

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**FRAMEWORK GENÉRICO BASEADO EM REDES
BAYESIANAS PARA CAPTURA DE EVIDÊNCIAS EM
SISTEMAS MÓVEIS**

ALEXANDRE BELLEZI JOSÉ

ORIENTADOR: HERVALDO SAMPAIO CARVALHO

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.DM - 328 A/08

BRASÍLIA/DF: FEVEREIRO – 2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FRAMEWORK GENÉRICO BASEADO EM REDES BAYESIANAS
PARA CAPTURA DE EVIDÊNCIAS EM SISTEMAS MÓVEIS

ALEXANDRE BELLEZI JOSÉ

DISSERTAÇÃO DE MESTRADO ACADÊMICO, SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

HERVALDO SAMPAIO CARVALHO, Dr. FMD/UNB
(ORIENTADOR)

ADSON FERREIRA DA ROCHA, Dr, ENE/UNB
(EXAMINADOR INTERNO)

MARIA IMACULADA MUNIZ BARBOZA JUNQUEIRA, Dr^a, FMD/UNB
(EXAMINADORA EXTERNA)

BRASÍLIA, 28 DE FEVEREIRO DE 2008.

FICHA CATALOGRÁFICA

JOSÉ, ALEXANDRE BELLEZI

FRAMEWORK GENÉRICO BASEADO EM REDES BAYESIANAS PARA CAPTURA DE EVIDÊNCIAS EM SISTEMAS MÓVEIS [Distrito Federal] 2008.

xi, 150p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1.Sistemas Móveis

2.Sistemas Vestíveis

3.Redes Bayesianas

4.Coleta de Evidências

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

JOSÉ, A. B. (2008). FRAMEWORK GENÉRICO BASEADO EM REDES BAYESIANAS PARA CAPTURA DE EVIDÊNCIAS EM SISTEMAS MÓVEIS. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM - 328 A/08 Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 150p.

CESSÃO DE DIREITOS

AUTOR: Alexandre Bellezi José.

TÍTULO: FRAMEWORK GENÉRICO BASEADO EM REDES BAYESIANAS PARA CAPTURA DE EVIDÊNCIAS EM SISTEMAS MÓVEIS.

GRAU: Mestre ANO: 2008

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Alexandre Bellezi José

Rua 04, Quadra 15, Lote 02, Alto da Boa Vista.
72960-000 Corumbá de Goiás – GO – Brasil.

AGRADECIMENTOS

Agradeço a Deus por cada dia vivido e cada conquista alcançada.

Ao meu pai Adécio Rissi José, minha mãe Angela Maria Bellezi José e ao meu irmão André Bellezi José por acreditarem em mim.

Ao professor Hervaldo Sampaio Carvalho pelo incentivo acadêmico e confiança.

E ao professor Adson Ferreira da Rocha por ajudar na caminhada e escolha dos trajetos pelo mestrado.

Dedicado à minha família.

RESUMO

FRAMEWORK GENÉRICO BASEADO EM REDES BAYESIANAS PARA CAPTURA DE EVIDÊNCIAS EM SISTEMAS MÓVEIS

Autor: Alexandre Bellezi José

Orientador: Hervaldo Sampaio Carvalho

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 28 de Fevereiro de 2008.

O objetivo deste trabalho foi criar um *framework* genérico baseado em Redes Bayesianas para captura de evidências em sistemas móveis cuja prova de conceito será demonstrada por meio da implementação de um sistema de coleta de sintomas, chamado Goitacá. Com o uso do Goitacá, mostra-se como é possível tornar o indivíduo um agente ativo no processo de coleta de sintomas, através da resposta de um questionário informatizado referente a revisão dos sistemas, utilizando-se telefones celulares como integrantes de um sistema vestível.

ABSTRACT

FRAMEWORK GENÉRICO BASEADO EM REDES BAYESIANAS PARA CAPTURA DE EVIDÊNCIAS EM SISTEMAS MÓVEIS

Author: Alexandre Bellezi José

Supervisor: Hervaldo Sampaio Carvalho

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 28 de Fevereiro de 2008.

The aim of this work was to create a generic framework based on Bayesian networks for capture of evidence in mobile systems whose proof of concept will be demonstrated through the implementation of a system for collection of symptoms, called Goitacá. The use of Goitacá shows how you can make the person an active agent during the collection of symptoms, through the answer of a computerized questionnaire regarding the revision of systems, using cell phones as members of a wearable system.

