

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

SISTEMA MÓVEL DE INDICAÇÃO DE TRANSTORNOS
MENTAIS UTILIZANDO A TÉCNICA DE RACIOCÍNIO
BASEADO EM CASOS

FELIPPE AUGUSTO MACHADO LOPES DO PRADO

BLUMENAU
2013

2013/2-13

FELIPPE AUGUSTO MACHADO LOPES DO PRADO

**SISTEMA MÓVEL DE INDICAÇÃO DE TRANSTORNOS
MENTAIS UTILIZANDO A TÉCNICA DE RACIOCÍNIO
BASEADO EM CASOS**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Sistemas
de Informação— Bacharelado.

Prof. Jacques Robert Heckmann, Mestre – Orientador

**BLUMENAU
2013**

2013/2-13

**SISTEMA MÓVEL DE INDICAÇÃO DE TRANSTORNOS
MENTAIS UTILIZANDO A TÉCNICA DE RACIOCÍNIO
BASEADO EM CASOS**

Por

FELIPPE AUGUSTO MACHADO LOPES DO PRADO

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Jacques Robert Heckmann, Mestre – Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos, Doutor – FURB

Membro: _____
Prof. Marcel Hugo, Mestre – FURB

Blumenau, 09 de dezembro de 2013.

Dedico este trabalho a todos os amigos e familiares, especialmente aqueles que me ajudaram diretamente na realização deste.

AGRADECIMENTOS

A Deus, pelo seu imenso amor e graça.

À minha família, que mesmo longe, sempre esteve presente.

Aos meus amigos, pelos empurrões e cobranças.

Ao meu orientador, Jacques Robert Heckmann, por ter acreditado na conclusão deste trabalho.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

Ame o teu ofício como uma religião, respeite suas convicções e as pratique de verdade, mesmo quando não tiver ninguém olhando. Milagres acontecem quando a gente vai à luta.

Sérgio Vaz

RESUMO

Este trabalho apresenta um aplicativo para dispositivo móvel que utiliza da técnica de Inteligência Artificial conhecida como Raciocínio Baseado em Casos para a indicação dos transtornos mentais mais prováveis com base nas diretrizes encontradas no Manual Diagnóstico e Estatístico de Transtornos Mentais. Buscou-se inovar na forma de identificação de transtornos e como uma ferramenta *mobile* pode ser utilizada de forma eficaz neste âmbito da saúde. Para o desenvolvimento do aplicativo utilizou-se a plataforma Android consumindo um *web service* hospedado em um serviço Apache Tomcat utilizando o *framework* Axis2, ambos desenvolvidos em linguagem de programação Java, e utilizando banco de dados MySQL. Como resultado obteve-se um aplicativo cujo mecanismo de identificação de transtornos reduz o tempo gasto para o estabelecimento de um diagnóstico.

Palavras-chave: Transtornos Mentais. Raciocínio Baseado em Casos. Dispositivos móveis. Inteligência artificial.

ABSTRACT

This work presents a mobile application that applies the Artificial Intelligence method known as Case-Based Reasoning to identify the most likely mental illnesses utilizing the premises found in the Diagnostic and Statistical Manual of Mental Disorders. It was look for innovation in how to identify the mental illness and how a mobile tool can effectively fit in this field of healthcare. For development it was used the Android platform consuming a web service hosted in an Apache Tomcat server utilizing Axis2 as framework, both using Java programming language, and MySQL database. The result was a system which identification mechanism of mental illness reduces the time spent in the search of the medical diagnosis.

Key-words: Mental illnesses. Case-Based Reasoning. Mobile Application. Artificial Intelligence.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 - Processo do RBC | 20 |
| Figura 2 - Informações usadas..... | 26 |
| Figura 3 - Tela de resultado da consulta de uma pesquisa com palavras-chave. | 27 |
| Figura 4 - Tela de definições de casos no CBR-Works 4..... | 28 |
| Figura 5 - Diagrama de Casos de Uso | 31 |
| Figura 6 - Diagrama de atividade da solicitação e processamento da requisição..... | 33 |
| Figura 7 - Diagrama de Classes..... | 34 |
| Figura 8 - Código fonte da criação da tela de <i>login</i> | 36 |
| Figura 9 - Código fonte do ciclo de vida recuperação..... | 37 |
| Figura 10 - Código fonte da inserção de um novo transtorno | 38 |
| Figura 11 - Código fonte da criação de uma requisição ao <i>web service</i> | 39 |
| Figura 12 - Tela de <i>login</i> | 40 |
| Figura 13 - Tela principal | 41 |
| Figura 14 - Tela de configuração de conta | 42 |
| Figura 15 - Tela de consulta de pacientes..... | 43 |
| Figura 16 - Tela de cadastro de pacientes | 43 |
| Figura 17 - Tela de consulta de transtornos..... | 44 |
| Figura 18 - Tela de cadastro de transtornos..... | 45 |
| Figura 19 - Tela de consulta de consultas | 46 |
| Figura 20 - Tela de cadastro de consultas | 46 |
| Figura 21 - Código de inicialização de casos em memória | 47 |
| Figura 22 - Tela de envio de pesquisa | 48 |
| Figura 23 - Parte do código do <i>SnowBall</i> | 49 |
| Figura 24 - Tela de seleção de transtorno..... | 50 |
| Figura 25 - Tela de seleção de transtorno (Criar novo caso)..... | 50 |
| Figura 26 - Código que procura o caso originador..... | 51 |
| Figura 27 - Respostas referente à indicação de transtornos pelo aplicativo..... | 54 |
| Figura 28 - Questionário referente ao grau de aceitação das funcionalidades | 54 |

LISTA DE QUADROS

| | |
|---|----|
| Quadro 1 - Fórmula de <i>Term Frequency</i> | 23 |
| Quadro 2 - Fórmula de <i>Inverse Document Frequency</i> | 24 |
| Quadro 3 - Fórmula de pontuação TF-IDF..... | 24 |
| Quadro 4 - Requisitos funcionais | 30 |
| Quadro 5 - Requisitos não funcionais | 31 |
| Quadro 6 - Comparativo das principais funções dos trabalhos correlatos | 53 |
| Quadro 7 - Descrição dos casos de uso | 60 |
| Quadro 8 - apresenta o dicionário de dados da tabela "medicos" | 63 |
| Quadro 9 - apresenta o dicionário de dados da tabela "pacientes" | 63 |
| Quadro 10 – apresenta o dicionário de dados da tabela “transtornospaciente” | 64 |
| Quadro 11 - apresenta o dicionário de dados da tabela “transtornos” | 65 |
| Quadro 12 - <i>Stop words</i> escolhidas | 66 |

LISTA DE SIGLAS

APA - Associação Psiquiátrica dos Estados Unidos

ADT - *Android Developer Tools*

CID - Classificação Internacional de Doenças

DSM - Manual Diagnóstico e Estatístico de Transtornos Mentais

IE - Extração da informação

IA - Inteligência Artificial

ICD-9-MC – *International Classification of Diseases Ninth Revision, Clinical Modification*

IDF - *Inverse Document Frequency*

IR - Recuperação da informação

JSON - *JavaScript Object Notation*

RBC - Raciocínio Baseado em Casos

SGBD - Sistema de Gerenciador de Banco de dados

SOAP - *Simple Object Access Protocol*

TF - *Term Frequency*

UNIFESP – Universidade Federal de São Paulo

UML - *Unified Modeling Language*

VSM - Modelo de Espaço Vetorial

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO..... | 12 |
| 1.1 OBJETIVOS DO TRABALHO | 13 |
| 1.2 ESTRUTURA DO TRABALHO | 13 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 TRANSTORNOS MENTAIS | 14 |
| 2.1.1 Causas | 15 |
| 2.1.2 Diagnóstico | 15 |
| 2.1.3 Tratamento | 16 |
| 2.1.4 Manual Diagnóstico e Estatístico de Transtornos Mentais | 17 |
| 2.2 RACICÍONIO BASEADO EM CASOS..... | 19 |
| 2.2.1 Vantagens do RBC..... | 21 |
| 2.2.2 Framework | 21 |
| 2.2.3 <i>Textual Case-based Reasoning</i> | 22 |
| 2.2.4 Modelo Espacial Vetorial..... | 23 |
| 2.3 ANDROID..... | 25 |
| 2.4 TRABALHOS CORRELATOS | 25 |
| 3 DESENVOLVIMENTO | 29 |
| 3.1 LEVANTAMENTO DE INFORMAÇÕES | 29 |
| 3.2 ESPECIFICAÇÃO | 30 |
| 3.2.1 Requisitos funcionais do sistema | 30 |
| 3.2.2 Requisitos não funcionais | 30 |
| 3.2.3 Diagramas de casos de uso..... | 31 |
| 3.2.4 Diagrama de atividades | 32 |
| 3.2.5 Modelo conceitual da base de dados | 34 |
| 3.3 IMPLEMENTAÇÃO | 35 |
| 3.3.1 Técnicas e ferramentas utilizadas..... | 35 |
| 3.3.1.1 Android | 36 |
| 3.3.1.2 jCOLIBRI 2 | 37 |
| 3.3.1.3 MySQL | 38 |
| 3.3.1.4 kSOAP2 e JSON..... | 38 |
| 3.3.2 Operacionalidade da implementação | 39 |

| | |
|--|-----------|
| 3.4 RESULTADOS E DISCUSSÃO | 52 |
| 4 CONCLUSÕES..... | 55 |
| 4.1 EXTENSÕES | 56 |
| REFERÊNCIAS | 57 |
| APÊNDICE A – Descrição dos Casos de Uso | 60 |
| APÊNDICE B – Dicionário de dados..... | 63 |
| APÊNDICE C – <i>Stop words</i> | 66 |

1 INTRODUÇÃO

Diariamente surgem novas técnicas de tratamento, formas de diagnóstico, métodos preventivos e vários outros ensinamentos em diversas áreas e subáreas médicas. Segundo Sabbatini e Cardoso (2013), neste âmbito são publicados 6.000 livros novos e mais de 25.000 revistas e outros tipos de periódicos mensalmente.

Os médicos, por sua vez, estão sentindo uma enorme pressão e urgência no sentido de informatizar seus consultórios, clínicas, hospitais e laboratórios. O acesso a um computador já não é mais o problema principal, mas sim a carência de um sistema que atenda suas necessidades (SABBATINI; CARDOSO, 2001).

O grande tempo gasto com as atividades administrativas, documentais e financeiras, e a necessidade de racionalizar e cortar custos de pessoal se somam ao interesse médico de ter os seus prontuários bem organizados, legíveis e fáceis de achar [...]. (SABBATINI; CARDOSO, 2001).

De acordo com Universidade Federal de São Paulo (UNIFESP) (2013) os profissionais mais afetados por esse estigma são os clínicos gerais, que têm contato com a área psicológica, mais precisamente no diagnóstico e tratamentos de transtornos. De acordo com UNIFESP (2013), seis distúrbios neuropsiquiátricos estavam entre as vinte principais causas de problemas na população, atendidas pelos médicos da rede ambulatorial do Sistema Único de Saúde (SUS), e do Programa de Saúde da Família.

Conforme estudo feito pela UNIFESP (2013) foi apontado que clínicos gerais têm dificuldade para diagnosticar distúrbios mentais devido ao curto tempo de atendimento e a falta da atenção necessária para os casos registrados. Em determinada parte do estudo um dos clínicos gerais envolvidos afirma que “é mais fácil prescrever uma medicação para o paciente não falar durante a consulta e assim não demorar muito”.

Coiera (1998) cita que entre todas as pesquisas tecnológicas modernas, a mais ambiciosa seria a busca por sistemas que utilizem a Inteligência Artificial (IA), que Rich e Knight (1991) definem como o "estudo de como fazer os computadores realizarem tarefas as quais, até o momento, os homens fazem melhor", para resolver este tipo de problema da medicina. Neste cenário, tais sistemas ajudariam os profissionais clínicos na realização de diagnósticos, como o de transtornos, utilizando metodologias de IA como Raciocínio Baseado em Casos (RBC), Redes Neurais e Sistemas Especialistas.

Accioly (2013) cita que médicos podem sanar dúvidas com rapidez e segurança, com apenas um toque na tela, utilizando dispositivos móveis, em plataformas como Android e iOS,

que se utilizam das técnicas de IA citadas e resolvem de forma mais cômoda os problemas dos profissionais médicos.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é disponibilizar um sistema capaz de manter informações de pacientes utilizando indicações de transtornos mais prováveis com base nas diretrizes encontradas no Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM) na quarta versão, ou DSM-IV, o mais recente até o início do desenvolvimento deste trabalho.

Os objetivos específicos do sistema são:

- a) permitir indicar possíveis transtornos através de entrevistas em textos feitas pelo profissional juntamente ao paciente;
- b) permitir o acompanhamento psicológico do paciente, através da disponibilização de informações vitais ao profissional;
- c) manter o histórico clínico de casos já solucionados.

1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica com os conceitos de transtornos mentais, raciocínio baseado em casos, plataforma Android e a apresentação de trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento do sistema iniciando-se com o levantamento de informações, tendo na sequência a especificação, a implementação e, por fim, os resultados e discussão.

No quarto capítulo tem-se as conclusões deste trabalho bem como apresentam-se sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os aspectos teóricos relacionados ao trabalho. São destacados nos subitens os aspectos importantes dos conceitos transtornos mentais, raciocínio baseado em casos, plataforma Android e trabalhos correlatos.

2.1 TRANSTORNOS MENTAIS

Transtornos mentais, também conhecidos como transtornos psicológicos, doenças mentais, ou comportamentos anormais, como definido por Holmes (1997), são condições médicas, amplamente estudadas nas áreas de psiquiatria. Tais condições causam anormalidades, sofrimento, mal-estar ou comprometimento em ordem psicológica, mental ou cognitiva em um determinado indivíduo, podendo constituir-se em escalas.

Para um determinado conjunto de comportamentos ou condições anormais serem consideradas como doença mental é preciso encontrar evidências de continuísmo ou persistência temporal.

Um comportamento anormal ou um curto período de anormalidade do estado afetivo não significa, em si, a presença de distúrbio mental ou de comportamento. Para serem categorizadas como transtornos, é preciso que essas anormalidades sejam persistentes ou recorrentes e que resultem em certa deterioração ou perturbação do funcionamento pessoal, em uma ou mais esferas da vida. Os transtornos mentais e comportamentais se caracterizam também por sintomas e sinais específicos e, geralmente, seguem um curso natural mais ou menos previsível, a menos que ocorram intervenções. Nem toda deterioração humana denota distúrbio mental (BALLONE, 2008).

