com o sucesso do cadastro, como é possível visualizar na Figura 23.

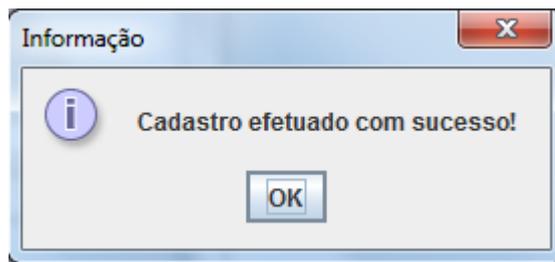


Figura 23 - Tela efetuado com sucesso

Se clicado em excluir algum dado, exibirá a mensagem de confirmação para apagar o respectivo dado, uma vez excluído não terá retorno desses dados.

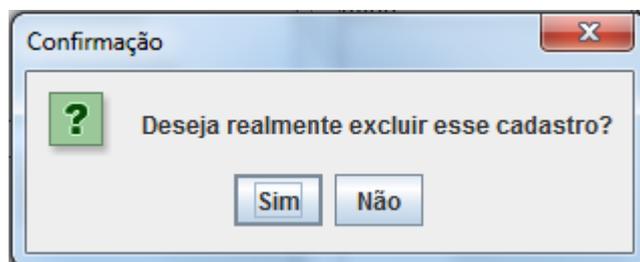


Figura 24 - Tela confirmação excluir

Se marcado a negação voltará para onde estava sem nenhuma ocorrência ser feita, já se marcado a afirmação a seguinte mensagem aparecerá e o dado será apagado do seu sistema.

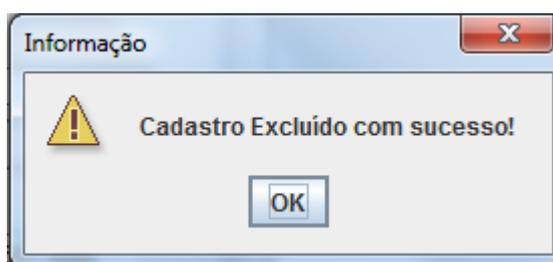


Figura 25 - Tela excluído com sucesso

A Figura 26 mostra a tela para cadastro de novos grupos com um clique na questão poderá inseri-la no grupo, com duplo clique poderá visualizar a questão como na Figura 27.

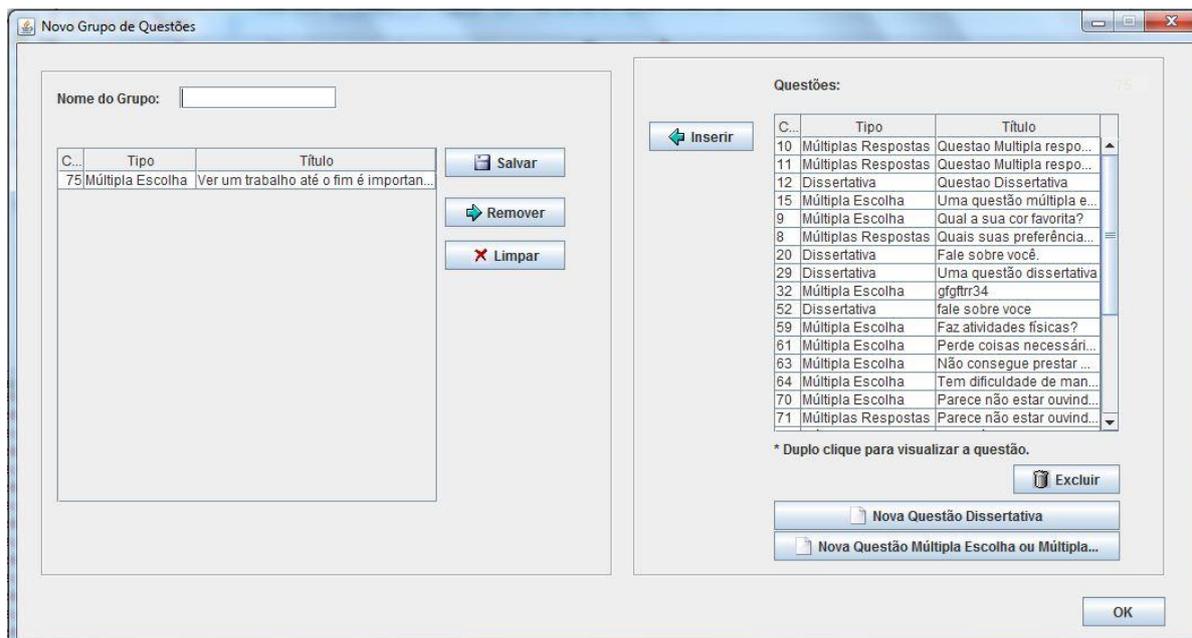


Figura 26 - Cadastro de grupo de questões

Quando dado duplo clique na questão do lado direito aparecerá uma exibição da respectiva questão, como exemplo abaixo.

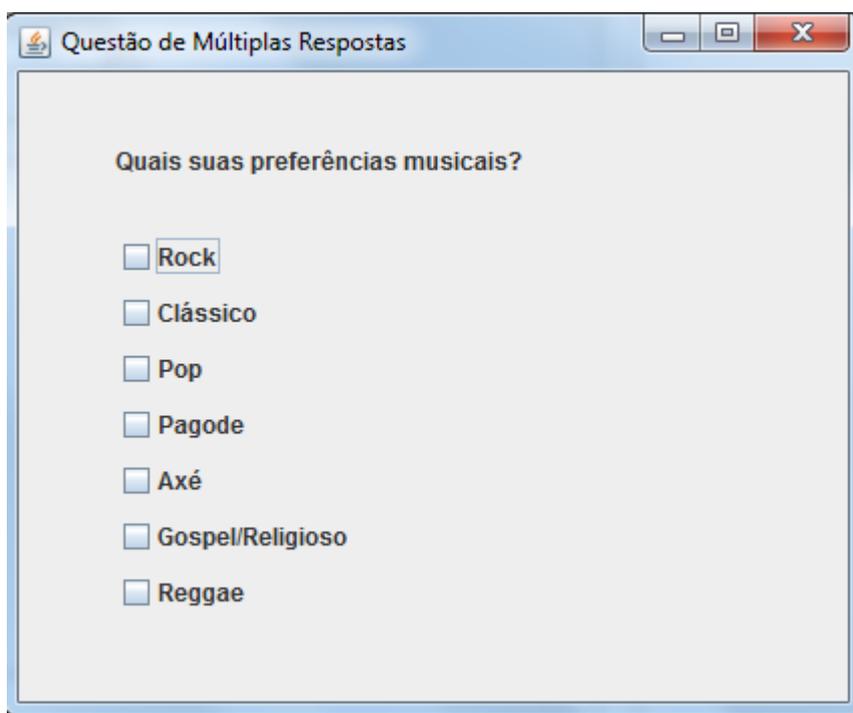


Figura 27 - Tela exibição Múltiplas Respostas

Se a questão for de múltipla escolha aparecerá em forma de 'radiobutton' como a seguir.