SUMÁRIO

1 - INTRODUÇÃO.....	13
1.1-Introdução ao problema.....	13
1.2-Problema.....	14
1.3-Justificativa.....	15
1.4-Objetivo Geral.....	16
1.4.1 - Objetivos Específicos.....	16
1.5-Metodologia.....	16
1.6-Limitações.....	17
1.7-Organização do Trabalho.....	17
2 - TRABALHOS RELACIONADOS E REDES BAYESIANAS.....	18
2.1-Redes Bayesianas.....	27
2.2-Formalização das Redes Bayesianas.....	28
2.3-Etapas para criação manual de Redes Bayesianas.....	29
3 - METODOLOGIA.....	31
3.1-Fage e sua arquitetura.....	31
3.2-A Escolha das Tecnologias: Linguagens de programação.....	35
3.3-Ferramenta de Modelagem.....	37
3.3.1 - MSBNx.....	37
3.4-Modelando Interrogatórios.....	39
4 - IMPLEMENTAÇÃO DO FRAMEWORK.....	44
4.1-Coleta de Evidências: Algoritmos em Grafos.....	44
4.1.1 - Busca em Profundidade.....	44
4.1.2 - Busca em Largura.....	45
4.1.3 - Algoritmo híbrido de caminhamento.....	46
4.2-As Classes do Framework.....	47
4.2.1 - A Classe TradutorDSC.....	47
4.2.2 - A classe nodo.....	48
4.2.3 - A classe GAD.....	50
5 - O SISTEMA GOITACÁ.....	53
5.1.1 - O Interrogatório Sintomatológico.....	53
5.2-A coleta de sintomas e sinais no Goitacá.....	58
5.3-A sincronização de dados.....	58
6 - CONCLUSÃO.....	61
6.1-Dificuldades e limitações do projeto.....	62
6.2-Trabalhos futuros.....	62
REFERÊNCIAS BIBLIOGRÁFICAS.....	64
APÊNDICE A.....	68
APÊNDICE B.....	75
APÊNDICE C.....	80
APÊNDICE D.....	83
APÊNDICE E.....	93

LISTA DE TABELAS

Tabela 1: Linguagens de programação e pervasividade.....	35
--	----

LISTA DE FIGURAS

Figura 2.1 - Relação entre Computação Ubíqua, Pervasiva e Móvel. FONTE:[29].....	17
Figura 2.2. - Arquitetura Clássica de uma RSSF.....	18
Figura 2.3. - Sistema Vestível: (a) Sem Fusão de Dados; (b) Com Fusão de Dados.....	19
Figura 2.4. - Sistema Vestível em modo de operação constante. FONTE:[28].....	19
Figura 2.5. - Monitor de Glicose não invasivo e vestível.....	20
Figura 2.6. - Eyetap, (esquerda); Retângulo verde sobreposto a imagem real e texto sendo ampliado(direita). FONTE:[39].....	21
Figura 2.7. - HealthGear. Fonte:[25].....	22
Figura 2.8. - Sistema vestível de auxílio a deficientes visuais. Fonte:[26].....	22
Figura 2.9. - HandMed teste MiniMental. FONTE:[27].....	23
Figura 2.10. - MedPlot. FONTE:[40].....	23
Figura 2.11. - Sistema Vestível do Projeto MIThril.....	24
Figura 2.12. - Wearable Health Monitoring Systems. Fonte:[41].....	24
Figura 2.13. - Diagrama geral do projeto GIMPA. Modificado de [43].....	25
Figura 2.14. - Modelo de programação para BSN. FONTE:[45].....	26
Figura 2.15. - Rede Bayesiana.....	28
Figura 3.1. - FAGE visão geral da arquitetura.....	31
Figura 3.2. - Uso genérico do framework: Cenário para BSN (a); Cenário agrícola (b).....	33
Figura 3.3. - Interface da ferramenta MSBNx.....	37
Figura 3.4. - Representação do nodo no arquivo dsc.....	38
Figura 3.5. - Representação da tabela de probabilidade do nodo.....	38
Figura 3.6. - Representação geométrica do grafo.....	39
Figura 3.7. - Representação geométrica do grafo das questões.....	40
Figura 3.8. - Grafo com nova questão de nível 1.....	41
Figura 3.9. - Grafo do questionário com raiz.....	42
Figura 4.1. - Busca em profundidade.....	44
Figura 4.2. - Busca em largura.....	44
Figura 4.3. - Busca híbrida.....	45
Figura 4.4. - Diagrama de classes.....	47
Figura 4.5. - Estruturas de dados da classe nodo.....	47
Figura 4.6. - Estruturas de dados da classe GAD.....	50
Figura 5.1. - Parte do Interrogatório Sintomatológico.....	53
Figura 5.2. - Representação gráfica de parte do Interrogatório Sintomatológico.....	54
Figura 5.3. - Propriedades do Nodo e Exibição das Evidências.....	54
Figura 5.4. - Caso de Uso contexto geral.....	55
Figura 5.5. - Sistema GSWeb.....	56
Figura 5.6. - Caso de Uso sincronização.....	56
Figura 5.7. - Exemplo de Interface. (a) Telefone Celular (b) Palm.....	57
Figura 5.8. - Tela do Diário de Sintomas.....	58
Figura 5.9. - XML de Representação dos Sintomas e Sinais.....	59