As doenças mentais são definidas pela Classificação Estatística Internacional de Doenças e Problemas Correlatos de Saúde, tendo sua identificação pela Classificação Internacional de Doenças (CID-10), ou seja, obedecem a descrições clínicas e normas de diagnóstico. Essas informações podem ser usadas em critérios diferentes de diagnóstico, como o caso do DSM, publicação da Associação Psiquiátrica dos Estados Unidos (APA).

2.1.1 Causas

De acordo com Ballone (2008, apud PSIQWEB, 2013), os transtornos podem ser causados por uma vasta gama de fatores, cada qual com sua complexidade, podendo em muitos casos ser divididos em fatores biológicos, psicológicos e sociais. Os determinantes e agravantes destes fatores ainda constituem um grande obstáculo ao estudo e compreensão dessa área, desde que estes envolvem tópicos de genética e ambiente, ou na interação da biologia com fatores socioambientais, conforme a seguir:

- a) fatores biológicos: envolvem idade e sexo do paciente, associadas a perturbações da comunicação neural, tendo como exemplo doenças como depressão e esquizofrenia;
- b) fatores psicológicos: envolvem o ambiente natural e social, os eventos que nele ocorreram, e o impacto que causam ao indivíduo, sendo estes desejáveis ou não, tendo como maior exemplo a ansiedade;
- c) fatores sociais: envolvem tópicos como urbanização, pobreza e industrialização que tem como consequência fatores estressores ao indivíduo como poluição, congestionamentos e a dependência da economia, tendo como exemplo o alcoolismo.

2.1.2 Diagnóstico

De acordo com Ballone (2008, apud PSIQWEB, 2013), os transtornos são identificados e diagnosticados através de métodos clínicos semelhantes, porém ao mesmo tempo distintos aos utilizados no diagnóstico de transtornos físicos. São três os métodos mais utilizados:

- a) entrevista: também conhecida como anamnese, é colhida junto ao paciente e com outras pessoas próximas;
- b) exame clínico: muitas vezes realizado sistematicamente, é usado para verificar o estado mental e condições orgânicas do paciente;

- c) testes especializados: são usados com enfoque em uma ou mais características do possível transtorno.

2.1.3 Tratamento

Dependendo do fator causador da doença mental podem ser usadas duas categorias de tratamentos: somática e psicoterapêutica. A categoria somática envolve terapias medicamentosas e eletroconvulsivantes, enquanto a segunda categoria envolve psicoterapia com técnicas de terapia comportamental e hipnoterapia.

Conforme Manual Merck (2013), cada distúrbio depende de uma combinação ou do uso isolado desses tratamentos, sendo os tratamentos mais eficazes as combinações terapêuticas com medicamentos e psicoterapias. Segue abaixo uma lista detalhando cada tipo de terapia:

- a) terapia medicamentosa: utiliza fármacos altamente eficazes e classificados de acordo com a perturbação principal. Podem ser antidepressivos, antipsicóticos, ansiolíticos e estabilizantes de humor;
- b) terapia eletroconvulsionante: utiliza elétrodos na cabeça com o fim de provocar uma série de descargas elétricas para induzir convulsões. É usado no tratamento de depressão grave;
- c) psicoterapia: tratamento onde um terapeuta (entenda-se como psiquiatra, psicólogo clínico, trabalhador social ou enfermeiro) aplica técnicas psicológicas e faz uso sistemático da relação paciente/terapeuta. Auxilia o paciente a compreender as suas estruturas e conflitos internos, amenizando o efeito da doença;
- d) hipnose: promovem técnicas de relaxamento, tratando dor e perturbações físicas causadas por algum transtorno. Pode ajudar pacientes com cancro, por exemplo.

2.1.4 Manual Diagnóstico e Estatístico de Transtornos Mentais

O DSM é um manual para profissionais da área da saúde mental que lista diferentes categorias de transtornos mentais e critérios para diagnosticá-los. É usado ao redor do mundo por clínicos e pesquisadores bem como por companhias de seguro, indústria farmacêutica e parlamentos políticos (AMERICAN PSYCHIATRIC ASSOCIATION, 2012).

De acordo com American Psychiatric Association (2002) o DSM teve sua primeira publicação em 1952 e até o presente momento, de acordo com American Psychiatric Association (2013), está na sua quinta revisão, o DSM-V. O manual começa com instruções a respeito do uso do manual, seguidas pela Classificação do DSM que fornece uma listagem sistemática dos códigos e categorias oficiais. Em seguida vem a descrição do sistema multiaxial do DSM para diagnóstico. Segue-se então os critérios de diagnóstico para cada perturbação do DSM acompanhada pelo texto descritivo. Finalmente, o DSM apresenta seus apêndices.

Conforme American Psychiatric Association (2002), muitas perturbações do DSM têm um código de diagnóstico que aparece várias vezes:

- a) precedendo o número da perturbação na sua devida classificação;
- b) no início da seção de texto para cada perturbação;
- c) acompanhando o conjunto de critérios para cada perturbação.

A utilização de códigos de diagnóstico é fundamental para o recolhimento de informação médica. A codificação dos diagnósticos facilita o recolhimento de dados e a recuperação e recompilação de informação estatística. Com frequência, também se requer dos códigos que facilitem a comunicação de dados de diagnóstico a terceiras partes interessadas, incluindo instituições governamentais, seguradoras privadas e a Organização Mundial de Saúde (AMERICAN PSYCHIATRIC ASSOCIATION, 2002).

Conforme American Psychiatric Association (2002), habitualmente, o diagnóstico do DSM aplica-se a manifestações atuais do sujeito e não costuma utilizar-se para denotar diagnósticos anteriores, dos quais o indivíduo já esteja recuperado. Uma vez estabelecido o diagnóstico, podem aplicar-se as seguintes especificações indicadoras da gravidade e curso evolutivo: Ligeiro, Moderado e Grave, Em Remissão Parcial, Em Remissão Total e História Anterior.

O texto do DSM-IV descreve cada perturbação sob as seguintes epígrafes, ou abordagens, de acordo com American Psychiatric Association (2002):

- a) características de diagnóstico: esta seção clarifica os critérios diagnósticos e podem ser dados exemplos ilustrativos;
- b) subtipos e ou especificações: esta seção proporciona definições e análises breves referentes aos subtipos e ou especificações aplicáveis;
- c) procedimentos de registro e tipificação: esta seção traz diretrizes para registrar o número da perturbação e selecionar e registrar os códigos diagnósticos ICD-9-MC adequados. Inclui também instruções para aplicar quaisquer subtipos e ou especificações adequadas;
- d) sintomas e perturbações associados: esta seção inclui certas características clínicas que se associam frequentemente à perturbação, mas não são consideradas essenciais para formular o diagnóstico. Esta seção também pode incluir informações de dados laboratoriais e dados da exploração física e estados físicos associados;
- e) sintomas dependentes da cultura, idade e gênero: esta seção fornece ao clínico diretrizes sobre variantes distintas da apresentação da perturbação que podem atribuir-se à situação cultural e ao estágio do desenvolvimento (por exemplo, primeira e segunda infâncias, adolescência, maturidade, velhice) ou o sexo do sujeito.
- f) prevalência: esta seção proporciona os dados disponíveis sobre a prevalência atual e ao longo da vida, incidência e risco ao longo da vida. Estes dados são fornecidos em relação com diferentes situações (por exemplo, comunidade, cuidados primários, consultas externas de clínicas de saúde mental e estabelecimentos de hospitalização psiquiátrica) quando se dispõe de tal informação;
- g) evolução: esta seção descreve os padrões típicos de apresentação e evolução da perturbação ao longo do tempo. Contém informação sobre a idade de início e o modo de início (por exemplo, súbito ou insidioso) típicos da perturbação; evolução episódica versus contínua;
- h) padrão familiar: esta seção descreve dados acerca da frequência da perturbação entre parentes biológicos de primeiro grau comparando-os com a frequência na população em geral;
- i) diagnóstico diferencial: esta seção discute como diferenciar esta perturbação de outras perturbações que têm características de apresentação semelhantes.

2.2 RACICÍONIO BASEADO EM CASOS

O RBC é uma técnica de IA que envolve o processo de resolver novos problemas baseado em soluções de outros problemas recentes, conforme Riesbeck e Schank (1989). Esta visão de como resolver problemas está relacionada com a teoria da prototipagem, uma área da ciência cognitiva. Pode ser usado como modelo cognitivo para se entender alguns aspectos do pensamento e comportamento humanos.

Conforme Delpizzo (1997), o RBC começou com o trabalho de Roger Schank, teórico da inteligência artificial e psicólogo cognitivo norte-americano, e seus estudantes na universidade de Yale no meados de 1980, quando desenvolveram o modelo Schank de memória dinâmica. Esse modelo foi utilizado por Janet Kolodner e Michael Lebowitz que o utilizaram nos primeiros sistemas de RBC: o Cyrus e IPP.

De acordo com Bartsch (1997, apud THÉ, 2001), com estes dois sistemas em uso, várias universidades próximas começaram a combinar o RBC com outros métodos de raciocínio. Em meados de 1990 com o aumento da popularidade dos sistemas RBC, seu interesse acabou se tornando internacional sendo utilizado em várias empresas do mundo.

Para RBC, um caso é a representação de um conhecimento no computador. Os casos mantêm as informações necessárias pelos usuários. Kolodner (1993) descreve as características de um caso em 3 partes:

- a) a descrição do caso, a qual permite sua identificação e armazenamento;
- b) o caso em si, contendo as informações relevantes para o domínio de sua aplicação;
- c) o estado posterior do domínio quando a solução é aplicada.

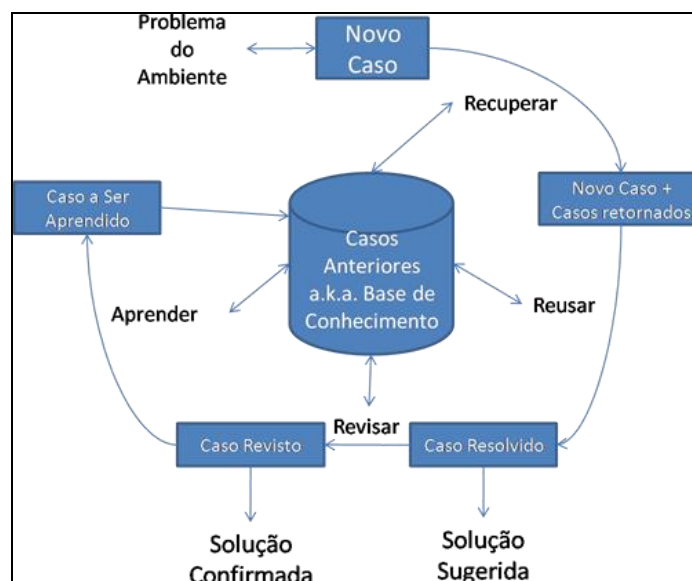
A experiência abstraída no caso deve estar descrita em termos de seu conteúdo e contexto, conforme Kolodner (1993). Kolodner e Leake (1996) citam que um caso pode ter diferentes formas e tamanhos, associando soluções com problemas, resultados com situações ou vice-versa. Ainda citando Kolodner e Leake (1996) “se o que for diferente numa nova situação ensinar algo que não possa ser facilmente inferido do novo caso já gravado, então é útil gravá-lo como um novo caso”.

A função da indexação é orientar a avaliação da similaridade dos casos da base, através de um conjunto de características que representam um caso. Watson (1997) indica que os índices devem ter as seguintes características: ser prognósticos, endereçar os propósitos dos casos que serão usados, ser abstrato o bastante para permitir a ampliação dos casos da base, ser concreto o bastante para serem reconhecidos no futuro e variar com o domínio e com a tarefa do sistema. Kolodner (1996) indica que dois aspectos devem ser enfocados ao tratar os índices: o primeiro é definir o vocabulário e o segundo é como estes índices vão ser valorados.

Segundo Aamodt e Plaza (1994), o RBC foi formalizado em quatro passos, conforme Figura 1:

- a) recuperação: recuperar os dados e informações necessárias através da identificação das características e usando indexação;
- b) reuso: reutilizar a solução de um caso em outros;
- c) revisão: diferenciar as soluções com seus problemas, as soluções nem sempre são idênticas e precisam ser revisadas;
- d) aprender: é o fato de armazenar um caso com sua possível solução para uma futura recuperação.

Figura 1 - Processo do RBC



Fonte: Larbc (2010).

2.2.1 Vantagens do RBC

A seguir listam-se algumas vantagens de desenvolver-se um sistema RBC:

- a) extração de conhecimento: é necessário apenas um banco de dados;
- b) aprendizagem: a atualização de conhecimento é feita de forma automática, na medida em que as experiências são utilizadas. Kolodner (1993) cita que na medida em que o sistema cresce, conseqüentemente, aumenta sua robustez e eficácia;
- c) justificativas: Weber (1996, apud DELPIZZO, 1997) cita que as justificativas são sempre consistentes com as soluções por serem sempre as próprias experiências, representando mais um aspecto de proximidade ao comportamento humano;
- d) fácil acesso às soluções: a recuperação de uma solução é feita de forma rápida, diferente de outros tipos de sistema. Outra vantagem que parte do mesmo princípio é que o sistema não entenda perfeitamente as condições e circunstâncias do problema para propor uma solução;
- e) raciocínio implícito: Riesbeck e Schank (1989) citam que “A incerteza implícita das informações contidas nos casos é utilizada sem a necessidade de um tratamento específico”.

2.2.2 Framework

Conforme definiu Anacleto (2011), *framework*, ou *shell*, é um conjunto de ferramentas utilizadas para aproveitamento de linha código e otimização de tempo no desenvolvimento de aplicação para as mais variadas situações.

Segundo Anacleto (2011) os *frameworks* mais conhecidos que utilizam RBC como método de aprendizado são:

- a) CBRWorks: desenvolvido pela empresa TecInno/empolis, essa ferramenta é um pacote de desenvolvimento de software adequado para a criação de soluções inteligentes em uma grande variedade de domínios de aplicação e ambientes operacionais;
- b) myCBR: é um *framework* de código aberto desenvolvido pelo grupo DFKI. Suas principais características são a facilidade de usar, permitir a prototipagem rápida,

ser extensível e adaptável e é desenvolvido sobre a ferramenta de ontologia Protégé;

- c) jCOLIBRI 2: é um *framework* em Java desenvolvido pelo grupo GAIA. Ele inclui mecanismos para recuperar, reusar, revisar e reter casos e foi desenvolvido para ser facilmente implementado com novos componentes.