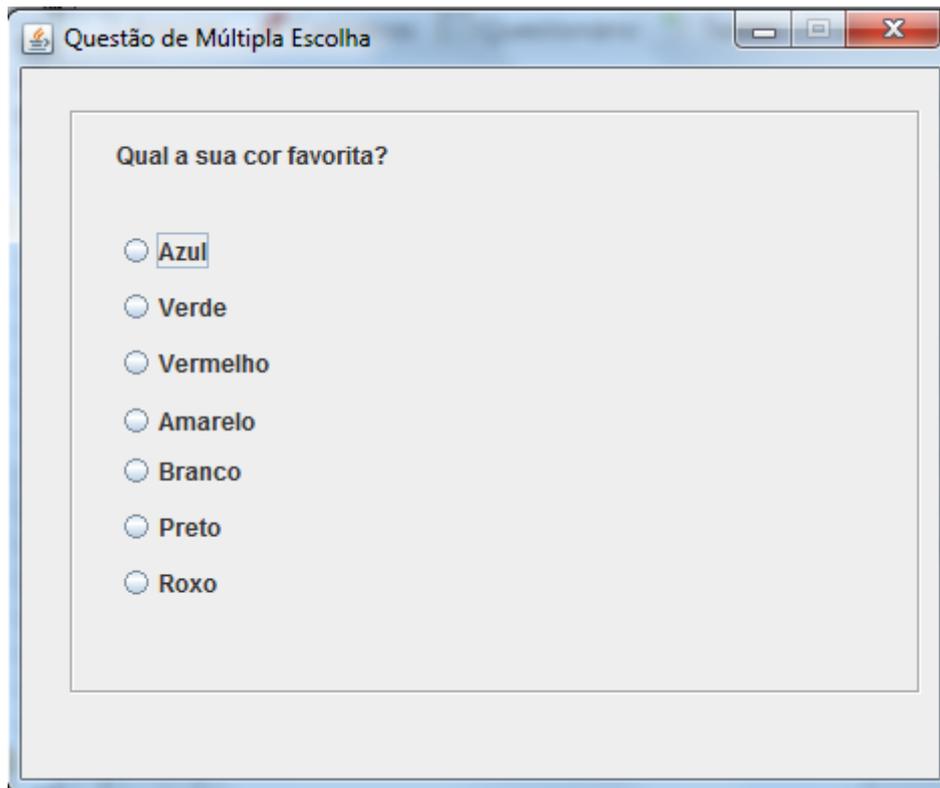


Figura 28 - Tela exibição Múltipla Escolha

Após optar pelo tipo de questão dissertativa, abrirá uma janela para esse cadastro, como mostrado na Figura 29.

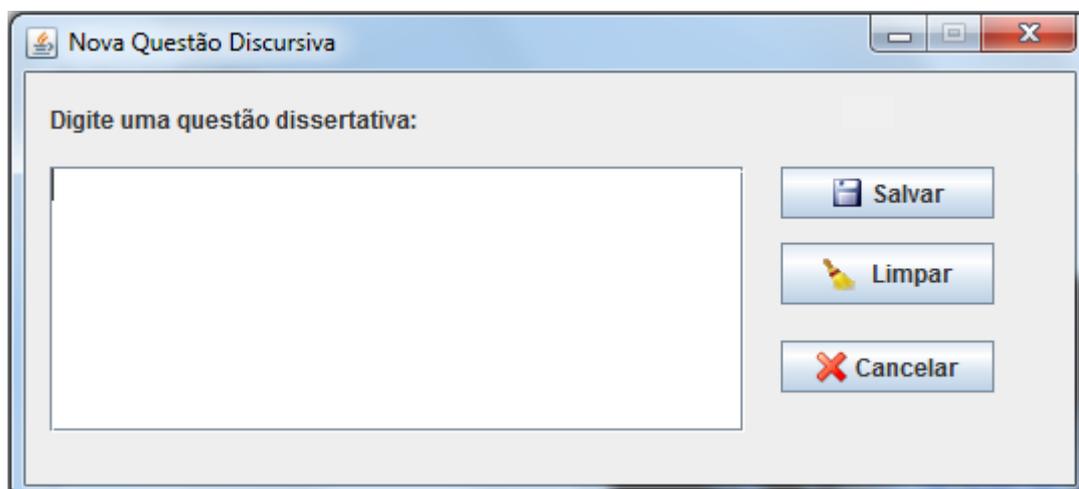


Figura 29 - Tela cadastro questão dissertativa

Para cadastrar questões múltiplas respostas ou múltipla escolha deverão ser escolhidas a quantidade de opções que ela terá como exibição abaixo.