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

Sigla	Significado
RSSF	Rede de Sensores Sem Fio
BSN	<i>Body Sensor Network</i>
J2ME	<i>Java Micro Edition</i>
HUB	Hospital Universitário de Brasília
UML	<i>Unified Modeling Language</i>
RFID	<i>Radio-Frequency Identification</i>
PDA	<i>Personal Digital Assistant</i>
ECG	Eletrocardiograma
EMG	Eletromiograma
GSR	Resistência Galvânica da Pele
C++	Linguagem C Orientada por Objetos
SO	Sistema Operacional
KVM	Kilobyte Virtual Machine
MSBNx	<i>Microsoft Belief Networks</i>
dsc	Extensão do arquivo no <i>Microsoft Belief Network</i>
GAD	Grafo Acíclico e Dirigido
XML	<i>eXtensible Markup Language</i>

1 - INTRODUÇÃO

Neste capítulo apresenta-se o objeto de estudo desta dissertação, como o mesmo se relaciona com os sistemas de monitoramento pessoal da saúde, a justificativa para a realização do trabalho e a forma de como o restante do texto está organizado.

1.1- Introdução ao problema

Atualmente vivencia-se um processo de mudança no paradigma de cuidado da saúde, o paciente torna-se um indivíduo ativo capaz de participar da manipulação e atualização de seus registros clínicos, normalmente mantidos somente em centros especializados em tratamento de doenças, como hospitais públicos e clínicas particulares. Ocorre ainda uma transição do objetivo primário desses centros de tratamento da saúde, com a participação mais ativa do indivíduo é possível pensar em prevenção de males da saúde ao invés de somente tratar problemas já plenamente estabelecidos.

Tais fatos só são possíveis devido aos avanços tecnológicos guiados por uma interdisciplinariedade cada vez maior de áreas do conhecimento humano, como por exemplo Computação, Engenharia e Medicina. Isto torna viável a concepção de sistemas completos para a monitoração pessoal da saúde. Duas abordagens são utilizadas hoje em dia na monitoração da saúde do indivíduo: monitoração em casa e monitoração no corpo humano. A monitoração em casa é obtida através do uso de casas especialmente projetadas para coletar informações sobre seus ocupantes. Atualmente, estas aplicações são mais utilizadas para idosos [30], e obviamente tal monitoramento só é realizado enquanto os indivíduos encontram-se presentes na casa. Naturalmente, para conseguir uma monitoração constante de indivíduos com um ritmo de vida comum, faz-se necessário utilizar outro meio de monitoração, que esteja sempre presente e que cause o menor impacto no estilo de vida do portador do sistema. Chega-se aqui aos sistemas de monitoramento da saúde vestíveis [31, 32]. Nota-se na abordagem que utiliza a residência como meio de monitoramento que há uma semelhança com o atual paradigma de cuidado

da saúde, onde os indivíduos se dirigem aos centros médicos para obter auxílio. Já no modelo baseado em sistemas vestíveis há uma real troca de paradigma. O paciente agora é visto como indivíduo, pois não precisa necessariamente estar doente, e passa a ser o centro de todo o sistema, pois a monitoração pode ocorrer em qualquer local e em qualquer momento. Assim os cuidados da saúde podem ser direcionados para o controle de fatores de risco, ou seja a prevenção primária, ou mesmo evitar o aparecimento de complicações decorrentes da doença, prevenção secundária.

O objetivo deste trabalho foi criar um *framework* genérico baseado em Redes Bayesianas para captura automática de evidências em sistemas moveis cuja prova de conceito será demonstrada por meio da implementação de um sistema de coleta de sintomas, chamado Goitacá¹. Com o uso do Goitacá, mostra-se como é possível tornar o indivíduo um agente ativo no processo de coleta de sintomas, através da execução e resolução de um questionário informatizado referente a revisão dos sistemas, utilizando-se telefones celulares como integrantes de um sistema vestível.

1.2- Problema

Um sistema de monitoramento vestível tem como principal característica permitir a mobilidade dos usuários durante a operação do sistema, assim como garantir que o usuário do sistema tenha em mãos as informações pertinentes ao que o sistema se propõe a monitorar. Devido a grande proximidade entre usuário e sistema, surgiu a idéia de usar os sistemas vestíveis para monitorar sinais vitais de seus usuários [12,16], podendo então tornar tais sistemas aliados na prevenção e no tratamento diário das mais diversas doenças.

Além disto um sistema vestível deve prover meios de interação entre o usuário do sistema e o próprio sistema, tal interação em muitos casos pode ser feita através de um conjunto de evidências fornecidas pelo usuário ao sistema através da resolução de um questionário.

Surge no cenário descrito acima um problema, como criar um modelo computacional capaz de ser aplicado a diferentes domínios, ou seja mais genérico, passível de ser usado em situações onde a coleta de evidências através da resolução de um questionário possa

¹ Do Tupi-guarani nômade, errante, que não se fixa em nenhum lugar.

trazer benefícios. Em Araújo [29], vê-se que sistemas ubíquos possuem como um dos principais requisitos o uso de *framework*, ou seja, uma estrutura básica para resolver uma certa classe de problemas. Assim, neste trabalho, o foco principal de estudo e desenvolvimento será a criação de um *framework* que será capaz de ser executado em aparelhos, tipo telefones celulares que suportem a tecnologia J2ME, *Java Micro Edition*.