2.2.3 Textual Case-based Reasoning

Segundo Weber, Ashley e Brüninghaus (2005), *Textual Case-Based Reasoning* (TCBR) é uma subárea do RBC focada na pesquisa e implementação de sistemas RBC onde alguma ou toda fonte de conhecimento estão disponíveis em forma de texto e os utiliza de forma automática ou semiautomática para resolver problemas através da comparação de casos.

Weber, Ashley e Brüninghaus (2005) citam que os principais objetivos dos sistemas TCBR são: recuperar os casos textuais relevantes para resolver o problema textualmente descrito, extrair ou destacar as passagens nos textos dos casos encontrados, extrair e assimilar os novos índices em casos para serem recuperados no futuro, ou usar os casos textuais para o sistema interpretar o problema.

De acordo com Recio-García, Díaz-Agudo e González-Calero (2007), não parece haver uma padronização estrutural em um sistema TCBR por causa dos diferentes domínios de conhecimentos que esse método pode se basear.

Recio-García, Díaz-Agudo e González-Calero (2007) descrevem os dois principais grupos de algoritmos utilizados pelo TCBR, tais como:

- a) extração da informação (IE): método que captura as características do texto e utiliza o algoritmo de similaridade *Nearest Neighbour* para definir quais os atributos mais similares. A Recuperação Semântica de casos é baseada no modelo de camadas de Lenz e utiliza algoritmos de Entropia Máxima que foram desenvolvidos e aplicados em Linguagem Natural no pacote OpenNLP1. O OpenNLP provê uma representação estruturada dos casos para que possam ser aplicados os algoritmos de similaridade do RBC. Porém, a principal desvantagem do OpenNLP é que estes métodos de similaridade somente podem ser utilizados onde os textos são mapeados em casos com uma estrutura fixa. Isso se deve ao fato de todos os casos possuírem os mesmos atributos (ANACLETO, 2011);

- b) recuperação da informação (IR): utilizado em motores de busca e baseados no Modelo Espacial Vetorial (VSM). Barros (2011) descreve que este modelo associa pesos positivos aos termos encontrados e permite o casamento parcial entre consulta realizada e caso na base de casos. Está disponível no *framework* jCOLIBRI 2 e é baseada no motor de busca Apache Lucene. O Lucene utiliza uma combinação do Modelo de Espaço Vetorial (VSM) da recuperação de Informação (IR) e o modelo Booleano para determinar quão relevante um dado documento é para uma busca feita pelo usuário. As principais vantagens deste método de busca são os resultados relevantes e sua utilização em textos não estruturados. A grande desvantagem é a falta de conhecimento semântico sobre os textos (ANACLETO, 2011).

2.2.4 Modelo Espacial Vetorial

Anacleto (2011) descreve o Modelo Espacial Vetorial como um modelo algébrico para representação de documentos e consultas como vetores.

O Modelo Espacial Vetorial pode ser descrito por dois principais métodos, conforme Anacleto (2011):

- a) *Term Frequency* (TF): quanto maior o número de ocorrências de um termo em um caso, mais relevante é o termo para descrever o documento;
- b) *Inverse Document Frequency* (IDF): termo que aparece em muitos casos não é útil para distinguir relevância.

Barros (2011) descreve que esses métodos podem ser expressos matematicamente, como pode ser visto no Quadro 1 e Quadro 2.

Quadro 1 - Fórmula de *Term Frequency*

$$tf_{i,j} = \frac{freq_{t,d}}{\max_i freq_{i,j}}$$

Quadro 2 - Fórmula de *Inverse Document Frequency*

$$idf_i = \log\left(\frac{N}{n_i}\right)$$

Onde:

- $freq_{t,d}$: frequência do termo k_i no documento d_j ;
- n_j : número de documentos que contém o termo k_i ;
- N : número total de documentos da base;
- $max_i freq_{i,j}$: A frequência do termo mais frequente no documento.

Barros (2011) sumariza a união dos dois cálculos, conhecido como pontuação TF-IDF, nas fórmulas expressas no Quadro 3.

Quadro 3 - Fórmula de pontuação TF-IDF

$$w_{i,j} = tf_{i,j} * idf_i$$

$$w_{i,j} = \left(\frac{freq_{i,j}}{max_i freq_{i,j}}\right) * \log\left(\frac{N}{n_i}\right)$$

Um exemplo dado por Barros (2011) é transcrito a seguir. Este exemplo considera uma variação da fórmula TF ao substituir a frequência do termo mais frequente pela quantidade de palavras no documento. Considere um documento contendo 100 palavras e que a palavra febre aparece 3 vezes. De acordo com o TF febre então será $(3 / 100) = 0,03$. Agora, assumo que existem 10 milhões de documentos e febre aparece em 1000 destes documentos. Então o IDF calculado será $\log(10000000 / 1000) = 4$. A pontuação TF-IDF é o produto destas quantidades: $0,03 * 4 = 0,12$.

2.3 ANDROID

Conforme definiu Lee (2011), Android é um sistema operacional móvel baseado em uma versão modificada do Linux, criada pela empresa homônima, Android Inc.

Lee (2011) cita que em 2005, como estratégia de entrar no espaço de aparelhos móveis, a Google comprou o Android e assumiu o seu desenvolvimento e o time de desenvolvimento.

Segundo Lee (2011), a Google queria que o Android fosse uma plataforma aberta e franca, resultando na liberação do seu código em licença Apache de *Open Source*, onde cada vez mais vendedores, em sua maioria desenvolvedores de hardware, podem criar sua própria extensão proprietária e customizá-la para diferenciar seus produtos. Este modelo de desenvolvimento faz o Android ser muito atrativo e por isso muitos vendedores estão desenvolvendo nesta plataforma.

De acordo com Lee (2011), a partir de 2001 o Android passou a estar nos seguintes dispositivos: smartphones, tablets, e-readers, netbooks, MP4 e TV's internet.

2.4 TRABALHOS CORRELATOS

Pode-se citar como trabalho correlato a monografia “O diagnóstico de problemas psicológicos mediante a aplicação do raciocínio baseado em casos” (MARTIGNAGO; PIUCO; BRAGA; FIALHO, 2012). O modelo proposto por este trabalho consiste no uso do CBR-Works utilizando sete informações que são armazenadas e posteriormente recuperadas para identificar casos similares no sistema. As sete informações são: estado civil, sexo, idade, se estuda, ocupação, sintomas e técnica aplicada. A Figura 2 mostra a tela da ferramenta correlata, solicitando as sete informações.

Figura 2 - Informações usadas

Fonte: Martignago, Piuco, Braga e Fialho (2012).

Outro trabalho correlato que se pode citar é o trabalho “Sistema Baseado Em Casos Para Diagnóstico Médico Veterinário” (ANACLETO, 2011) cujo objetivo foi desenvolver um sistema para apoio a diagnóstico médico veterinário sanando dúvidas quanto aos diagnósticos prescritos e responder perguntas relacionadas a esses diagnósticos. Para o trabalho de Anacleto (2011) foi utilizado o TCBR, no qual o aplicativo partilha os princípios para o diagnóstico dos transtornos. A Figura 3 mostra uma tela de resultado da consulta efetuada com a ferramenta.

Figura 3 - Tela de resultado da consulta de uma pesquisa com palavras-chave.

The screenshot shows a web application window titled 'SBCDMV'. The interface is divided into several sections:

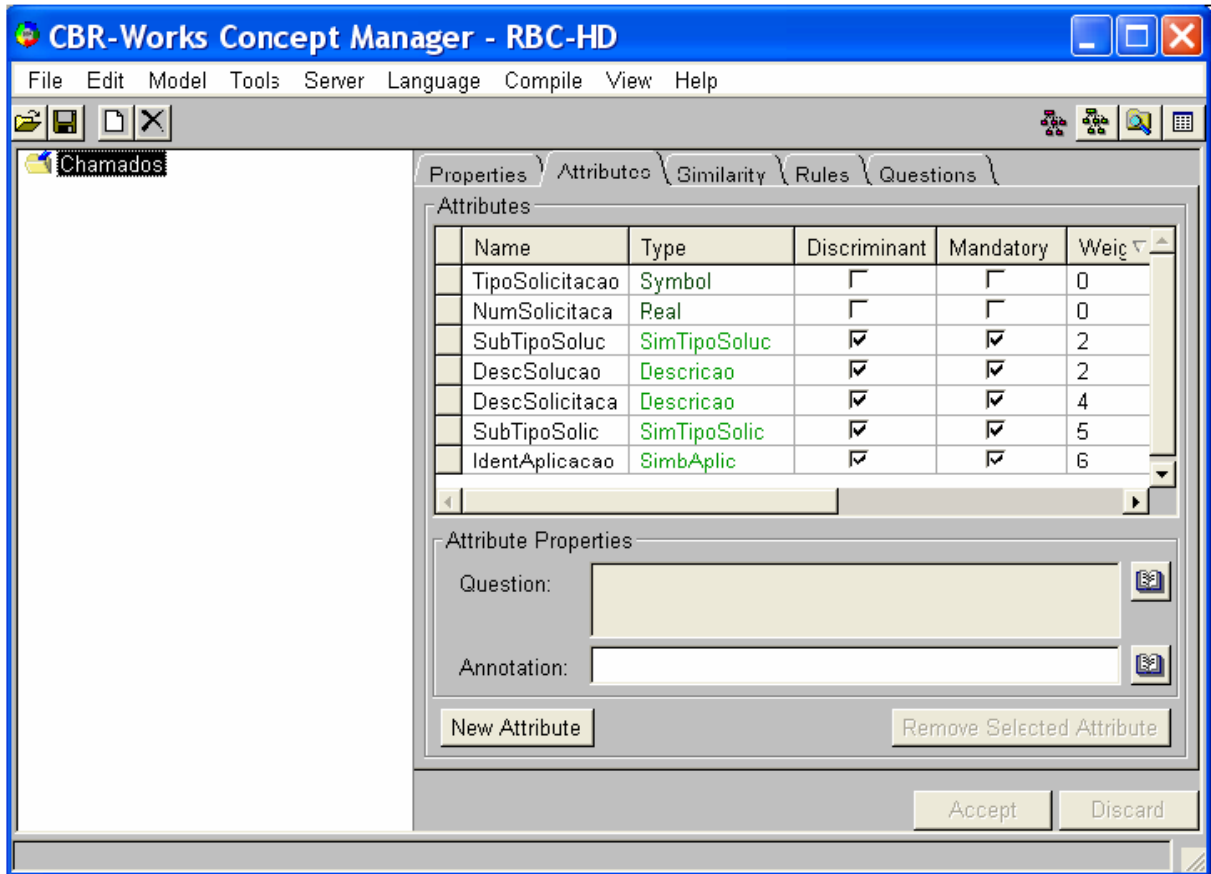
- Arquivo Ajuda**: Located at the top right of the window.
- Casos encontrados:** A list of search results including 'Doença de Lyme a febre maculosa das Montanhas Rochosas', 'Hepatite infecciosa canina', 'Carapatos', 'Leptospirose', and 'Parvovirose'. A search bar contains the text 'Febre maculosa das Montanhas Rochosas'.
- Palavras consultadas:** A text input field containing 'Febre carrapato juntas inchadas'.
- Grau de similaridade:** A slider control set to 1,0.
- Diagnóstico:** A text area containing the text: 'Doença de Lyme a febre maculosa das Montanhas Rochosas.'
- CAUSAS:** A text area containing a detailed paragraph about the disease, its symptoms, and treatment options.
- Descrição:** A text area containing a detailed paragraph about the disease, its symptoms, and treatment options.
- Buttons:** A 'Nova Consulta' button is located at the bottom right of the main content area.

Fonte: Anacleto (2011).

Por fim, o TCC de título “Utilização do raciocínio baseado em casos como apoio a um sistema de *help desk*” (SILVA, 2007), que também utiliza de uma ferramenta de RBC e de

uma forma similar de gerência dos casos do sistema desenvolvido. A Figura 4 mostra uma tela das definições de configurações da ferramenta correlata.

Figura 4 - Tela de definições de casos no CBR-Works 4



Fonte: Silva (2007).

3 DESENVOLVIMENTO

Neste capítulo são detalhados o levantamento de informações, a especificação e a implementação, além dos resultados obtidos e discussão a respeito dos mesmos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

O levantamento de informações foi realizado em conversas com alunos do curso de psicologia, a partir das quais foi possível verificar a necessidade de um sistema que os auxiliasse no momento do diagnóstico e agilizasse os processos gerenciais necessários.

Com esta visão, desenvolveu-se um aplicativo mobile para gerenciar pacientes e indicar possíveis transtornos mentais com base em relatos textuais extraídos pelo médico juntamente ao paciente. No desenvolvimento do aplicativo uma das maiores preocupações foi apresentar um aplicativo que contemple a facilidade em seu uso, bem como indicar ao especialista os possíveis transtornos mentais associados a um grau de similaridade.

O aplicativo utiliza a técnica de TCBR, possibilitando que os médicos cadastrem pacientes, submetam-nos a entrevistas, indicando por qual abordagem gostaria que fosse feita a pesquisa na base de conhecimento para que, então, sejam mostrados os resultados mais similares. Possibilita também salvar a pesquisa atualmente feita, vinculando-a com o resultado (transtorno mental) selecionado para ser usado futuramente em uma nova pesquisa.

Foi criado um controle de acesso para possibilitar que vários médicos utilizem o aplicativo através do cadastro de usuários. Foram disponibilizadas as opções de pesquisa dos pacientes cadastrados no sistema, bem como os transtornos mentais e os casos já resolvidos.

O aplicativo consome um web service Axis2, usa a linguagem de programação Java, framework jCOLIBRI 2 como base para a utilização do TCBR e utiliza o banco de dados MySQL.

3.2 ESPECIFICAÇÃO

Para o desenvolvimento dos casos de uso, diagramas de casos e diagramas de atividade, foi utilizado a ferramenta Enterprise Architect e a notação *Unified Modeling Language* (UML). A representação e a modelagem do banco de dados foi criada utilizando o MySQL Workbench.

3.2.1 Requisitos funcionais do sistema

O Quadro 4 apresenta os requisitos funcionais previstos para o aplicativo e sua rastreabilidade, ou seja, sua vinculação com os casos de uso associados.

Quadro 4 - Requisitos funcionais

| Requisitos Funcionais | Caso de Uso |
|---|--------------------|
| RF01: O sistema deve permitir o usuário efetuar o <i>login</i> no sistema | UC01 |
| RF02: O sistema deve permitir o usuário alterar senha do <i>login</i> . | UC02 |
| RF03: O sistema deve permitir o usuário gerenciar pacientes. | UC03 |
| RF04: O sistema deve permitir o usuário gerenciar transtornos. | UC04 |
| RF05: O sistema deve permitir o usuário gerir consultas. | UC05 |

3.2.2 Requisitos não funcionais

O Quadro 5 lista os requisitos não funcionais previstos para o sistema.