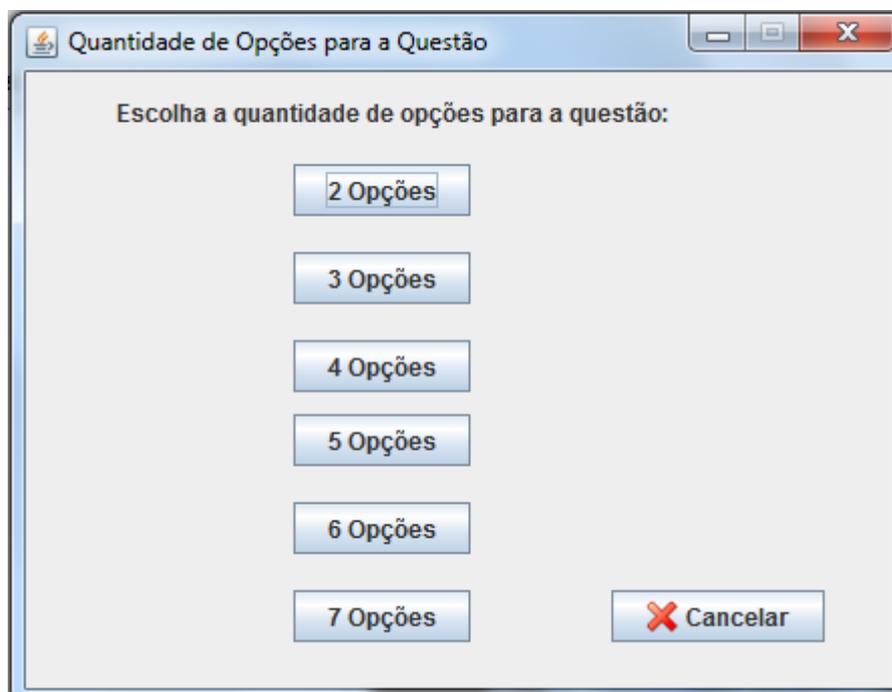
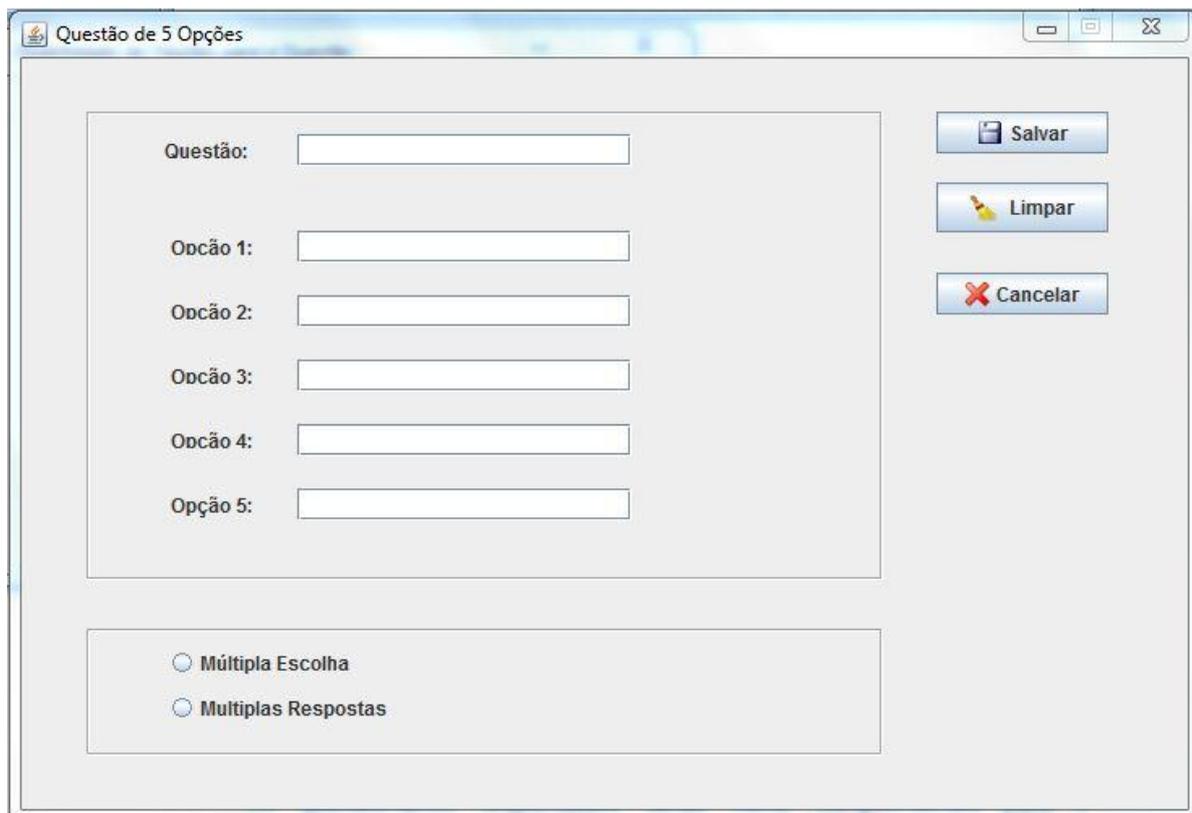


Figura 30 - Tela quantidade opções para questões.

Logo que escolhido a quantidade de opções abrirá uma contendo a quantidade de campos escolhida, sendo assim, quando preenchidas, identificar se é do tipo múltipla escolha ou múltiplas respostas. Abaixo um exemplo de uma questão com cinco opções.



The image shows a software window titled "Questão de 5 Opções". The window contains a form with the following elements:

- A label "Questão:" followed by a text input field.
- A label "Opção 1:" followed by a text input field.
- A label "Opção 2:" followed by a text input field.
- A label "Opção 3:" followed by a text input field.
- A label "Opção 4:" followed by a text input field.
- A label "Opção 5:" followed by a text input field.
- Three buttons on the right side: "Salvar" (with a floppy disk icon), "Limpar" (with a trash can icon), and "Cancelar" (with a red X icon).
- At the bottom, a group box containing two radio buttons: "Múltipla Escolha" and "Múltiplas Respostas".

Figura 31 - Tela exibição para questão com cinco opções

Para criar um novo questionário, serão exibidos os grupos existentes, onde é possível criar um novo grupo, clicando em 'novo grupo', e ao clicar no grupo exibe todas as questões pertencentes, e se desejar incluir no questionário basta clicar no botão inserir, é obrigatório nomear o questionário, pois por meio dele que fará a execução do mesmo. Segue uma imagem de cadastro de questionários. (Figura 32).

**Questionários**

Cód...	Título do Questionário
1	Primeiro Questionario
76	seg
77	terceiro Questionario
78	Quarto Questionario
79	quinto
80	sexto questionario
93	plkjho
110	Questionario 9870
111	Questionario 987098
112	questionario 098
113	questiooo
116	oi
119	oie

Responder  
Excluir

**Nome do Questionário \***

Nome do Questionário

Novo Questionário

Salvar  
Remover  
Limpar

**Grupo de Questões**

Nome do Grupo

Patologia

Novo Grupo  
Excluir

**Questões**

Incluir

Figura 32 - Tela cadastro de questionários.

Os questionários respondidos serão como complementos para o resultado final, ao gerar o relatório de resultados virão incluso as respostas dos questionários.

Para auxiliar na utilização do sistema, foi criado tópicos de ajuda, no qual são descritos os passos para cada atividade. (Figura 33).