1.3- Justificativa

As aplicações para sistemas ubíquos possuem requisitos especiais, e o uso de um *framework* que trate de um cenário genérico é de grande valia para a criação de novas aplicações que possuam a característica de resolução de questionários para coleta de evidências. Como técnica computacional de raciocínio evidencial existem as Redes Bayesianas, que podem ser manipuladas em *software* gráficos [6, 9]. Facilitando com isso a modelagem da rede e garantindo em sua composição possíveis conhecimentos probabilísticos sobre a relação das perguntas.

Assim, o uso deste *framework* facilitará a construção desses sistemas e por ser totalmente implementado em J2ME, possuirá a pervasividade necessária para uso em diversos tipos de dispositivos portáteis. Propiciando assim que estes tipos de dispositivo venham a ser usados como nodos do tipo *sink* em uma Rede de Sensores Sem Fio (RSSF), pois a grande maioria possui sistemas de transmissão de dados e com a futura criação de um novo padrão de *bluetooth* de baixo consumo [4], poderá permitir que tais aparelhos se comuniquem com nós sensores [2, 3] com a mesma tecnologia.

A atual conjuntura da medicina e informática já alcançada no Brasil, principalmente com o uso de prontuários informatizados, torna viável a concepção de sistemas onde o próprio paciente pode agir sobre os dados de seu próprio prontuário de maneira ativa. Em um cenário de inserção tecnológica seria plenamente concebível o uso de uma *Body Sensor Network* (BSN), nos sistemas de saúde. Hoje é possível pensar em uma aplicação móvel, capaz de coletar os sintomas, sincronizar tais informações com um sistema de prontuário eletrônico e agregar uma indumentária contendo uma BSN.

O *framework* aqui construído e sua aplicação real no sistema Goitacá não possuem correspondentes na literatura, sendo que os mesmos já permitiram a publicação de trabalhos em eventos nacionais e internacionais [33, 34].

1.4- Objetivo Geral

Desenvolvimento de um *framework* para coleta de evidências definidas na forma de questões hierárquicas e modeladas em uma Rede Bayesiana, capaz de ser executada em dispositivos móveis tipo telefone celular.

1.4.1 - Objetivos Específicos

No decorrer desta dissertação, postula-se:

- Apresentar o Interrogatório Sintomatológico e como pode-se transformá-lo em uma Rede Bayesiana, estendendo a metodologia para qualquer interrogatório.
- A definição e especificação do *framework* genérico.
- Uso do *framework* em um sistema de monitoração de pacientes desenvolvido no Hospital Universitário de Brasília (HUB).

1.5- Metodologia

Neste trabalho foi adotada a metodologia inicial de levantamento de textos de referências para compreensão dos problemas, reuniões em grupo, levantamento de requisitos sobre o Interrogatório Sintomatológico, definição do escopo do projeto do *framework*, criação do protótipo inicial de validação do *framework* e como prova de conceito foi feito o uso do

framework em um sistema móvel de monitoração de pacientes no HUB chamado de Goitacá.

Toda a parte de modelagem e levantamento de requisitos para a criação de *software* neste trabalho faz uso de notações definidas na *Unified Modeling Language* (UML) [10].

1.6- Limitações

Este trabalho tem seu escopo na construção e aplicação de um *framework* para coleta de evidências mapeadas em uma Rede Bayesiana que representa um conjunto de questões hierárquicas. A avaliação de sua utilização prática deverá ser realizada por meio de um estudo prospectivo sobre o impacto da tecnologia e da aplicação da mesma na prevenção primária e secundária de doenças.

1.7- Organização do Trabalho

Este trabalho está organizado em 6 capítulos. Uma descrição sumária de cada capítulo é apresentada a seguir:

O Capítulo 1 é a introdução do trabalho.

O Capítulo 2 é utilizado para um levantamento de trabalhos relacionados e suas aplicações. Também é abordado neste capítulo uma introdução às Redes Bayesianas.

O Capítulo 3 descreve toda a metodologia adotada.

O Capítulo 4 trata do processo de criação do *framework*

O Capítulo 5 exibe e descreve a aplicação do *framework* no sistema Goitacá capaz de executar o Interrogatório Sintomatológico.

O Capítulo 6 condensa as conclusões, limitações do trabalho e trabalhos futuros.

2 - TRABALHOS RELACIONADOS E REDES BAYESIANAS

Os sistemas de monitoramento da saúde humana vestíveis podem ser classificados como uma aplicação real da Computação Pervasiva, uma forma de computação onipresente que graças a miniaturização dos componentes computacionais e de sensoriamento pode ser inserida em objetos do dia a dia, como telefones celulares, brinquedos, casas, móveis, roupas, etc [8,29].