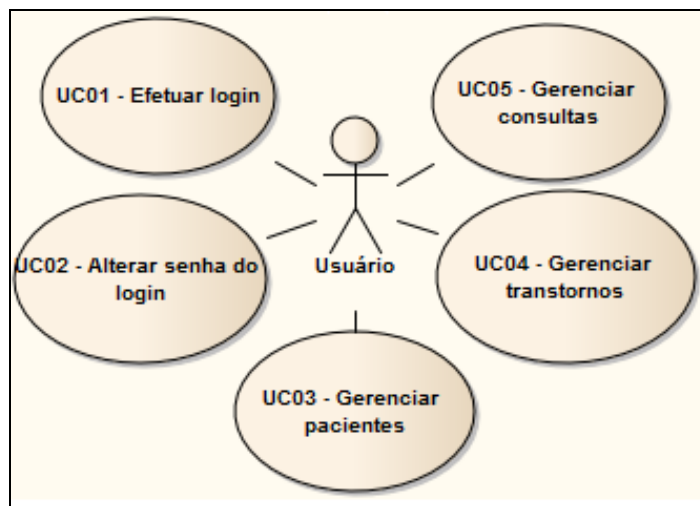
Quadro 5 - Requisitos não funcionais

| Requisitos Não Funcionais |
|---|
| RNF01: O sistema deve utilizar o banco de dados MySQL. |
| RNF02: O sistema deve ser implementado para o sistema operacional Android. |
| RNF03: O sistema deve utilizar o processo de TCBR na gerência de consultas (referente ao RF05). |

3.2.3 Diagramas de casos de uso

Esta subseção apresenta os diagramas de casos de uso conforme Figura 5. A descrição dos principais casos de uso é apresentada no Apêndice A.

Figura 5 - Diagrama de Casos de Uso



As principais funções do usuário do aplicativo (Figura 5) são gerenciar transtornos mentais pré-cadastrados e os cadastrados efetivamente pelas rotinas do TCBR; realizar a gerência dos pacientes, mantendo as informações atualizadas; e realizar a gerência das consultas que vinculam os pacientes aos transtornos selecionados após a entrevista.

Após a função de *login* do usuário do aplicativo é possível verificar através de *grids* os cadastros já realizados por outros usuários, podendo ele pesquisar pelas informações encontradas e alterá-las quando necessário.

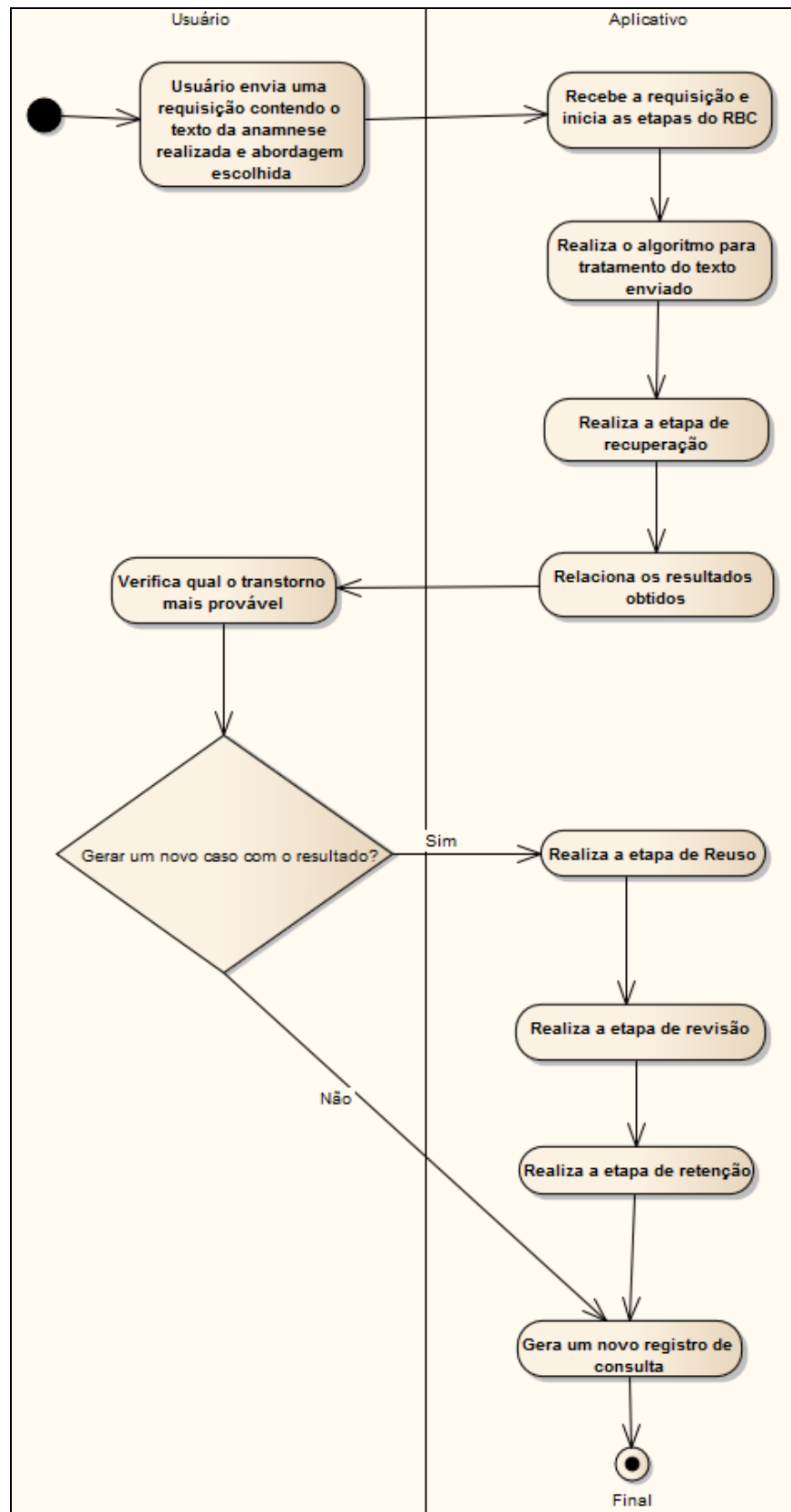
O ator usuário poderá então efetuar o cadastramento dos pacientes com as informações necessárias, as quais irão auxiliá-lo futuramente na escolha do transtorno mais coerente dentre os indicados pelo aplicativo.

Com o paciente escolhido é possível iniciar o cadastramento da consulta, que consiste basicamente em escrever a anamnese, ou entrevista, selecionar a abordagem do DSM e enviar a requisição de pesquisa à base de conhecimento na parte servidora. Com o retorno da requisição será possível selecionar o transtorno mais indicado para o diagnóstico e então gerar um novo registro de consulta contendo o resultado da anamnese, bem como, indicar se o resultado será salvo como um novo caso na base de conhecimento.

3.2.4 Diagrama de atividades

No diagrama de atividade do sistema apresentado na Figura 6, pode ter-se a visão ampla do processamento das etapas do RBC realizadas internamente pela parte servidora do aplicativo e o seu retorno como os transtornos mais prováveis encontrados. Quando uma requisição de pesquisa é feita pelo aplicativo e enviada à parte servidora, o texto contido na mesma passa por uma rotina responsável por tratá-lo em um novo formato para ser então pesquisado efetivamente na base de conhecimento. Com o término dos processos do TCBR, a parte servidora envia ao aplicativo os resultados encontrados para avaliação do usuário.

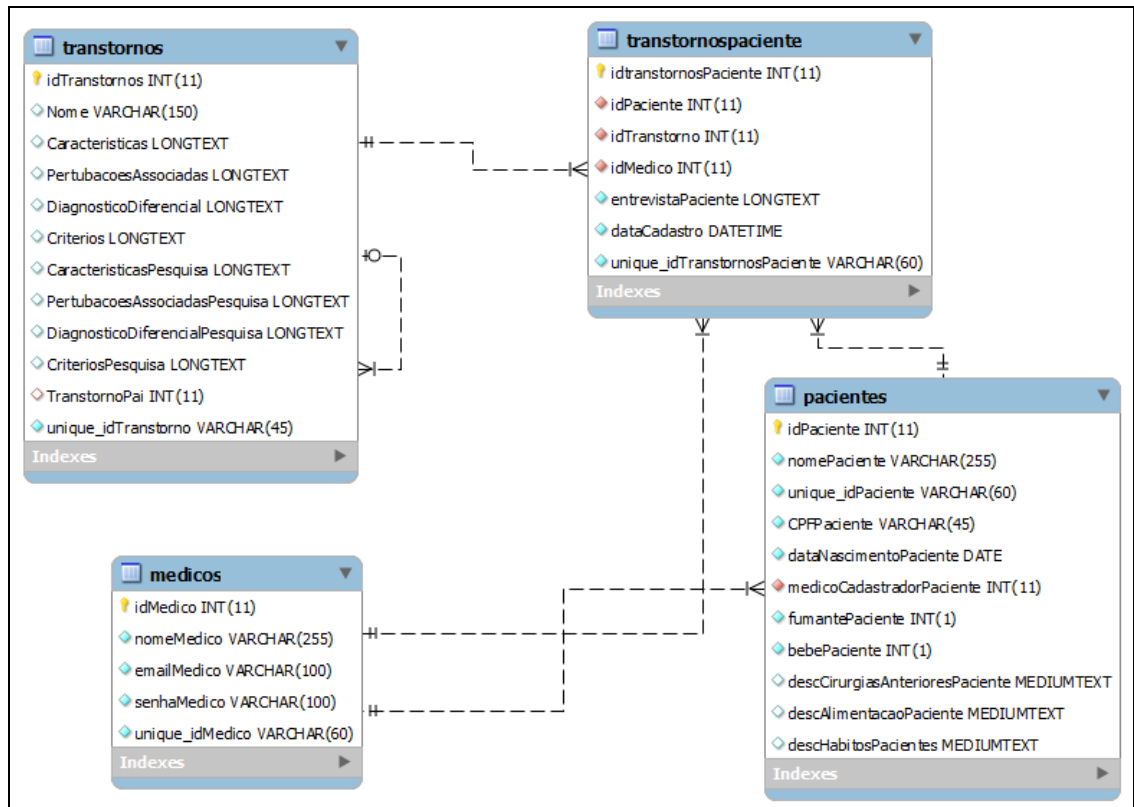
Figura 6 - Diagrama de atividade da solicitação e processamento da requisição



3.2.5 Modelo conceitual da base de dados

A Figura 7 é uma representação gráfica das tabelas utilizadas para armazenar os dados do aplicativo.

Figura 7 - Diagrama de Classes



O aplicativo utiliza o banco de dados MySQL para armazenar todos os dados.

A tabela *Medicos* armazena as propriedades dos usuários utilizados para diferenciar os utilizadores do aplicativo, permitir a entrada e controlar o acesso. Todos os usuários têm permissão para criar novos usuários.

A tabela *Pacientes* armazena as informações mais relevantes dos pacientes cadastrados no aplicativo. As informações variam desde a descrição dos seus hábitos até a sua alimentação, bem como um vínculo com qual usuário o cadastrou.

A tabela *Transtornos* armazena as informações dos transtornos utilizados como base de conhecimento para os processos do TCBR e contém campos com as abordagens no seu formato original e em formato refinado para pesquisa, bem como, se houve o reuso de algum caso, ou seja, qual transtorno originou dado registro (identificado pelo campo *TranstornoPai*). Quando dado registro desta tabela não tiver transtorno antecessor

(`TranstornoPai`) ele é pré-carregado na base de conhecimento, ou seja, é retirado diretamente do DSM. Esta tabela se vincula com a tabela `TranstornosPacientes` para se tornar o transtorno escolhido na resolução da consulta. A tabela também armazena informações como o usuário e paciente envolvidos na consulta.

O dicionário de dados está descrito no Apêndice B.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

O aplicativo foi desenvolvido utilizando a linguagem Java no ambiente de desenvolvimento Eclipse em conjunto com o *Android Developer Tools* (ADT), ou seja, com todas as ferramentas para o desenvolvimento em Android já instaladas. O Eclipse foi utilizado tanto para o desenvolvimento do aplicativo em si, bem como para o ambiente do *framework* Axis2.

Os dados do aplicativo são armazenados em um Sistema de Gerenciador de Banco de dados (SGBD) MySQL. Para auxílio no controle e alterações de tabelas e informações do banco de dados foi utilizado o MySQL Workbench, uma ferramenta visual (ORACLE, 2013). Para as criações de telas do aplicativo foram usadas as ferramentas visuais do ADT.

O servidor de aplicações para comportar o *web service* utilizado é o Apache Tomcat na sua versão 6.0. Tal *web service* foi criado especificamente para manter as bibliotecas e rotinas do jCOLIBRI2 para o processamento das rotinas do TCBR, bem como a comunicação entre o aplicativo e o banco de dados. Foi utilizado como *framework* o Apache Axis2 para criação do *web service*, cujo projeto é *open-source* e utiliza o padrão *Simple Object Access Protocol* (SOAP) (APACHE, 2013).

Para a criação de um meio de comunicação entre o aplicativo e a parte servidora foi utilizado o kSOAP2, uma biblioteca SOAP leve e eficiente (GOOGLE, 2013), e o *JavaScript Object Notation* (JSON), um formato leve para intercâmbio de dados computacionais (JSON,

2013).

3.3.1.1 Android

O Android foi escolhido como plataforma de desenvolvimento, sendo utilizado então para a criação da interface do aplicativo. A versão escolhida para o desenvolvimento foi a *Jelly Bean* ou 4.3. Na Figura 8 pode-se observar um trecho do código da criação da tela de login do aplicativo.