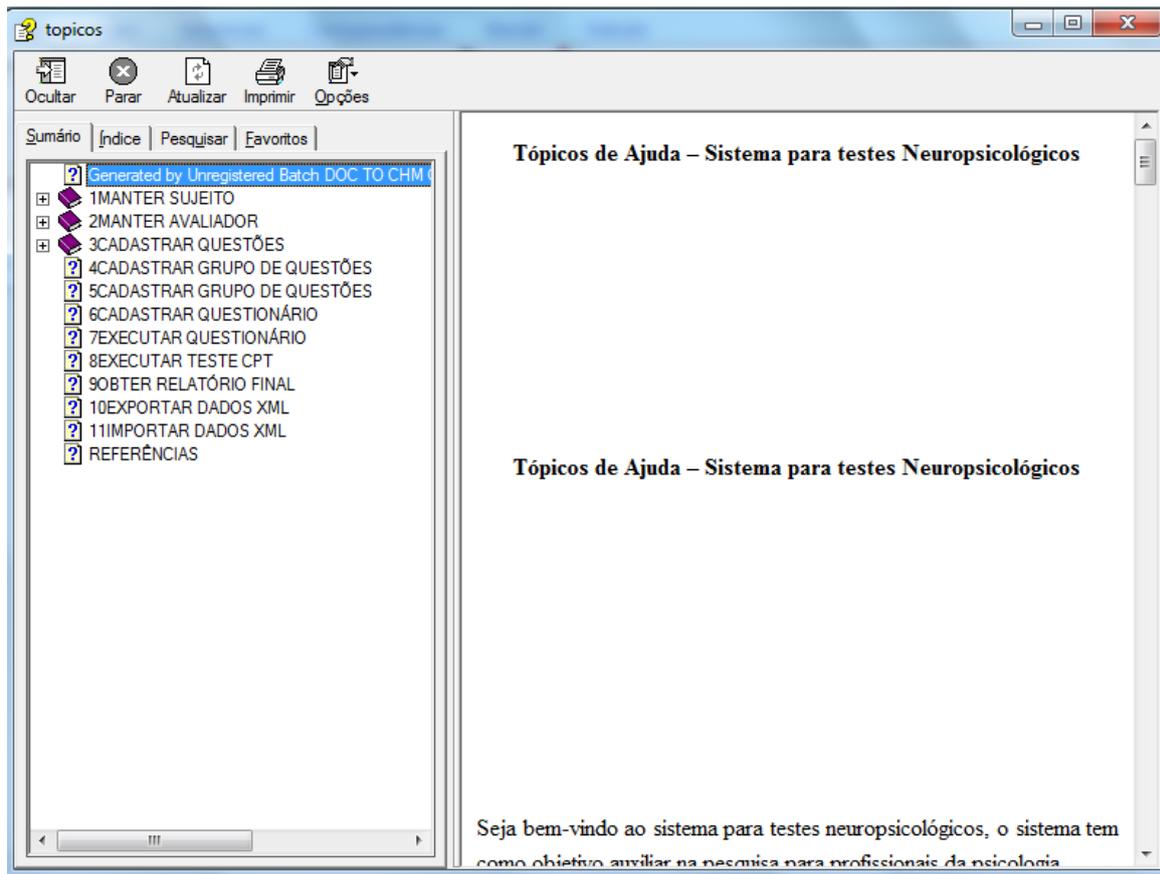


Figura 33 – Tela dos tópicos de ajuda

Ao clicar na aba Sobre, opção Sobre o Sistema para obter informações aparecerá uma janela como na tela abaixo. (Figura 34)

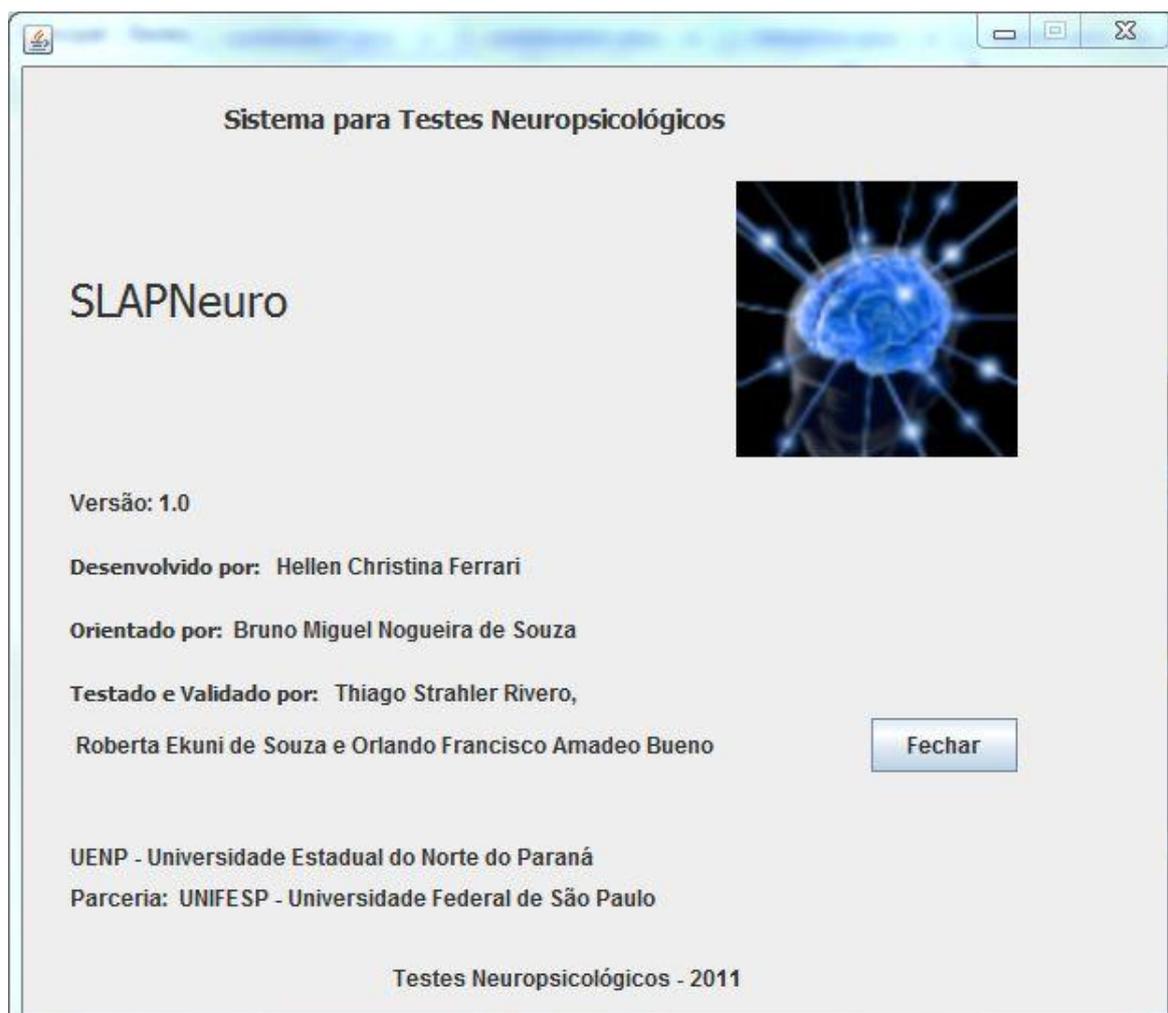


Figura 34 – Tela sobre o sistema

### 3.2.2 Execução do Questionário

Depois de cadastrados as questões, grupos e questionários, na opção da aba questionário aparecerá uma tela com os questionários existentes para optar a execução, listando as questões no quadro abaixo, como mostrado na Figura 35.