No parágrafo acima nota-se que os sistemas de monitoramento da saúde humana possuem dois termos chave: a mobilidade do sistema e sua pervasividade. Tal fato permite classificá-los como um sistema ubíquo. Na Figura 2.1 temos um diagrama de conjuntos onde a intersecção do conjunto de sistemas móveis com sistemas pervasivos gera uma gama de sistemas ubíquos, ou seja, um sistema ubíquo possui características dos sistemas móveis e pervasivos.

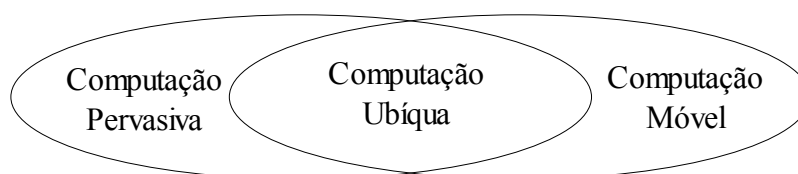


Figura 2.1 - Relação entre Computação Ubíqua, Pervasiva e Móvel.

FONTE:[29]

Há uma convergência cada vez maior para o uso de Redes de Sensores Sem Fio (RSSF) para o monitoramento humano, as chamadas *Body Sensor Network* (BSN) são um exemplo disso [14, 32, 35]. O uso das RSSF garantem a mobilidade e pervasividade que esses sistemas necessitam. A Figura 2.2 mostra uma arquitetura clássica de uma RSSF.

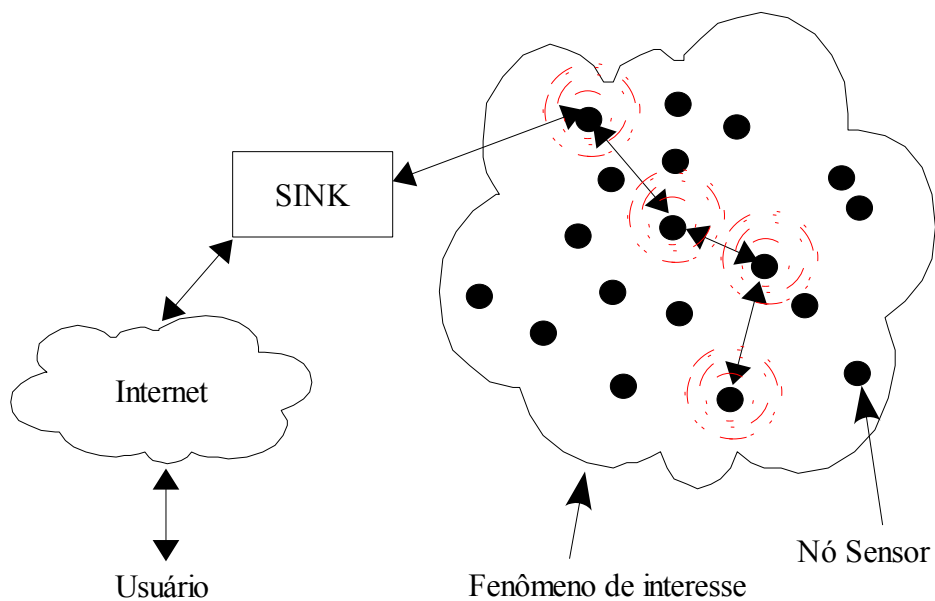


Figura 2.2. - Arquitetura Clássica de uma RSSF.

Da arquitetura clássica da RSSF tem-se o fenômeno de estudo em si, como por exemplo um campo de batalha, plantações, áreas de acidentes ambientais, um organismo vivo, etc. Como elo de interconexão entre a RSSF e a outras redes de comunicação, como por exemplo a Internet, há um nó chamado *sink*, que atua como um *gateway*. E finalmente o usuário a qual se destina a informação proveniente da rede. Os nós sensores não só obtém dados do objeto em monitoramento, como também podem gerar uma informação de maior valor semântico, como definido em [36], a essa capacidade de agrupar dados e extrair informação de maior qualidade [37] dá-se o nome de fusão de dados.

Quando pensa-se em sistemas vestíveis de monitoramento humano, inicialmente um dos consumidores das informações geradas é o próprio portador do sistema, neste ponto o uso de fusão de dados torna-se um aliado importante para o usuário, como mostrado na Figura 2.3. Pois o usuário não precisa ser necessariamente sobrecarregado com dados pontuais de sua condição física, ao invés disso o mesmo pode obter um visão sobre seu estado físico atual, ou seja, uma avaliação automatizada.