Figura 8 - Código fonte da criação da tela de *login*

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // setting default screen to login.xml
    setContentView(R.layout.Login);

    // Session Manager
    session = new SessionManagement(getApplicationContext());

    if (session.isLoggedIn()) {
        Intent i = new Intent(getApplicationContext(),
            MainPageActivity.class);
        startActivity(i);
    }

    TextView admScreen = (TextView) findViewById(R.id.MudarParaAdm);

    admScreen.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Intent i = new Intent(getApplicationContext(),
                ConfigConectActivity.class);
            startActivityForResult(i, 0);
        }
    });

    TextView registerScreen = (TextView) findViewById(R.id.MudarParaRegistrar);

```

O trecho de código da Figura 8 inicia com a criação da *View*, ou tela, onde o usuário irá manusear as informações. Este procedimento é realizado pelo método `setContentView`. Após a tela ser inicializada, uma variável do tipo `SessionManagement` será criada e fará o controle se o usuário já estava ou não conectado

ao aplicativo através do método `isLoggedIn`. Caso o usuário já tenha se conectado ao aplicativo ele será redirecionado para tela principal, caso contrário, permanecerá na tela de *login*. Se o usuário não foi redirecionado, a rotina irá continuar com os procedimentos de construção de tela instanciando uma variável do tipo `TextView` e no evento chamado `onClick`, ou seja quando o usuário der um *clica-lá*, irá indica-lá que deverá ser feito um redirecionamento para a tela de criação de usuário.

3.3.1.2 jCOLIBRI 2

O jCOLIBRI2 foi o *framework* usado para o desenvolvimento do TCBR, visto que suporta o ciclo de vida completo do RBC. Na Figura 9 pode-se observar um trecho do código do ciclo de vida recuperação.

Figura 9 - Código fonte do ciclo de vida recuperação

```
public List<TranstornoDescricao> cycle2(CBRQuery query,
    String AtributoPesquisa) {
    List<TranstornoDescricao> casosMaisProximo = new ArrayList<TranstornoDescricao>();
    Collection<CBRCCase> cases = _caseBase.getCases();
    NNConfig nnConfig = new NNConfig();

    nnConfig.setDescriptionSimFunction(new Average());

    // We only compare the "description" attribute using Lucene
    Attribute textualAttribute = new Attribute(AtributoPesquisa,
        TranstornoDescricao.class);
    synchronized (lock) {
        nnConfig.addMapping(textualAttribute, new LuceneTextSimilarity(
            luceneIndex, query, textualAttribute, true));
    }
    Collection<RetrievalResult> res = NNScoringMethod.evaluateSimilarity(
        cases, query, nnConfig);
    res = SelectCases.selectTopKRR(res, 5);

    Iterator<RetrievalResult> itr = res.iterator();
    while (itr.hasNext()) {
        Object o = itr.next();
        TranstornoDescricao t = (TranstornoDescricao) ((RetrievalResult) o)
            .get_case().getDescription();
        casosMaisProximo.add(t);
    }
    return casosMaisProximo;
}
```

3.3.1.3 MySQL

O MySQL é um SGBD suportado pela Oracle Corporation. É utilizado para armazenar todos os dados do aplicativo proposto. Na Figura 10 pode-se observar a inserção de um novo transtorno na base de dados.

Figura 10 - Código fonte da inserção de um novo transtorno

```

public long SalvarTranstorno(String Transtorno, String CampoPesquisa)
    throws InstantiationException, IllegalAccessException,
    ClassNotFoundException, SQLException, ExecutionException {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    JSONObject json = new JSONObject(Transtorno);
    String Create = MessageFormat
        .format("insert into transtornos(TranstornoPai, {0}, {1}) values ({2}\", \"{3}\", \"{4}\")",
            CampoPesquisa.substring(0,
                CampoPesquisa.indexOf("Pesquisa")),
            CampoPesquisa, json.get("IDPai"), json.get("Pesquisa"),
            PrepararTexto(json.getString("Pesquisa")));
    ResultSet generatedKeys = null;
    java.sql.Statement statement = SLC.GetConnection().createStatement();
    statement.execute(Create, Statement.RETURN_GENERATED_KEYS);
    generatedKeys = statement.getGeneratedKeys();
    threadPreCycle t = new threadPreCycle();
    t.start();
    if (generatedKeys.next())
        return generatedKeys.getLong(1);
    else
        return -1;
}

```

O trecho de código da Figura 10 inicia com a instância do *Driver* de conexão ao banco de dados. Após isso é criada uma nova variável `JSONObject` que servirá como mensagem de retorno ao aplicativo caso haja algum erro. Após isso é criada uma nova *query* de inserção na tabela `transtornos` com os valores presentes no pacote enviado pelo aplicativo. É criado então um novo `Statement` que irá executar a *query* criada anteriormente e indicará a variável `generatedKeys` qual o ID gerado para o novo registro de transtornos. Ao final é reiniciado o Indexador do Lucene e retornado o ID gerado.

3.3.1.4 kSOAP2 e JSON

O kSOAP2 foi a biblioteca escolhida para ser usada como o canal de comunicação entre o aplicativo e o *web service*. Esta biblioteca trabalha no formato de pacotes e para padronizar o conteúdo dos mesmos foi utilizado o JSON. Na Figura 11 é possível verificar como uma requisição à parte servidora é realizada.

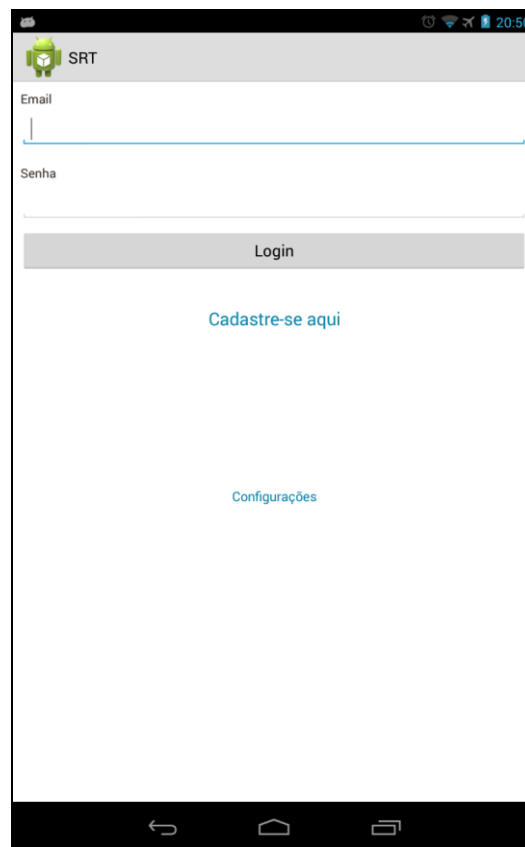
Figura 11 - Código fonte da criação de uma requisição ao *web service*

```
SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(  
    SoapEnvelope.VER11);  
envelope.dotNet = true;  
envelope.setOutputSoapObject(request);  
HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);  
androidHttpTransport.call(SOAP_ACTION, envelope);  
SoapPrimitive resultsString = (SoapPrimitive) envelope  
    .getResponse();  
result = resultsString.toString();
```

O trecho de código da Figura 11 inicia com a criação de um envelope do tipo `SoapSerializationEnvelope` que servirá para gerar a requisição para a parte servidora e fazer retorno com a resposta recebida. A variável do tipo `HttpTransportSE` fará o transporte de pacotes contendo as requisições através do protocolo HTTP, utilizando o endereço (variável `URL`) que aponta para o servidor. Após a requisição ter sido feita, o pacote retorna com a resposta em formato de JSON dada pela parte servidora.

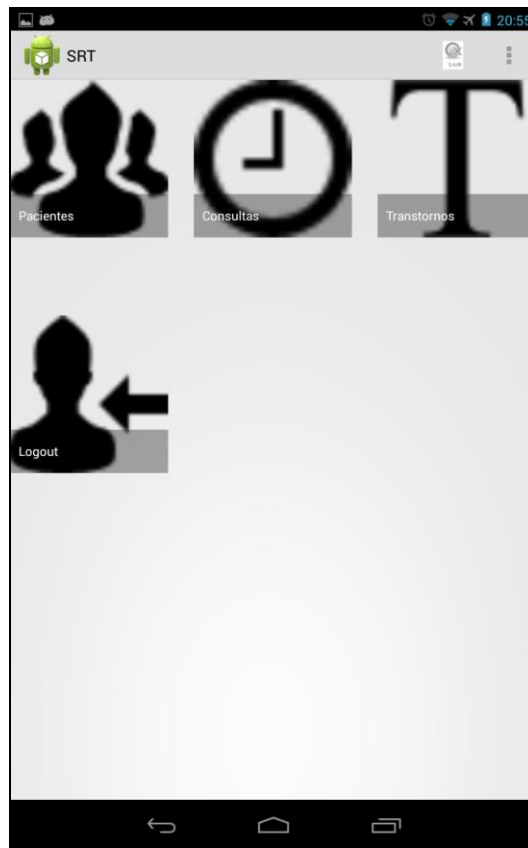
3.3.2 Operacionalidade da implementação

Ao abrir o aplicativo, é apresentada a tela de autenticação do usuário. A Figura 12 apresenta a tela de *login*, que atende ao requisito funcional RF01.

Figura 12 - Tela de *login*

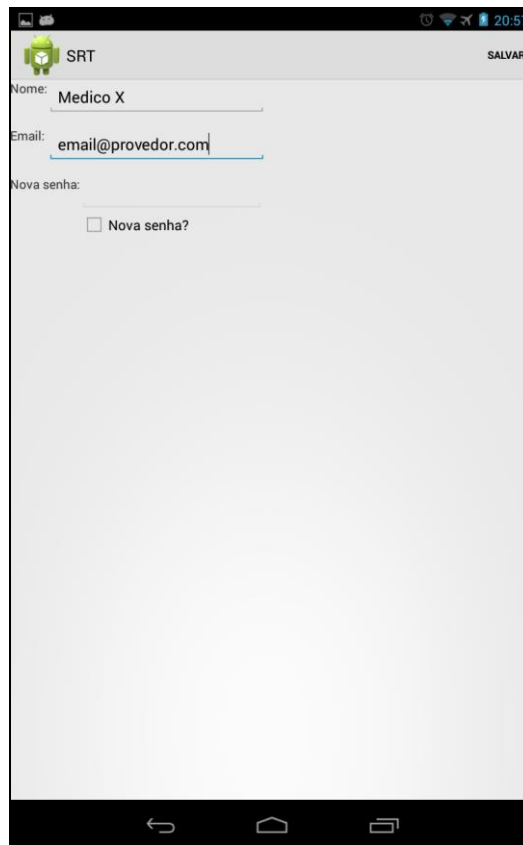
Após a autenticação do usuário, ele é redirecionado para a tela de menu principal onde apresentam-se os menus de consulta de pacientes, transtornos e consultas, bem como as suas configurações de conta e conectividade na bandeja superior ou *Action Bar*, conforme mostra Figura 13.

Figura 13 - Tela principal



Na Figura 14 visualiza-se a configuração de conta do usuário atual, podendo alterar seu *e-mail* e senha. Por padrão, o campo 'Nova senha' é desabilitado para que o usuário possa alterar os outros dados sem precisar modificar a senha. Caso o usuário necessite alterar a senha, é preciso marcar o *checkbox* 'Nova senha' para que o campo seja liberado e alterado. Esta tela atende ao requisito funcional RF02.

Figura 14 - Tela de configuração de conta



The screenshot displays a mobile application interface for account configuration. At the top, the application name 'SRT' is visible on the left, and a 'SALVAR' button is on the right. Below this, there are three input fields: 'Nome' containing 'Medico X', 'Email' containing 'email@provedor.com', and 'Nova senha'. Under the 'Nova senha' field, there is a checkbox labeled 'Nova senha?'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Nas Figuras 15 e 16 visualizam-se as telas de consulta e cadastro de pacientes, respectivamente. A tela de cadastro contém as informações de todos os pacientes cadastrados na base e que poderão ser atualizadas por qualquer usuário. Este cadastro possui todas as funcionalidades de gerência, atendendo então ao requisito funcional RF03.

Figura 15 - Tela de consulta de pacientes



Figura 16 - Tela de cadastro de pacientes



Em seguida, nas Figuras 17 e 18 visualizam-se as telas de consulta e cadastro de transtornos, respectivamente. Este cadastro contém todas as informações dos transtornos pré-cadastrados pelas diretrizes do DSM, bem como os novos transtornos gerados pelas rotinas do TCBR. Apenas os transtornos que não foram pré-carregados poderão ser modificados ou excluídos, para manter fidelidade aos princípios definidos nas especificações do aplicativo. Este cadastro possui todas as funcionalidades de gerência, atendendo então ao requisito funcional RF04.

Figura 17 - Tela de consulta de transtornos

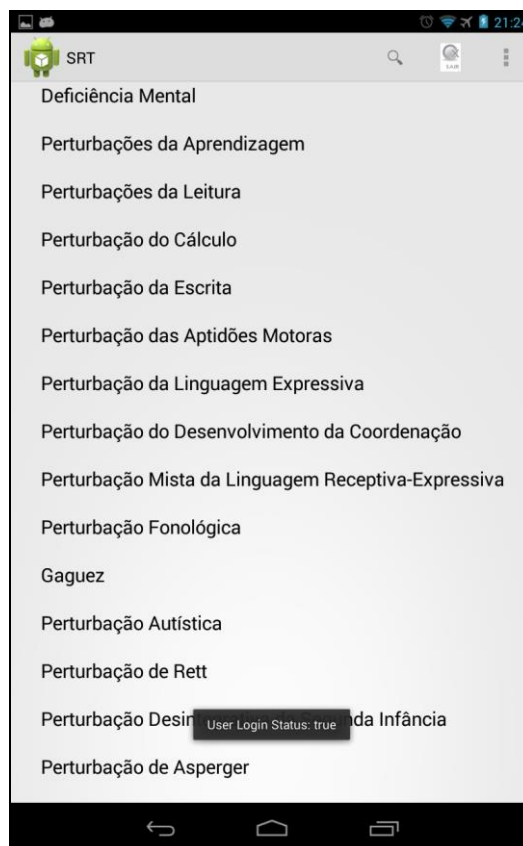
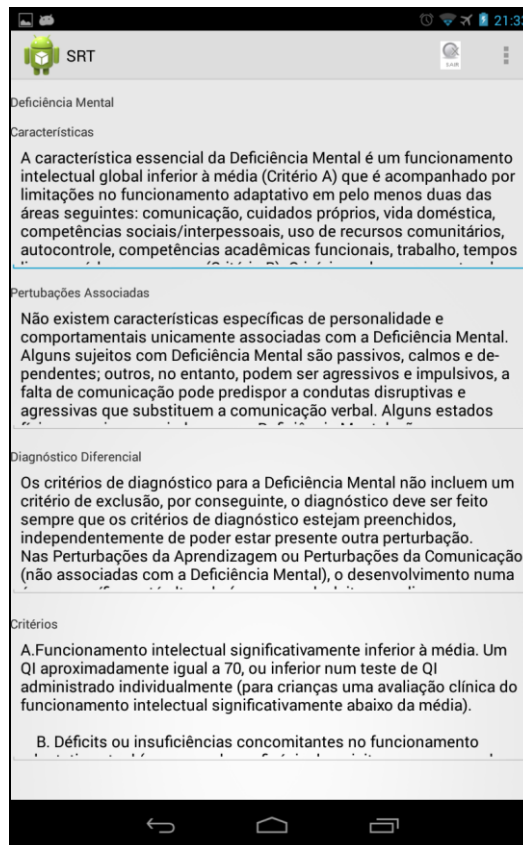


Figura 18 - Tela de cadastro de transtornos



Nas Figuras 19 e 20 visualizam-se as telas de consulta e cadastro de consultas, respectivamente. Este cadastro contém as informações geradas após as rotinas do TCBR. Este cadastro possui todas as funcionalidades de gerência, atendendo então ao requisito funcional RF05.

Figura 19 - Tela de consulta de consultas

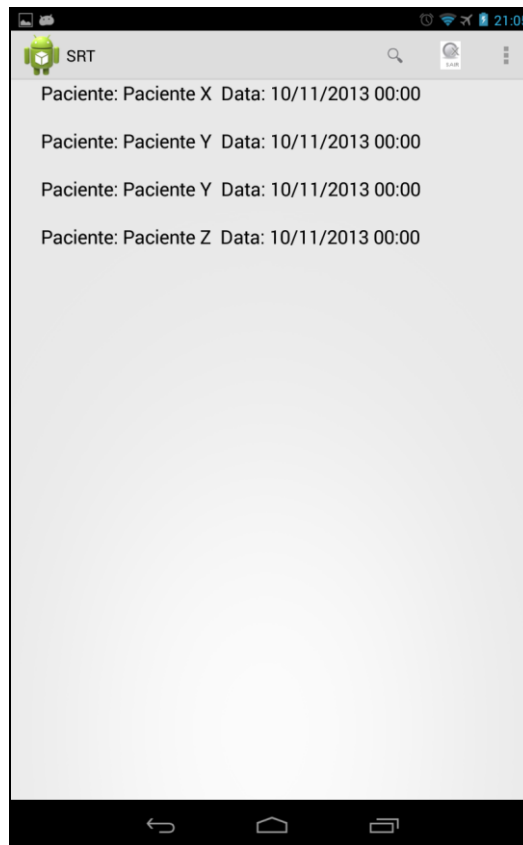
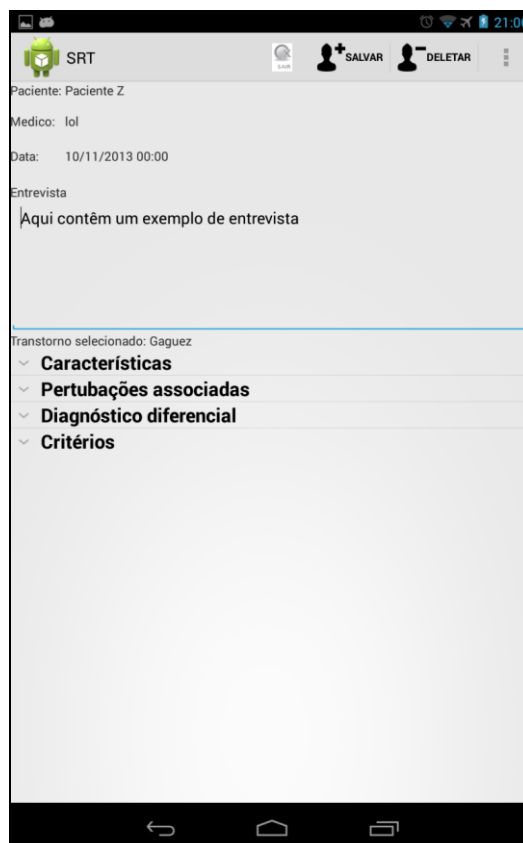


Figura 20 - Tela de cadastro de consultas



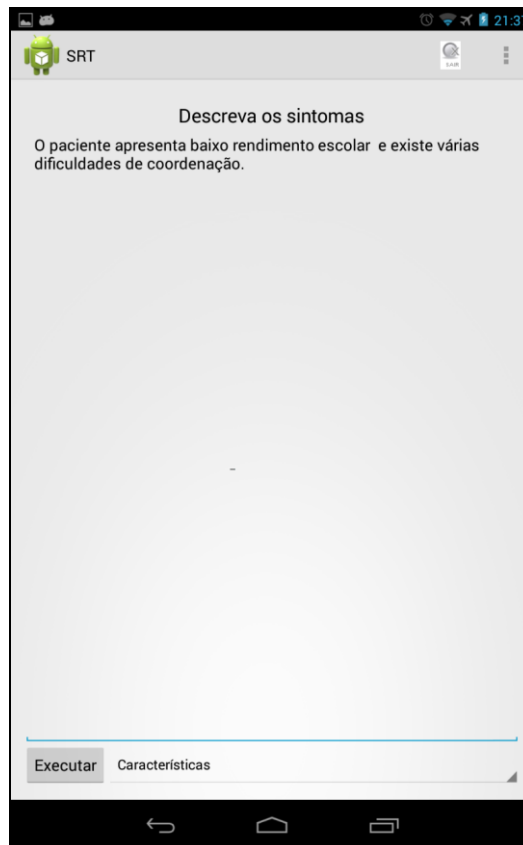
Ao iniciar o *web service* todos os registros da base de conhecimento são carregados em uma lista e indexados pelo Lucene, conforme Figura 21, chamado de Indexador do Lucene. A variável `_caseBase` executa o método `init` indicando o conector do banco de dados que deve ser utilizado. Ao realizar a operação `init`, a parte servidora consegue acesso aos dados da base de conhecimento e através do método `createLuceneIndex` cria o indexador que será usado pelas outras rotinas envolvendo o TCBR. Ao final são retornados os dados encontrados.

Figura 21 - Código de inicialização de casos em memória