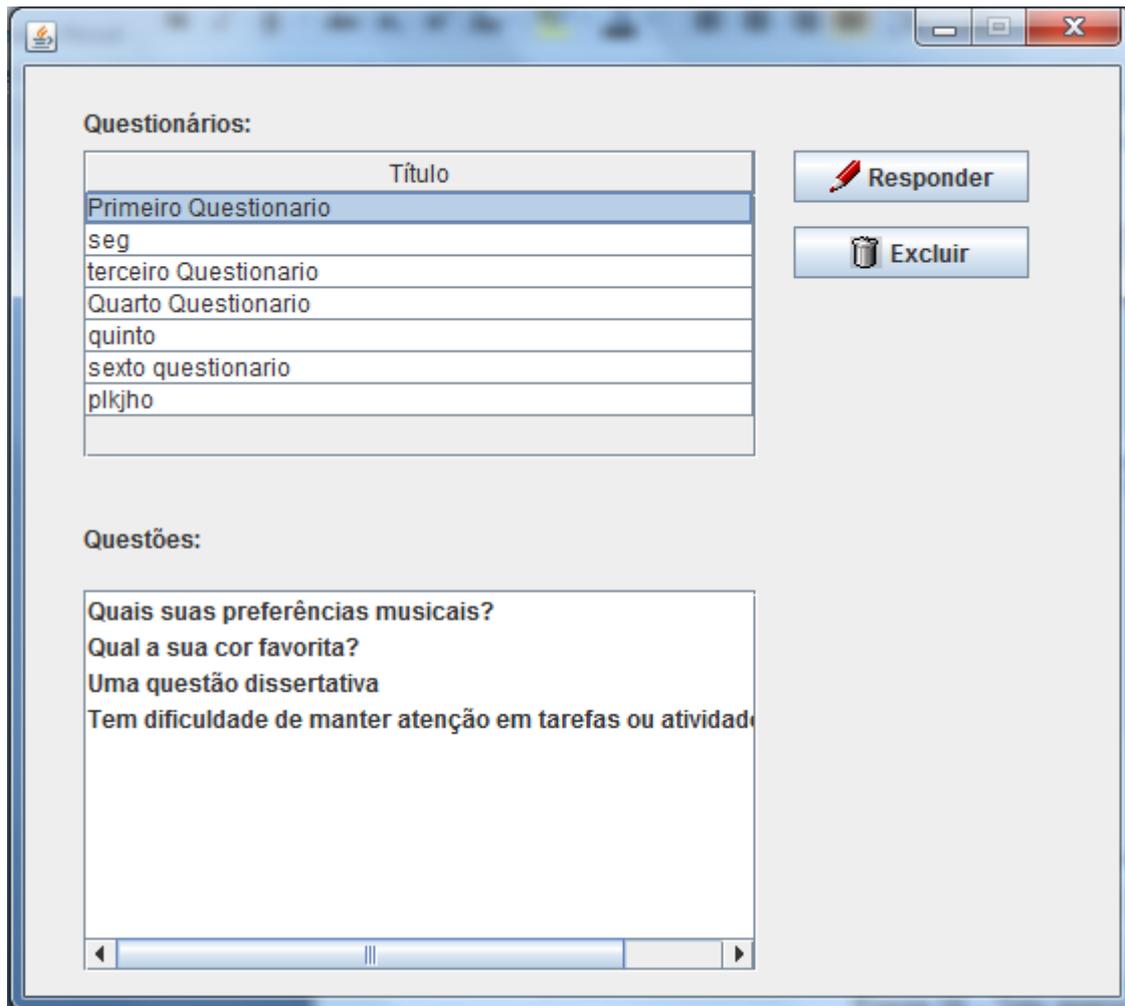
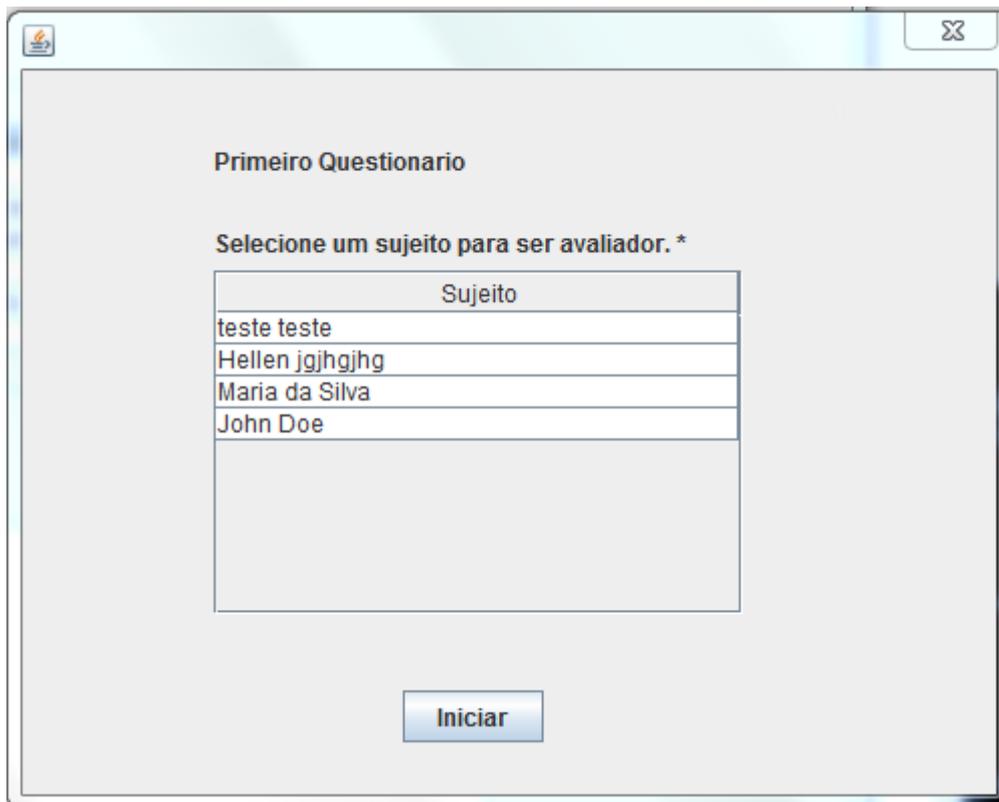


Figura 35 – Tela Lista de Questionários

A partir de escolhido o questionário clicando em responder, começará a execução do mesmo.