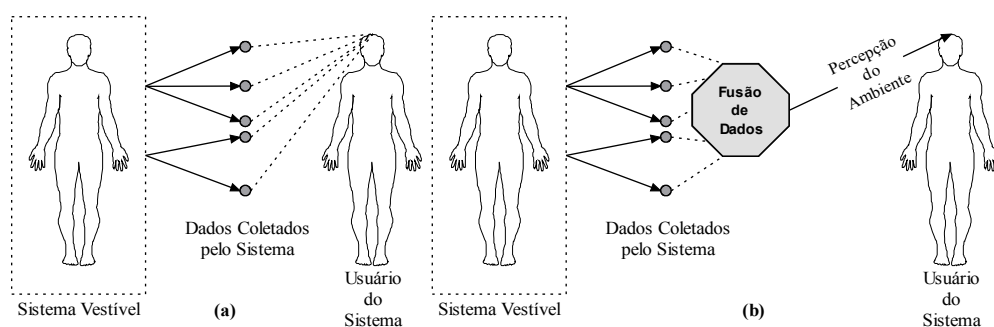


Figura 2.3. - Sistema Vestível: (a) Sem Fusão de Dados; (b) Com Fusão de Dados

Infelizmente há alguns aspectos que ainda não são passíveis de sensoriamento, pois dependem de uma avaliação até mesmo subjetiva do próprio portador do sistema. Por exemplo como um sensor poderia inferir sobre quantidade de dor de um paciente, ou mesmo qual o tipo de dor? Surge então um problema nesse tipo de sistema de monitoramento que deve ser solucionado: como obter tais informações subjetivas que só o usuário as possui? A resposta, neste caso, vem da própria medicina, através do uso de uma ferramenta chamada Interrogatório Sintomatológico, ou seja, pergunta-se ao usuário o que ele está sentindo.

Aqui neste ponto, pode-se fazer uma constatação. Se inserirmos em uma BSN um nodo sensor com capacidade de executar questionamentos aos usuários, o usuário do sistema, quando interagir com esse nodo, irá se tornar uma espécie de “nó sensor” pois através de suas respostas estará fornecendo novas informações ao sistema (*human-in-the-loop*). Em [28] vê-se uma taxonomia de classificação de sistemas vestíveis criada para ajudar a compreender suas características, o trabalho aqui proposto se enquadra na categoria de sistemas com tipo de operação constante, assim o dispositivo estará “sempre ligado” e acessível ao usuário, formando um sistema circular de troca de informações entre o computador e usuário, como visto na Figura 2.4.

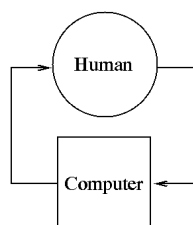


Figura 2.4. - Sistema Vestível em modo de operação constante. FONTE:[28]

É possível ainda generalizar a utilização de nós sensores capazes de questionar humanos para uso em RSSF e não só restringi-lo as BSN, pois podem haver situações onde somente uma avaliação feita por uma pessoa possa ser válida.

Por exemplo em um ambiente de computação pervasiva e com sensibilidade ao contexto por onde o usuário passe, caminhe, ele pode ser reconhecido por sistemas de informação que desejam lhe enviar informações. O usuário deve então responder, ou interagir, de alguma maneira com estes sistemas e informar se deseja ou não as informações vinda destes “exosistemas computacionais”.

O desenvolvimento tecnológico relacionado a medicina sempre chama a atenção de uma maneira quase que instintiva, afinal todos querem viver mais e com melhor qualidade. Nesta nova visão, a troca de paradigma no sistema de saúde, onde o paciente agora passa a ser um indivíduo ativo e o centro do sistema, aliada a constante preocupação econômica de baixar custos são os principais determinantes das novas políticas. Desta forma, os grandes centros de atenção a saúde e os seus gestores estão cada vez mais preocupados com a prevenção primária de doenças.

Surge neste contexto um grande conjunto de soluções com o intuito de monitorar os indivíduos, para a prevenção primária de doenças, ou mesmo secundária onde doenças já estabelecidas podem ter seus efeitos controlados.

A diabete, por exemplo, é uma doença onde a monitoração constante das taxas de glicose no sangue é fundamental para garantir o sucesso da terapia. Na maioria das vezes os pacientes fazem uso de medidores de glicose sendo necessário o uso de uma pequena gota de sangue. Para ilustrar a convergência deste tipo de monitoramento para um sistema vestível, vê-se na Figura 2.5 o uso de um monitor de glicose não invasivo na forma de um relógio de pulso, o GlucoWatch [38].



Figura 2.5. - Monitor de Glicose não invasivo e vestível.

Ainda na mesma linha de monitores não invasivos que tendem a fundir-se nas indumentárias humanas, temos os conhecidos medidores de frequência cardíaca, muito utilizado por praticantes de atividades físicas.

O uso de óculos é talvez uma das inserções de aparelhos mais presentes no nosso dia a dia. Deixando os aspectos estéticos de lado, pode-se dizer que tais dispositivos são essenciais na melhora da qualidade de vida de milhares de pessoas com problemas em sua acuidade visual. Mesmo esses dispositivos tão clássicos, não estão imunes aos efeitos da pervasividade da computação sobre objetos do dia a dia. Em um dos trabalhos realizados por Steve Mann vê-se a criação de um dispositivo chamado Eyetap [39]. Uma espécie de tapa olho de alta tecnologia, que é capaz de sobrepor dados virtuais sobre imagens do mundo real.