```
@Override
public CBRCaseBase preCycle() throws ExecutionException {
    // TODO Auto-generated method stub
    // Here we create the Lucene index
    _caseBase.init(_connector);
    luceneIndex = jcolibri.method.precycle.LuceneIndexCreator
        .createLuceneIndex(_caseBase);
    return _caseBase;
}
```

Após ser selecionado algum registro de paciente, poderá ser iniciada a criação de uma consulta. O usuário então será enviado para a tela de pesquisa (Figura 22) onde o usuário poderá escrever a anamnese em texto livre, podendo ser usada a transcrição de áudio para texto do Android para facilitar o procedimento. Ao final da anamnese será preciso indicar qual a abordagem deverá ser usada para a pesquisa na base de conhecimento. Foram separadas quatro abordagens do DSM para pesquisa, as quais são: características, diagnóstico diferencial, critérios e perturbações associadas. A base de conhecimento já é pré-carregada com as informações de transtornos que não contenham subtipos, pois para estes tipos de transtornos não há abordagens definidas. Ao selecionar as configurações e solicitar a execução, será iniciada a pesquisa na base de conhecimento.

Figura 22 - Tela de envio de pesquisa



Antes do texto ser usado para pesquisa, o mesmo será refinado para buscar resultados mais consistentes. O refinamento consiste em retirar *stop words* e radicalizar o texto, conforme o que segue:

- a) retirar *stop words*: *stop words* são palavras que são filtradas antes, ou depois, do processamento de linguagem natural (texto), podendo ser escolhidas dependendo do propósito. Não há uma lista de *stop words* definitiva, variando entre sistemas (RAJAMARAN;ULLMAN, 2011). As *stop words* escolhidas para este trabalho se encontram no Apêndice C.
- b) radicalizar o texto: mais conhecido como *stemming*, é o método para redução de um termo ao seu radical, removendo as desinências, afixos e vogais temáticas. Com sua utilização, os termos derivados de um mesmo radical serão contabilizados como um único termo (PIRES, 2008). Este procedimento é feito através das rotinas do jCOLIBRI2, conhecido como *SnowBall*, conforme demonstrado na Figura 23.

Figura 23 - Parte do código do *SnowBall*

```

if word in self.stopwords:
    return word

step1_success = False
step2_success = False

word = (word.replace("\xE3", "a~")
        .replace("\xF5", "o~"))

r1, r2 = self._r1r2_standard(word, self.__vowels)
rv = self._rv_standard(word, self.__vowels)

# STEP 1: Standard suffix removal
for suffix in self.__step1_suffixes:
    if word.endswith(suffix):
        if suffix == "amente" and r1.endswith(suffix):
            step1_success = True

            word = word[:-6]
            r2 = r2[:-6]
            rv = rv[:-6]

            if r2.endswith("iv"):
                word = word[:-2]
                r2 = r2[:-2]
                rv = rv[:-2]

            if r2.endswith("at"):
                word = word[:-2]
                rv = rv[:-2]

            elif r2.endswith(("os", "ic", "ad")):
                word = word[:-2]
                rv = rv[:-2]

elif (suffix in ("ira", "iras") and rv.endswith(suffix) and

```

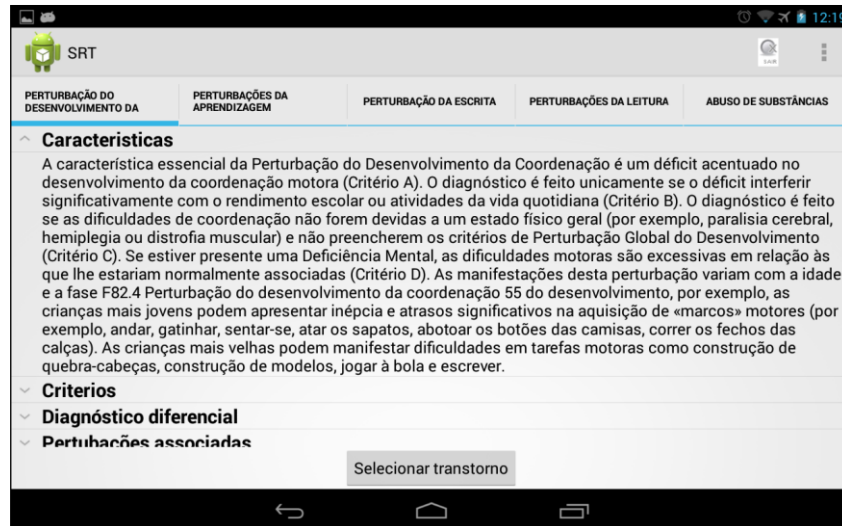
Todos os casos são criados contendo as quatro abordagens em campos não refinados e, em contrapartida, outros quatros campos refinados. O que o usuário vê em tela são os campos não refinados, enquanto nas buscas internas são usados os campos refinados. Os refinamentos já carregados em memória juntamente com o Indexador do Lucene são a chave para uma pesquisa rápida e concisa.

Após o refinamento do texto, a pesquisa será finalmente realizada. Na Figura 9 é possível verificar que estão ocorrendo os seguintes procedimentos:

- criar uma lista que será usada futuramente para retorno ao aplicativo;
- criar uma coleção com todos os casos que estão na base de conhecimento;
- criar um nova variável do tipo `NNConfig` que irá fazer a busca na base de conhecimento;
- é indicado uma variável `Attribute` indicando por qual atributo da classe `TranstornoDescricao` será realizada a pesquisa;
- é usada a variável `nnConfig` para iniciar a pesquisa, indicando o atributo selecionado, o indexador do Lucene (que contém todos os casos em memória) e a *query* (o texto da anamnese).

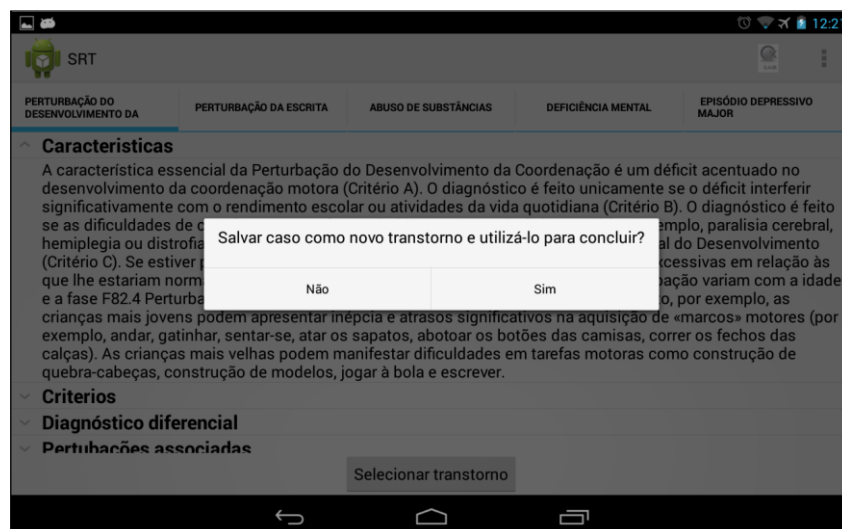
Ao final é mostrada a tela de seleção do transtorno mais indicado, conforme mostra a Figura 24. É possível verificar todas as informações das abordagens, o que facilita a escolha do usuário.

Figura 24 - Tela de seleção de transtorno



Selecionado um transtorno será perguntado ao usuário se ele deseja apenas salvar a consulta, ou se deseja primeiramente salvar o resultado da consulta como um novo registro na base de conhecimento, conforme mostra a Figura 25.

Figura 25 - Tela de seleção de transtorno (Criar novo caso)



Se for escolhida a segunda opção, um novo registro de transtorno será inserido na base, no qual:

- a) serão apenas preenchidos os campos (com e sem refinamento) da abordagem selecionada, ou seja, se foi selecionada a abordagem Características, o texto da anamnese será salvo no campo não refinado de Características, depois ele será refinado e inserido no campo refinado de Características;
- b) será salvo o transtorno que originou este registro, ou seja, um indicador apontando que este registro foi gerado a partir do registro Deficiência Mental, por exemplo.

Quando um registro de transtorno que tenha indicação para outro é visualizado no sistema, o campo de abordagem originador é mantido, enquanto o resto das informações são buscadas do seu registro de transtorno originador.

É possível que um transtorno tenha vários registros “filhos”, da mesma forma que esses “filhos” podem ter seus próprios “filhos”. Quando é visualizado um registro “neto” (por exemplo) será executado um método que irá “escalar” até o último nível da relação, ou seja, a rotina irá verificar se o registro atual contém um indicador, se tiver, vai até o registro indicado e repete a operação anterior, onde só serão buscadas as informações do registro que não contém indicador. A rotina pode ser visualizada na Figura 26.

Figura 26 - Código que procura o caso originador