Aparecendo o título do questionário e uma lista de sujeito que é obrigatória a escolha, e inicia as questões. As telas de execução do questionário foram desenvolvidas como telas de diálogos, e com a propriedade "modal", sendo assim, estão em primeiro plano e só abre a próxima questão clicando no botão próximo, caso contrário não será gravada a resposta no banco de dados e não poderá ser feita ações em outras janelas.



Primeiro Questionario

Selecione um sujeito para ser avaliador. \*

Sujeito
teste teste
Hellen jgjhghg
Maria da Silva
John Doe

Iniciar

Figura 36 – Tela execução do questionário – Início

Na figura abaixo é mostrado um exemplo de como as questões aparecerão no momento da execução do questionário.

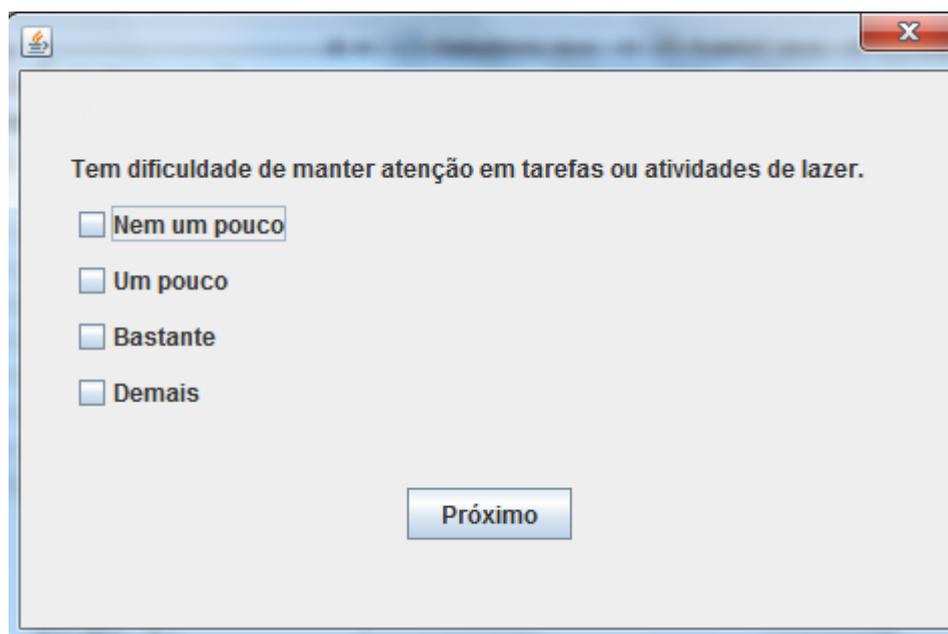


Figura 37 – Tela execução questionário – Questões

Terminando as questões contidas no questionário aparecerá uma tela indicando que chegou ao fim e as respostas foram gravadas.

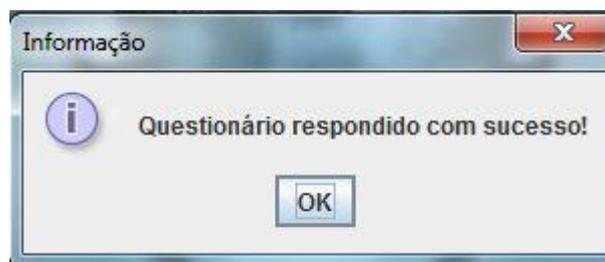


Figura 38 – Mensagem Final de Execução de Questionário

### 3.3 EXECUÇÃO DO TESTE CPT

No caso do teste CPT foi desenvolvido da seguinte forma: Pressionar barra de espaço sempre que uma letra aparecer na tela (324 vezes). Inibir sua resposta quando a letra 'X' aparecer na tela (36 vezes). Na implementação foi dividido em 6 blocos de estímulos, cada sub-bloco têm 20 tentativas. Cada letra apresentada é de 250 milissegundos. A aparição de cada letra varia entre 1, 2 e 4 segundos, sendo aleatoriamente. O texto foi desenvolvido em *fullscreen*, sendo assim, ocupando todo o monitor, para que a atenção seja focada apenas nos estímulos.

Ao iniciar o teste, há a primeira parte para treino do sujeito, no qual ele irá se adaptar com o sistema, sendo apenas com 20 estímulos, e resultando na tela sua quantidade de omissões e comissões que é calculada ao omitir um não-estímulo ou respondendo um estímulo.

Logo então começa o teste, obtendo no final medidas para o cálculo do psicólogo como: tempo de reação de cada estímulo, esse cálculo é a média de todos os seus tempos, detectabilidade é como o sujeito detectou a presença ou não de um estímulo, esse resultado varia de 0 a 1, sendo 1 o perfeito para detecção de sinais. A mudança é a média do tempo entre os blocos, sendo o bloco dividido por 60 estímulos. Segue uma imagem para melhor visualizar o teste. (Figura 39).