Tal sistema pode ser muito útil para ajudar pessoas com problemas de memória, pois podem sobrepor as imagens do mundo real “etiquetas textuais” informando sobre o objeto em evidência na cena. Além disso é possível o uso do mesmo para ampliar determinadas áreas da imagem. A Figura 2.6 dá uma visão geral do dispositivo e um exemplo de uma imagem sendo realçada. É importante salientar o nível de proximidade da interface homem e máquina neste caso.

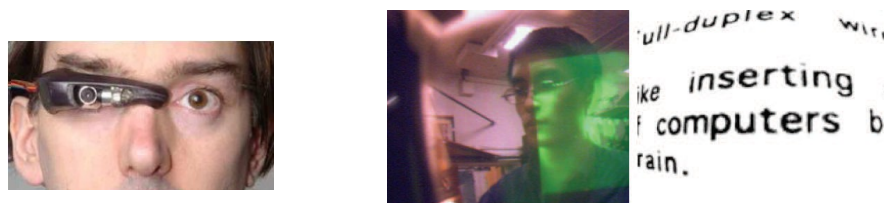


Figura 2.6. - Eyetap, (esquerda); Retângulo verde sobreposto a imagem real e texto sendo ampliado(direita). FONTE:[39]

O projeto *HealthGear* [25] já acrescenta uma funcionalidade extra aos telefones tipo *SmartPhones* com sistema operacional *Windows Mobile 2003*, permite a comunicação via *bluetooth* entre o telefone e um sensor, no caso do projeto foi utilizado um sensor de saturação de oxigênio no sangue e um sensor de pulso.

Diferentemente do que ocorre com outros projetos de monitoração, onde os resultados só são conhecidos quando o usuário descarrega as informações coletadas durante o uso em um computador para ter a avaliação. O *HealthGear* faz o processamento de análise dos dados no próprio telefone celular. A Figura 2.7 mostra o *hardware* e arquitetura do projeto.

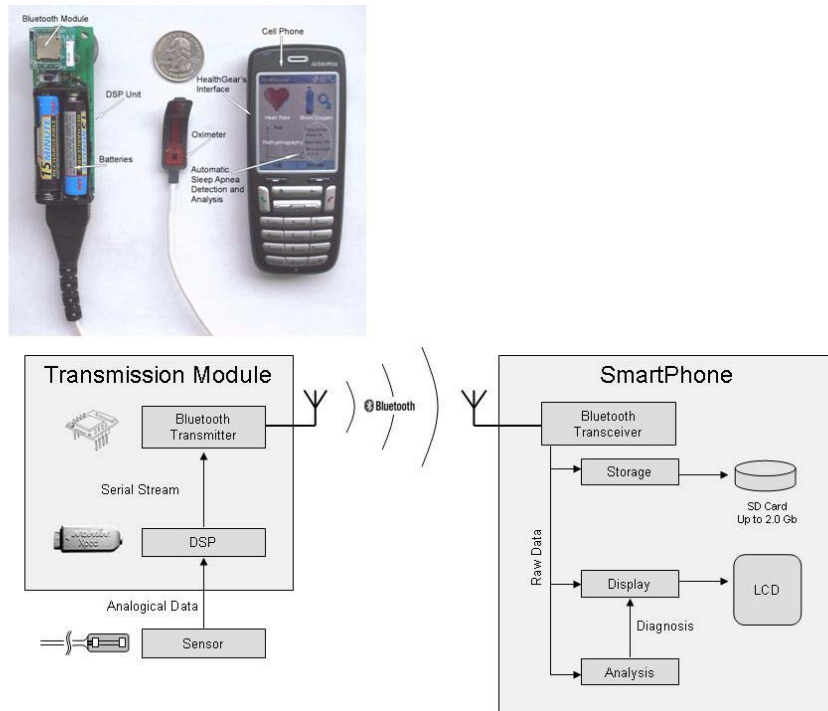


Figura 2.7. - *HealthGear*. Fonte:[25]

Em [26] há a proposta de um sistema vestível para uso por pessoas com deficiência visual. Tal sistema é composto por um *framework*, sistemas de identificação de objetos via *Radio-frequency identification* (RFID), Figura 2.8. O sistema gera para o usuário informações táteis de objetos que estão fora do seu alcance manual, através do uso de uma luva que faz a estimulação, através de vibrações, para representar as texturas dos objetos avaliados pelo sistema.



Figura 2.8. - Sistema vestível de auxílio a deficientes visuais. Fonte:[26]

Para a automação na captura de sintomas há o trabalho descrito em [27], onde os autores exibem uma aplicação para uso em *Personal Digital Assitant* (PDA), com sistema operacional Linux, visando facilitar a coleta de sintomas dos pacientes durante suas atividades normais. A Figura 2.9 mostra uma das tela do projeto HandMed.