```
public TranstornoDescricao RetornaTranstornoPai(long ID) {
    Iterator<CBRCCase> itr = _caseBase.getCases().iterator();
    while (itr.hasNext()) {
        Object o = itr.next();
        TranstornoDescricao t = (TranstornoDescricao) ((CBRCCase) o)
            .getDescription();
        if (t.getID() == ID) {
            if (t.getIDPai() <= 0)
                return t;
            else {
                TranstornoDescricao z = RetornaTranstornoPai(t.getIDPai());
                return z;
            }
        }
    }
    return null;
}
```

3.4 RESULTADOS E DISCUSSÃO

O sistema desenvolvido neste trabalho atendeu às expectativas propostas possibilitando ao profissional da saúde obter indicações de possíveis transtornos através de uma pesquisa em texto, bem como a facilidade de uso por ser da plataforma *mobile*, propiciando, assim, uma nova ferramenta para estes profissionais através da técnica de RBC, e claro, da IA como um todo.

A combinação de uso entre o *framework* jCOLIBRI2 e o Modelo Espacial Vetorial foi uma peça fundamental no desenvolvimento deste trabalho. Com o grande potencial de customização das bibliotecas do jCOLIBRI2 foi possível modificá-las a fim de usá-las de forma adequada, para que, então, fosse aproveitada a principal vantagem do Modelo Espacial Vetorial: a ordenação por similaridade.

O Quadro 6 mostra um comparativo entre as principais funções dos sistemas correlatos com o aplicativo desenvolvido.

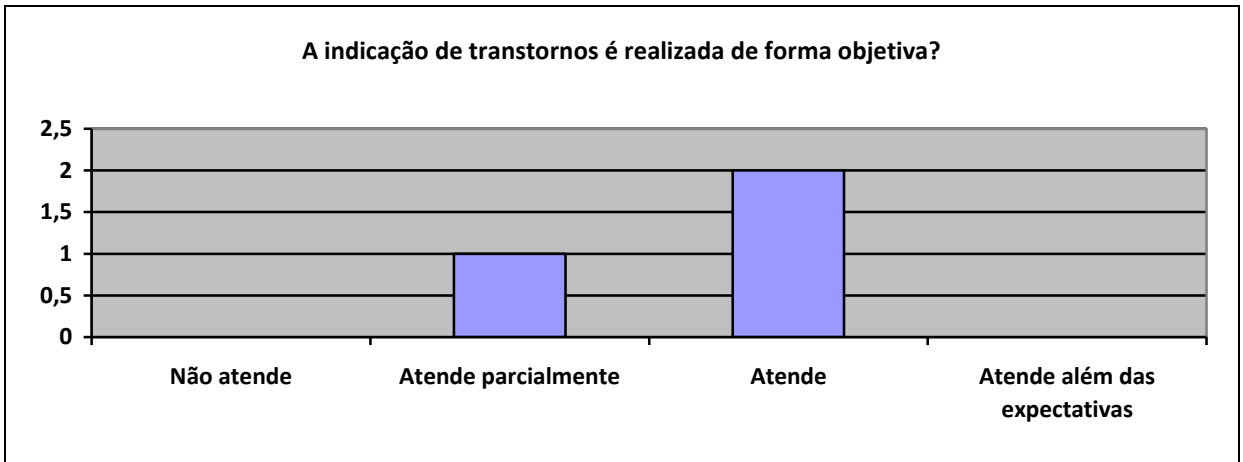
Quadro 6 - Comparativo das principais funções dos trabalhos correlatos

| Funções | Sistema de diagnóstico de problemas psicológicos (MARTIGNAGO; PIUCO; BRAGA;FIALHO, 2012) | Sistema Veterinário (ANACLETO, 2011) | Sistema Help Desk (SILVA, 2007) | Aplicativo desenvolvido no presente trabalho |
|---|--|--|---------------------------------|---|
| Usa <i>Textual Case-based Reasoning</i> | | SIM | | SIM |
| <i>framework</i> RBC utilizado | CBR-Works | jCOLIBRI 2 | CBR-Works | jCOLIBRI 2 |
| Plataforma utilizada | <i>Desktop</i> | <i>desktop</i> | <i>desktop</i> | <i>mobile</i> |
| Capacidade de integração com outros recursos e sistemas | | SIM , em ambientes <i>web</i> através da bibliotecas do jCOLIBRI 2 | | SIM, ao disponibilizar a pesquisa do <i>web service</i> , por exemplo |
| Facilidade de atualização do sistema/aplicativo | | | | SIM, através do próprio gerenciador de atualizações do aparelho móvel |

Ao término do desenvolvimento deste trabalho foi realizado um questionário referente à validação das funções do aplicativo bem como à aceitação da forma de pesquisa e navegação. Foram entrevistados 3 usuários de Blumenau, ligados a esse ramo.

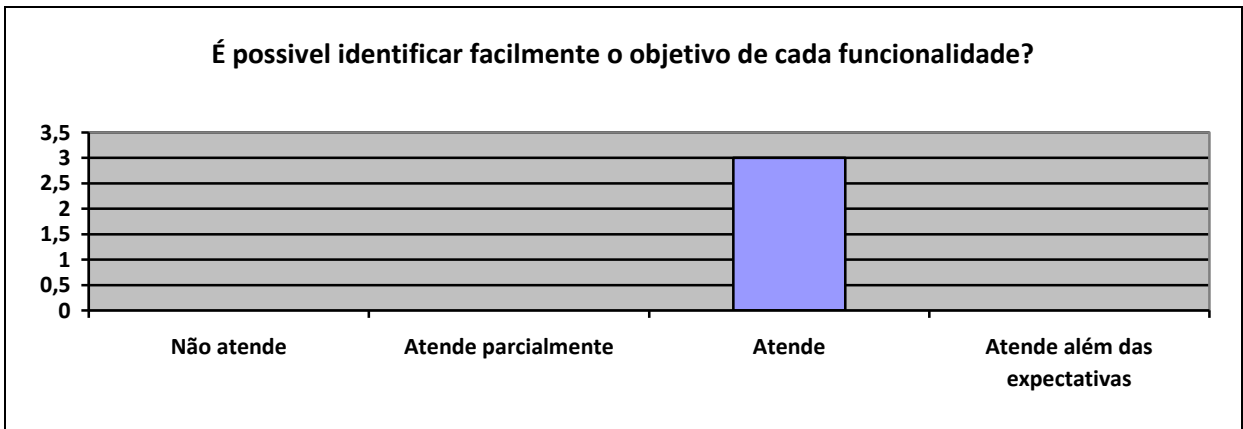
Na Figura 27 tem-se o resultado referente à indicação de transtornos via aplicativo. Nesta questão houve boa aceitação por parte dos entrevistados.

Figura 27 - Respostas referente à indicação de transtornos pelo aplicativo



Na Figura 28 observa-se que o objetivo das funcionalidades está em um grau aceitável para os usuários entrevistados.

Figura 28 - Questionário referente ao grau de aceitação das funcionalidades



4 CONCLUSÕES

A utilização de sistemas de informação, desenvolvidos para atender as necessidades específicas na área da saúde mental, permite dar maior velocidade e concisão no encontro de uma solução aos pacientes atendidos diariamente e assegura uma forma de manter as consultas às informações necessárias. Com este ponto de vista, o presente trabalho propôs-se ao desenvolvimento de um aplicativo com o objetivo de ser capaz de auxiliar o especialista através de indicações de transtornos.

Os objetivos deste trabalho foram alcançados. As rotinas responsáveis pela indicação dos transtornos mais prováveis do aplicativo garantem que as indicações sejam rapidamente obtidas e sejam as mais concisas possíveis. O módulo de gerência de consultas que possibilita manter as informações dos casos já resolvidos, assim como o módulo de gerência de transtornos e módulo de gerência de pacientes facilitam a decisão do profissional. Com o uso dos princípios do TCBR possibilitou-se ter uma nova forma de pesquisa e indicação de transtornos sem restrições por parte de informações ou sintomas e onde a base de conhecimento está sendo sempre alimentada.

A linguagem Java foi utilizada como linguagem de programação para o desenvolvimento do sistema, com utilizações das bibliotecas JSON, kSOAP2, Axis2 e do *framework* jCOLIBRI2 em plataforma Android. Procedeu-se de modo a extrair as qualidades da linguagem e das bibliotecas utilizadas, para assim elaborar um sistema que atingisse os objetivos propostos.

De modo geral, este trabalho proporcionou aprendizado e um novo olhar sobre a área da saúde mental. Isto facilitou o desenvolvimento da solução, permitindo aplicar os conteúdos e conhecimentos adquiridos no decorrer do curso para se criar uma nova ferramenta para estes profissionais.

No decorrer do desenvolvimento foram encontradas e superadas algumas dificuldades. Pode-se citar a falta de conhecimento no desenvolvimento para plataforma Android, das bibliotecas do *framework* escolhido e no controle da comunicação. Com referência à comunicação, houve problemas com os pacotes JSON enviados entre o aplicativo e o *web service*, devido à instabilidade de rede e à capacidade do aplicativo de continuar rodando após um erro.

4.1 EXTENSÕES

Para dar continuidade e aprimorar o aplicativo, sugere-se desenvolver as solicitações recebidas dos entrevistados após a demonstração do *software*. Uma delas é expandir o *software* para outras plataformas, principalmente para a *web*, a mais pedida. Outra sugestão feita foi a possibilidade de se ter modelos de anamnese separados em questionários.

Mais uma melhoria seria desenvolver rotinas para procura e indicações de possíveis soluções, ou tratamentos, para os casos cadastrados pelo aplicativo.

Outra melhoria desejável, que refinaria mais ainda os textos das pesquisas, seria desenvolver uma tabela de sinônimos na qual seriam verificadas as “palavras raízes” e seus sinônimos do texto pesquisado, onde cada sinônimo encontrado no texto seria alterado para a sua devida “palavra raiz”.

Por fim, pequenas melhorias processuais, tais como relatórios e agendamentos de consultas, também foram citadas pelos entrevistados.

REFERÊNCIAS

- AAMODT, A.; PLAZA, E. **Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches**. IOS Press Amsterdam: Artificial Intelligence Communications, 1994.
- ACCIOLY, Renatha. **A inteligência artificial no mundo mobile**. Portland, 2013. Disponível em: <<http://prezi.com/x6hoi8ehks0z/a-inteligencia-artificial-no-mundo-mobile/>>. Acesso em: 22 maio. 2013.
- ANACLETO, Austeciano de Sá. **Sistema baseado em casos para diagnóstico veterinário**. 2011. 133 f. TCC (Bacharel em Ciências da Computação) – Centro Universitário Vila Velha, Espírito Santo.
- AMERICAN PSYCHIATRIC ASSOCIATION. **DSM**. Arlington, 2012. Disponível em: <<http://www.psych.org/practice/dsm/>>. Acesso em: 21 mai. 2013.
- AMERICAN PSYCHIATRIC ASSOCIATION. **DSM-IV. Manual Diagnóstico e Estatístico de Transtornos Mentais**. Porto Alegre : ARTMED, 2002, 4a. ed.
- APACHE. **Apache Axis2/Java**. Forest Hill, 2012. Disponível em: <<http://axis.apache.org/axis2/java/core/>>. Acesso em: 7 nov. 2013.
- BALLONE, Geraldo José. **Portal de psiquiatria geral**. Miami, 2013. Disponível em: <<http://www.psiqweb.med.br/site/>>. Acesso em: 1 abr. 2013.
- BARROS, Flávia. **Modelo de Recuperação de Documentos**. Recife, 2011. Disponível em: <www.cin.ufpe.br/~if796/aulas/cap2-1.ppt>. Acesso em: 5 nov. 2013.
- COIERA, Enrico W. **Inteligência artificial na Medicina**. Campinas, 1998. Disponível em: <<http://www.informaticamedica.org.br/informaticamedica/n0104/coiera.htm>>. Acesso em: 22 maio. 2013.
- DELPIZZO, Vanessa Lins Francalacci. **Prescrição de atividades físicas através do uso da inteligência artificial**. 1997. 81 f. Dissertação (Pós-Graduação em Engenharia de Produção) – Universidade Federal de Santa Catarina, Santa Catarina.
- GOOGLE. **A lightweight and eficiente SOAP library for Android platform**. Mountain View, 2013. Disponível em: <<https://code.google.com/p/ksoap2-android/>>. Acesso em: 7 nov. 2013.
- HOLMES, D.S. **Psicologia dos transtornos mentais**. Porto Alegre: Artes Médicas, 1997.
- JSON. **JSON**. Chesterbrook, 2013. Disponível em: <<http://json.org/json-pt.html>>. Acesso em: 7 nov. 2013.
- KOLODNER, Janet. **Case-Based Reasoning**. San Mateo, CA: Morgan Kaufmann, 1993. 668 p.

KOLODNER, J; LEAKE, D. **A tutorial introduction to CBR. Case-Based Reasoning: Experiences, Lessons, and Future Directions**. Menlo Park: AAAI Press/The MIT Press, 1996.

LARBC. **Fundamentação**. Mountain View, 2010. Disponível em: <<http://code.google.com/p/larbc/wiki/Fundamentacao/>>. Acesso em: 20 mai. 2013.

LEE, Wei-Meng. **Beginning Android Application Development**. 1.ed. Estados Unidos: Wrox, 2011. 448 p.

Manual_Merck. **Tratamento das doenças mentais**. Whitehouse Station, 2013. Disponível em: < <http://www.manualmerck.net/?id=106&cn=947>>. Acesso em: 4 abr. 2013.

MARTIGNAGO, E. V.; PIUCO, J. Z.; BRAGA, M. M; FIALHO, F. A. P. **Aplicação de raciocínio baseado em casos em psicologia**. 2012. 79 f. Pesquisa (Diversas áreas) – Universidad de Jaén, Andalucía, Espanha.

ORACLE. **MySQL Workbench**. Redwood Shores, 2013. Disponível em: <<http://www.mysql.com/products/workbench/>>. Acesso em: 8 nov. 2013.

PIRES, Marina Melo. **Agrupamento incremental e hierárquico de documentos**. 2008. 80 f. Tese (Pós-Graduação em Engenharia Civil) – Universidade Federal do Rio de Janeiro, Rio de Janeiro.

PSIQWEB. **O que são Transtornos Mentais**. Miami, 2013. Disponível em: <<http://www.psiqweb.med.br/site/?area=NO/LerNoticia&idNoticia=230>>. Acesso em: 4 abr. 2013.

RAJAMARAN, A.;ULLMAN, J. D. **Mining of Massive Datasets**. Cambridge University Press, 2011.

RECIO-GARCÍA, J. A.; DÍAZ-AGUDO, B.; GONZÁLEZ-CALERO, P. **Textual CBR in jCOLIBRI: From retrieval to reuse**. Madri, 2007. Disponível em: < gaia.fdi.ucm.es/aigaion2/index.php/attachments/single/4>. Acesso em: 4 nov. 2013.

RICH, E.; KNIGHT, K. **Artificial Intelligence**. 2.ed. New York: McGraw-Hill, 1991.

RIESBECK, C. K.;SCHANK, R. C. **Inside Case-Based Reasoning**. New Jersey: Lawrence Erlbaum Associates, 1989.

SABBATINI, RM.E; CARDOSO, S.H. **Informática e Internet em medicina**. Sunnysvale, 2013. Disponível em: <<http://www.edumed.net/Paginas/pfizer/livro2.html>>. Acesso em: 5 abr. 2013.

SILVA, Jorge Moacir Farias da. **Utilização do raciocínio baseado em casos como apoio a um sistema de help desk**. 2007. 80 f. TCC (Bacharel em Sistemas de Informação) – Centro Universitário Feevale, Novo Hamburgo.

THÉ, Maria Alice Lagos. **Raciocínio baseado em casos: Uma abordagem fuzzy para diagnóstico nutricional**. 2001.170 f. Tese (Pós-Graduação em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis.

UNIVERSIDADE FEDERAL DE SÃO PAULO. **Departamento de Comunicação e Marketing Institucional**. São Paulo, 2013. Disponível em:
<<http://dgi.unifesp.br/sites/comunicacao/index.php?c=Noticia&m=ler&cod=4896defe/>>.
Acesso em: 5 abr. 2013.

WATSON, Ian. **Applying Case-Based Reasoning: techniques for enterprise systems**. San Francisco: Morgan Kaufmann, 1997.