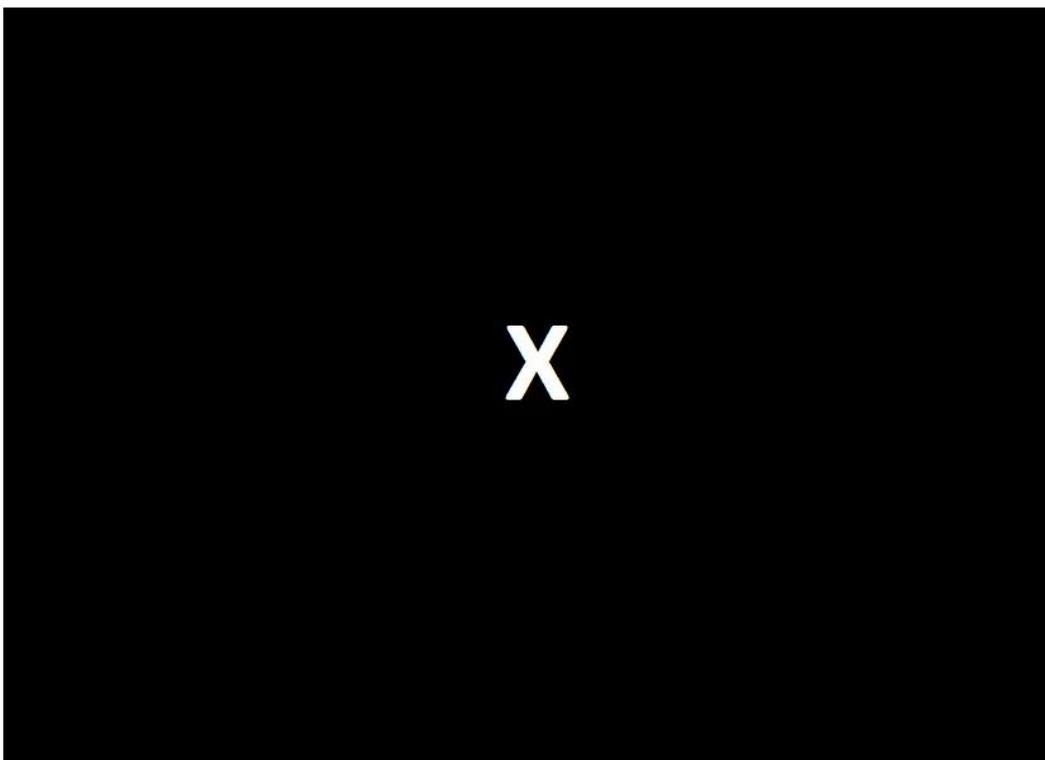


Figura 39 - Tela estímulo X do teste CPT.

Esta tela indica que a barra de espaço não deve ser apertada, caso seja, é contabilizado como uma comissão.

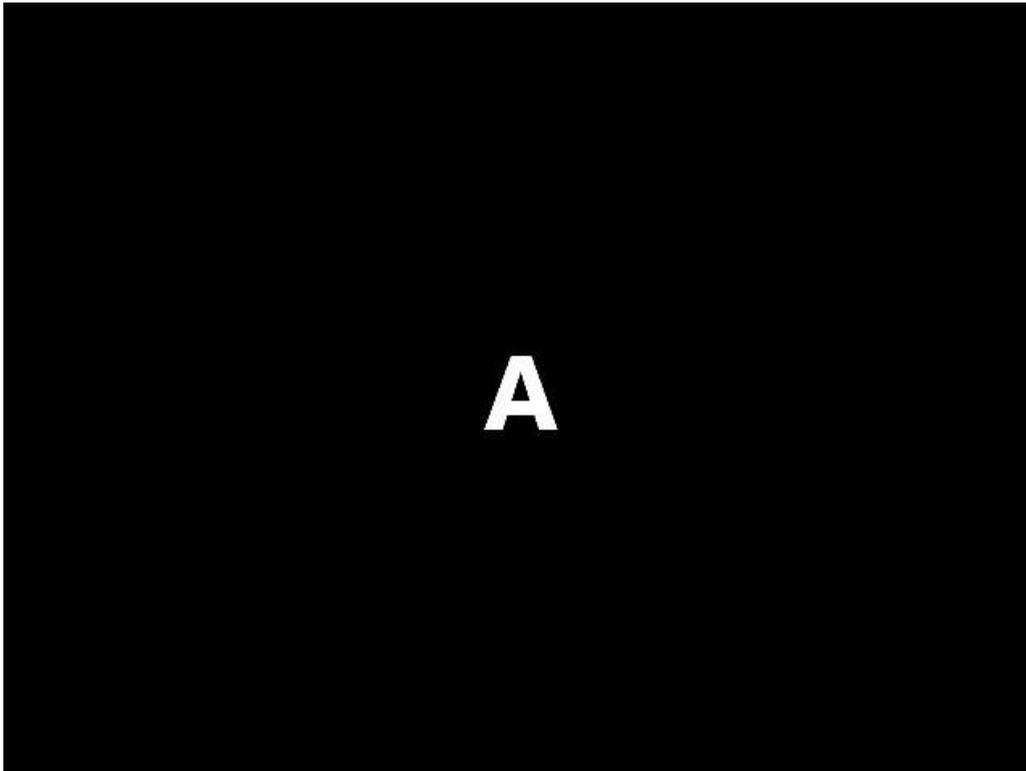


Figura 40 – Tela estímulo teste CPT

Esta tela indica que a barra de espaço deve ser apertada, caso não seja, é contabilizado como uma omissão.

### 3.4 CONSIDERAÇÕES FINAIS

Após responder os questionários e aplicar o teste CPT, o sistema apresenta um relatório final que é composto pelo sujeito que foi optado pelas respostas, pelo avaliador que aplicou o teste, os resultados do teste CPT, sendo suas 15 medidas e perguntas e respostas do questionário aplicado, na Figura 41 exemplifica um relatório emitido.

## Testes Neuropsicológicos

---

**Sujeito:**  
 Código: 15      Data Nasc.: 23/09/1976      Sexo: Feminino      Data null  
 Nome: Maria José da Silva  
 Endereço: Rua das Flores  
 Cidade: Ponta Grossa      Estado: PR      CEP: 34567-890  
 Telefone: (44)3545-4555      Celular: ( ) -  
 E-mail: mariajose@email.com  
 Transtorno: nao      Epilepsia: nao      Depressão: sim      Autismo: nao  
 Obs:

---

**Avaliador:**  
 Nome: Creuza Silva  
 E-mail: creuza@email.com      Instituição: UENP  
 Obs.: asdf

---

**Medidas CPT**      Erros:

Omissão: 4	% Omissão: 0	Comissão: 2	% Comissão: 0.0
------------	--------------	-------------	-----------------

---

**Tempo de Reação (Hit Reaction Time):**

Tempo Reação: 456	Desvio Padrão TR: 59.0	Variabilidade: 123.0
-------------------	------------------------	----------------------

---

**Teoria da Detecção de Sinais (Signal Detections Theory)**

Estilo Resposta (b): 4.0	Detectabilidade: 0.6	Perseverações: 0	% Perseverações:
--------------------------	----------------------	------------------	------------------

---

**Resultados por Bloco:**

Mudança TR: 300	Desvio Padrão Mudança TR: 200
-----------------	-------------------------------

---

**Resultados por ISIs:**

Mudança ISI: -1	Desvio Padrão Mudança ISI: 1
-----------------	------------------------------

---

**Questionário:**

Questão:	Resposta:
Quais suas preferências musicais?	resp1
Tem dificuldade de manter atenção em tarefas ou atividades de lazer.	Demais
Uma questão dissertativa	testessss
Qual a sua cor favorita?	opcao1
Uma questão dissertativa	resposta dissertativa
Qual a sua cor favorita?	Azul

---

Sexta-feira 18 Page 1 of 2

Figura 41 – Relatório Final

Depois dos testes e questionários, os resultados estarão disponíveis para que o profissional possa fazer o diagnóstico do sujeito, obter o grau de transtorno de déficit de atenção e hiperatividade do sujeito.