WEBER, R. O.; ASHLEY, K.; BRUNINGHAUS, S. **Textual Case-Based Reasoning**. Cambridge University Press: Knowledge Engineering Review, 2005.

APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos principais casos de uso descritos na seção de especificação deste trabalho. No Quadro 7 estão descritos os principais casos de uso do módulo.

Quadro 7 - Descrição dos casos de uso

UC01 efetuar o *login*.

Permite ao usuário através da identificação por usuário e senha conectar-se ao sistema.

Ator: Usuário

Pré-condição: Usuário deve estar cadastrado no banco de dados.

Pós-condição: Usuário conectado ao sistema.

Cenários

Fluxo Principal:

- a) Usuário preenche seu *login* e sua senha;
- b) Sistema valida os dados de *login* e senha do usuário;
- c) Sistema direciona o Usuário para a página principal.

Fluxo Alternativo:

nome de usuário e/ou senha inválido(s): alerta com mensagem “usuário ou senha inválida” é mostrada.

UC02 alterar senha do *login*.

Usuário acessa a tela de alterar de senha informando seu *login*, a senha atual por ele utilizada para *login* no sistema, informa a nova senha desejada.

Ator: Usuário

Pré-condição: Usuário deve fazer *login* no sistema.

Pós-condição: Usuário alterou a senha de acesso.

Cenários

Fluxo Principal:

- a) Sistema apresenta a tela alterar senha.

Fluxo Alternativo:

nome de usuário e/ou senha inválido(s): alerta com mensagem “usuário ou senha inválida” é mostrada.

UC03 gerenciar pacientes.

Usuário acessa a tela cadastro de pacientes para informar dados do paciente. Serão cadastrados dados como: nome, data de nascimento, CPF e informações relevantes para controle médico.

Ator: Usuário

Pré-condição: Usuário deve fazer *login* no sistema.

Pós-condição: Usuário editou, apagou ou cadastrou um paciente.

Cenários

Fluxo Principal:

- a) sistema informa os pacientes cadastrados;
- b) usuário opta por alterar, apagar ou cadastrar um paciente. Ao cadastrar ou editar o paciente, o usuário poderá informar itens como nome, data de nascimento, CPF, se fuma, se bebe, hábitos...

Fluxo Alternativo:

campo(s) obrigatório(s) não preenchido(s): alerta com mensagem “Favor preencher todos os campos obrigatórios” é mostrada. Os campos obrigatórios são nome, data de nascimento e CPF.

UC04 gerenciar transtornos.

Usuário acessa a tela cadastro de transtornos para informar seus dados.

Ator: Usuário

Pré-condição: Usuário deve fazer *login* no sistema.

Pós-condição: Usuário editou, apagou ou cadastrou um transtorno.

Cenários

Fluxo Principal:

- a) sistema informa os transtornos cadastrados;
- b) usuário opta por alterar ou apagar um transtorno.

Fluxo Alternativo:

campo(s) obrigatório(s) não preenchido(s): alerta com mensagem “Favor preencher todos os campos obrigatórios” é mostrada.

UC05 gerenciar consultas.

Usuário acessa a tela cadastro de consultas para informar dados da consulta e anamnese.

Ator: Usuário

Pré-condição: Usuário deve fazer *login* no sistema.

Pós-condição: Usuário editou ou apagou uma consulta.

Cenários

Fluxo Principal:

- a) usuário acessa o módulo de pacientes;
- b) usuário seleciona o paciente desejado;
- c) usuário insere uma nova entrevista clicando no botão correspondente;
- d) o sistema inicia as fases do RBC, executando a pesquisa;
- e) o sistema questiona o usuário se quer ampliar a base de conhecimento;
- f) o sistema cria uma nova consulta de acordo com a resposta do usuário.

Fluxo Alternativo 1:

- a) sistema informa as consultas cadastradas;
- b) usuário opta por alterar ou apagar uma consulta.

Fluxo Alternativo 2:

campo(s) obrigatório(s) não preenchido(s): alerta com mensagem “Favor preencher todos os campos obrigatórios” é mostrada.

APÊNDICE B – Dicionário de dados

Neste apêndice é apresentado o dicionário de dados das principais tabelas utilizadas no aplicativo, com informações de atributos, tipos, tamanhos, campos obrigatórios, descrições, chaves primárias e estrangeiras, ou *Primary Keys* (PK) e *Foreign Keys* (FK). Foram utilizados cinco tipos de atributos:

- a) *int*: para valores numéricos;
- b) *varchar*: para valores de texto;
- c) *date*: para valores do tipo data;
- d) *mediumtext*: para valores de texto de médio conteúdo;
- e) *longtext*: para valores de texto de grande conteúdo.

No Quadro 8 tem-se os campos utilizados para persistência dos dados armazenados no sistema responsável pelo controle de médicos.

Quadro 8 - apresenta o dicionário de dados da tabela "medicos"

| MEDICOS | | | | |
|--|----------------|---------|-------------|-------------------------------------|
| Serve para armazenar informações dos usuários do aplicativo. | | | | |
| Nome do atributo | Tipo | Tamanho | Obrigatório | Descrição |
| idMedico | <i>int</i> | 11 | Sim | Identificação única. |
| nomeMedico | <i>varchar</i> | 255 | Sim | Nome do médico. |
| emailMedico | <i>varchar</i> | 100 | Sim | E-mail do médico. |
| senhaMedico | <i>varchar</i> | 100 | Sim | Senha de acesso ao sistema. |
| unique_idMedico | <i>varchar</i> | 60 | Sim | Identificador único para o sistema. |

No Quadro 9 tem-se os campos utilizados para persistência dos dados armazenados no sistema responsável pelo controle de pacientes.

Quadro 9 - apresenta o dicionário de dados da tabela "pacientes"

| PACIENTES | | | | |
|---|----------------|---------|-------------|----------------------|
| Serve para armazenar informações dos pacientes do aplicativo. | | | | |
| Nome do atributo | Tipo | Tamanho | Obrigatório | Descrição |
| idPaciente | <i>int</i> | 11 | Sim | Identificação única. |
| nomePaciente | <i>varchar</i> | 255 | Sim | Nome do paciente. |

| | | | | |
|--------------------------------------|-------------------|----|-----|---|
| CPFPaciente | <i>varchar</i> | 45 | Sim | CPF do paciente. |
| dataNascimentoPa- ciente | <i>varchar</i> | - | Sim | Data de nascimento do paciente. |
| unique_idPaciente | <i>varchar</i> | 60 | Sim | Identificador único para o sistema. |
| medicoCadastrador- Paciente (FK) | <i>int</i> | 11 | Sim | Identificação do médico. |
| fumantePaciente | <i>int</i> | 1 | Não | Paciente é fumante? |
| bebePaciente | <i>int</i> | 1 | Não | Paciente bebe? |
| descCirurgiasAnteri- oresPaciente | <i>mediumtext</i> | - | Não | Descrição das Cirurgias anteriores do paciente. |
| descAlimentacaoPa- ciente | <i>mediumtext</i> | - | Não | Descrição da alimentação do paciente. |
| descHabitosPaciente | <i>mediumtext</i> | - | Não | Descrição dos hábitos do paciente. |

No Quadro 10 tem-se os campos utilizados para persistência dos dados armazenados no sistema responsável pelo controle de consultas.

Quadro 10 – apresenta o dicionário de dados da tabela “transtornospaciente”

| TRANSTORNOSPACIENTE | | | | |
|---|-----------------|---------|-------------|-------------------------------------|
| Serve para armazenar informações das consultas. | | | | |
| Nome do atributo | Tipo | Tamanho | Obrigatório | Descrição |
| idTransornoPaciente | <i>int</i> | 11 | Sim | Identificação única. |
| idPaciente (FK) | <i>int</i> | 11 | Sim | Identificador do paciente. |
| idTranstorno (FK) | <i>int</i> | 11 | Sim | Identificador do transtorno. |
| idMedico (FK) | <i>int</i> | 11 | Sim | Identificador do médico. |
| unique_idTranstor- noPaciente | <i>varchar</i> | 60 | Sim | Identificador único para o sistema. |
| entrevistaPaciente | <i>longtext</i> | - | Sim | Entrevista em texto realizada. |
| dataCadastro | <i>date</i> | - | Sim | Data da consulta. |

No Quadro 11 tem-se os campos utilizados para persistência dos dados armazenados no sistema responsável pelo controle de transtornos.

Quadro 11 - apresenta o dicionário de dados da tabela “transtornos”

| TRANSTORNOS | | | | |
|---|-----------------|---------|-------------|--|
| Serve para armazenar informações dos transtornos do aplicativo. | | | | |
| Nome do atributo | Tipo | Tamanho | Obrigatório | Descrição |
| idTranstornos | <i>int</i> | 11 | Sim | Identificação única. |
| Nome | <i>varchar</i> | 255 | Sim | Nome do transtorno. |
| Caracteristicas | <i>longtext</i> | - | Não | Texto das características do transtorno. |
| CaracteristicasPesquisa | <i>longtext</i> | - | Não | Texto refinado das características do transtorno. |
| PertubacoesAssociadas | <i>longtext</i> | - | Não | Texto das perturbações associadas do transtorno. |
| PertubacoesAssociadasPesquisa | <i>longtext</i> | - | Não | Texto refinado das perturbações associadas do transtorno. |
| DiagnosticoDiferencial | <i>longtext</i> | - | Não | Texto do diagnóstico diferencial do transtorno. |
| DiagnosticoDiferencialPesquisa | <i>longtext</i> | - | Não | Texto refinado do diagnóstico do transtorno. |
| Critérios | <i>longtext</i> | - | Não | Texto dos critérios de diagnóstico do transtorno. |
| CritériosPesquisa | <i>longtext</i> | - | Não | Texto refinado dos critérios de diagnóstico do transtorno. |
| TranstornoPai (FK) | <i>int</i> | 11 | Sim | Identificador do transtorno. |
| unique_idTranstorno | <i>varchar</i> | 45 | Sim | Identificador único para o sistema. |

APÊNDICE C – *Stop words*

Este apêndice apresenta as *stop words* escolhidas para o pré-processamento da pesquisa. O Quadro 12 apresenta todas as *stop words*.

Quadro 12 - *Stop words* escolhidas

| | | | | | | | | | |
|---------|----------|---------|----------|----------|----------|----------|-----------|----------|-----------|
| A | Agora | Ainda | Alguém | Algum | Alguma | Algumas | Alguns | Ampla | Amplas |
| Amplo | Amplos | Ante | Antes | Ao | Aos | Após | Aquela | Aquelas | Aquele |
| Aqueles | Aquilo | As | Até | Através | Cada | Coisa | Coisas | Com | Como |
| Contra | Contudo | Da | Daquele | Daqueles | Das | De | Dela | Delas | Dele |
| Deles | Depos | Dessa | Dessas | Desse | Desses | Destas | Destas | Deste | Destes |
| Deve | Devem | Devendo | Dever | Deverá | Deverão | Deveria | Deveriam | Devia | Deviam |
| Disse | Disso | Disto | Dito | Diz | Dizem | Do | Dos | E | É |
| Ela | Elas | Ele | Eles | Em | Enquanto | Entre | Era | Essa | Essas |
| Esse | Esses | Esta | Está | Estamos | Estão | Estas | Estava | Estavam | Estávamos |
| Este | Estes | Estou | Eu | Fazendo | Fazer | Feita | Feitas | Feito | Feitas |
| Feito | Feitos | Foi | For | Foram | Fosse | Fossem | Grande | Grandes | Há |
| Isso | Isto | Já | La | Lá | Lhe | Lhes | Lo | Mas | Me |
| Mesma | Mesmas | Mesmo | Mesmos | Meu | Meus | Minha | Minhas | Muita | Muitas |
| Muito | Muitos | Na | Não | Nas | Nem | Nenhum | Nessas | Nesta | Nestas |
| Ninguém | No | Nos | Nós | Nossa | Nossas | Nosso | Nossos | Num | Numa |
| Nunca | O | Os | Ou | Outra | Outras | Outro | Outros | Para | Pela |
| Pelas | Pelo | Pelos | Pequenas | Pequenas | Pequeno | Pequenos | Per | Perante | Podê |
| Pude | Podendo | Poder | Poderia | Poderiam | Podia | Podiam | Pois | Por | Porem |
| Porque | Posso | Pouca | Poucas | Pouco | Poucos | Primeiro | Primeiros | Própria | Próprias |
| Próprio | Próprios | Quais | Qual | Quando | Quanto | Que | Quem | São | Se |
| Seja | Sejam | Sem | Sempre | Sendo | Será | Serão | Seu | Seus | Si |
| Sido | Só | Sob | Sobre | Sua | Suas | Talvez | Também | Tampouco | Te |
| Tem | Tendo | Tenha | Ter | Teu | Teus | Ti | Tido | Tinha | Tinham |
| Toda | Todas | Todavia | Todo | Todos | Tu | Tua | Tuas | Tudo | Última |
| Últimas | Último | Últimos | Um | Uma | Umas | Uns | Vendo | Ver | Vez |
| Vindo | Vir | Vos | Vós | | | | | | |