## 4. CONCLUSÃO

O objetivo deste trabalho foi propor e criar um sistema para facilitar o diagnóstico dos especialistas da neurociência. Foi criada uma infraestrutura básica para coletar informações dos sujeitos que realizarão os testes e um teste sistematizado para avaliação de atenção, o Teste Contínuo de Desempenho (*Continuous Performance Test - CPT*). Extraíndo relatórios dos testes e questionários do banco de dados da aplicação. Durante o desenvolvimento do trabalho, ocorreram algumas restrições de desenvolvimento que dificultaram o bom andamento do projeto, desta forma, houveram algumas restrições no que diz respeito ao desenvolvimento do layout, para deixá-lo com a interface mais agradável e melhorar sua usabilidade, houve a tentativa do uso do framework OpenSwing, mas causaram conflitos com bibliotecas, portanto foi retirado da construção, dificuldades também ao utilizar o banco de dados embarcado, foi necessário a pesquisa de materiais sobre sua utilização e materiais para melhorar o conhecimento do *Hibernate Annotations*.

Apesar das dificuldades de implementação, o software possui suporte para acoplar novos testes neuropsicológicos que deverão ser implementados utilizando como base a infraestrutura do sistema para que os questionários possam ser aproveitados para o novo teste.

Uma das grandes vantagens de utilização deste software é que com ele é possível desenvolver testes e ao mesmo tempo armazenar questionários dos sujeitos a serem avaliados, além disso, o software é de caráter livre, e pode ser incrementado e utilizado sem a necessidade de custos adicionais. Como trabalhos futuros deverão ser realizados testes para verificar a eficácia do software no que diz respeito principalmente a avaliação do déficit de atenção (CPT) e desenvolver novos testes neuropsicológicos visando a avaliação e reabilitação dos pacientes, como por exemplo o DRL – *Diferencial Reinforcement of Low rate* (Diferenciamento referencial de baixa taxa de resposta) e treinos de memória, sendo que tais testes podem ser acoplados a

infraestrutura já desenvolvida. E podendo também como trabalhos futuros fazer análise e mineração de dados.

Desta forma, o software criado dá suporte computacional para aplicação de testes possibilitando a avaliação neuropsicológica e o armazenamento de informações dos sujeitos. Com as informações armazenadas, fica possível melhor análise dos resultados, impactando positivamente no desenvolvimento da pesquisa do profissional envolvido.

## REFERÊNCIAS

ALAN. (1997). **How to use MVC** <<http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>>

BIEDERMAN, J. (1998). Attention-deficit/hyperactivity disorder: **A life-span perspective**. *Journal of Clinical Psychiatry*, 59 (Suppl. 7), 4–16.

CLÁUSULA GRANT - Documentação PostgreSQL.<  
<http://www.postgresql.org/docs/8.1/static/sql-grant.html>>

Conners' Continuous Performance Test (CPT II) - **Conners' Kiddie Continuous Performance Test (K-CPT)** -

<[http://portal.wpspublish.com/portal/page?\\_pageid=53,252579&\\_dad=portal&\\_schema=PORTAL](http://portal.wpspublish.com/portal/page?_pageid=53,252579&_dad=portal&_schema=PORTAL)>

DEITEL, H.M. & DEITEL, P.J. **Java: como programar**. 6<sup>a</sup>.ed. São Paulo: Pearson Prentice Hall, 2005.

Documentation – Hibernate – Jboss Community -  
<<http://www.hibernate.org/docs>>

EYSENCK, Michael W.; KEANE, Mark T. Reconhecimento de objetos. In: **Manual de Psicologia Cognitiva**. 5 ed. Porto Alegre: Artmed, 2007. cap. 3, p. 74-113.

HILLEGAS, Richard (2007) -< <http://wiki.apache.org/db-derby/>>

JASPERFORGE – IReport - <<http://jasperforge.org/projects/ireport>>

JDK 6 Swing (Java Foundation Classes (JFC)) -  
<<http://download.oracle.com/javase/6/docs/technotes/guides/swing/>>

LURIA, A.R. **Fundamentos de Neuropsicologia**. Tradução de Juarez Aranha Ricardo. Rio de Janeiro: Livros Técnicos e Científicos; São Paulo: EDUSP. 1981.

LYON Jenny (2009) - CPT II - *IPS Information Circular* -  
<<http://www.devdis.com/conners2.html>>

NEATBEANS IDE – Features Site Oficial -  
<<http://netbeans.org/features/index.html>>

O'Laughlin, E.M., & Murphy, M.J. (2000) Use of computerized continuous performance tasks for assessment of ADHD: A guide for practitioners. *Independent Practitioner*.

Persistência usando hibernate  
<<http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/hibernate/hibernate.htm>>

RESOLUÇÃO nº 196, de 10 de outubro de 1996. Disponível em  
<[http://dtr2004.saude.gov.br/susdeaz/legislacao/legislacao\\_det.php?co\\_legislacao=107](http://dtr2004.saude.gov.br/susdeaz/legislacao/legislacao_det.php?co_legislacao=107)>.

RIVERO, Thiago Strahler. **Análise do desempenho dos índices do *Conners' Continuous Performance Test (CCPT)* em uma amostra de adolescentes com TDAH.** Dissertação de Mestrado (Mestrado em Neuropsicologia) – Universidade Federal de São Paulo, 2010.

RIZZUTTI, S., *et al.*(2008). Clinical and neuropsychological markers in the diagnosis of attention deficit hyperactivity disorders (ADHD). *Arquivos de Neuropsiquiatria*, 66(4), 821-827.

ROOT, R.W., & RESNICK, R.J. (2003). An update on the diagnosis and treatment of Attention-Deficit/Hyperactive Disorder in Children. *Professional Psychology: Research and Practice*, 24 (1), 34-41.

ROSVOLD, H. E., *et al.* (1956).A continuous performance test of brain damage.*Journal of Consulting Psychology*, 20, 343-350.

SKINNER, B. F. *The behavior of organisms: an experimental analysis.* Oxford, England: Appleton-Century. 1938.

WHO - World Health Organization – Global Burden of Neurological Disorders: Estimates and Projections. Disponível em  
<[http://www.who.int/mental\\_health/neurology/chapter\\_2\\_neuro\\_disorders\\_public\\_h\\_challenges.pdf](http://www.who.int/mental_health/neurology/chapter_2_neuro_disorders_public_h_challenges.pdf)>.

XEROX PARC. (1978) <<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>>

ZIKOPOULO, Paul C.; BAKLARZ, George; SCOTT, Dan (November 6, 2005), *Apache Derby -- Off to the Races: Includes Details of IBM Cloudscape* (First ed.), IBM Press, pp. 600, ISBN 0131855255