Figura 2.9. - HandMed teste MiniMental. FONTE:[27]

Ainda na mesma linha de uso de PDA para monitoramento de sinais biomédicos em seres humanos, há o projeto MedPlot[40], onde foi construída uma aplicação capaz de representar graficamente os sinais fisiológicos medidos no paciente, Figura 2.10.

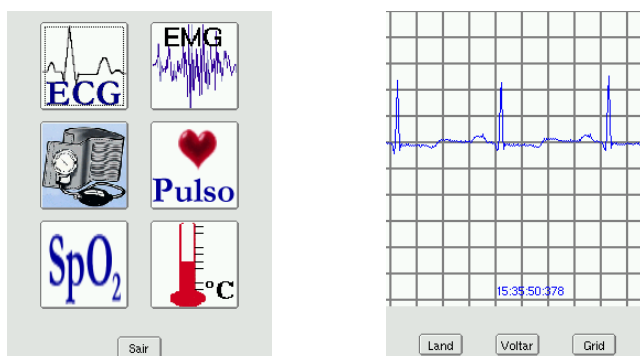


Figura 2.10. - MedPlot. FONTE:[40]

No quesito de sistemas vestíveis multi paramétricos tem-se o projeto MITHril que é composto por um traje com capacidade de sensorar eletrocardiograma (ECG), eletromiograma (EMG), a resistência galvânica da pele (GSR), temperatura e movimentos dos usuários. A Figura 2.11 exhibe o aspecto do sistema. No que se refere ao *software* possui um motor de inferência sobre o contexto de utilização do sistema, um tipo de *feedback* entre o ambiente de uso e as opções e ações no sistema [31].

O projeto é feito nos moldes *open-source*, tanto para o *hardware* quanto para o *software*. O que permite sua aplicação em outros projetos. Um dos pontos forte do projeto é sua real intenção de romper a barreira entre computação vestível e a vida diária das pessoas[31].

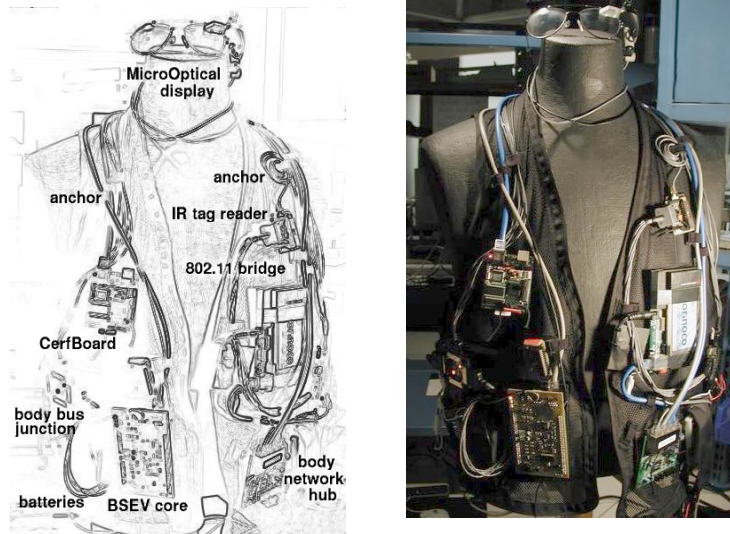


Figura 2.11. - Sistema Vestível do Projeto MITHril.

O projeto *Wearable Health Monitoring Systems* (WHMS) possui uma abrangente arquitetura e já abre a possibilidade de que o módulo chamado de *personal server* possa ser um PDA ou um telefone celular, como exibido na Figura 2.12. Possui sensores para ECG, de movimentos e oximetria de pulso [41].

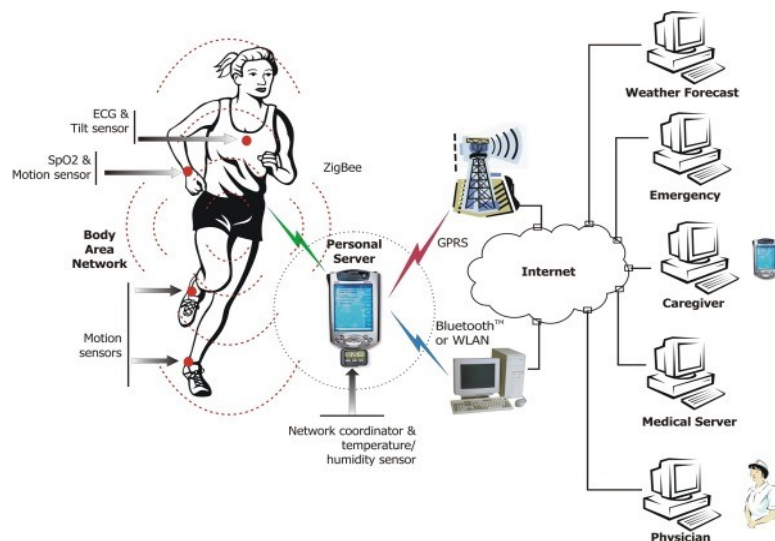
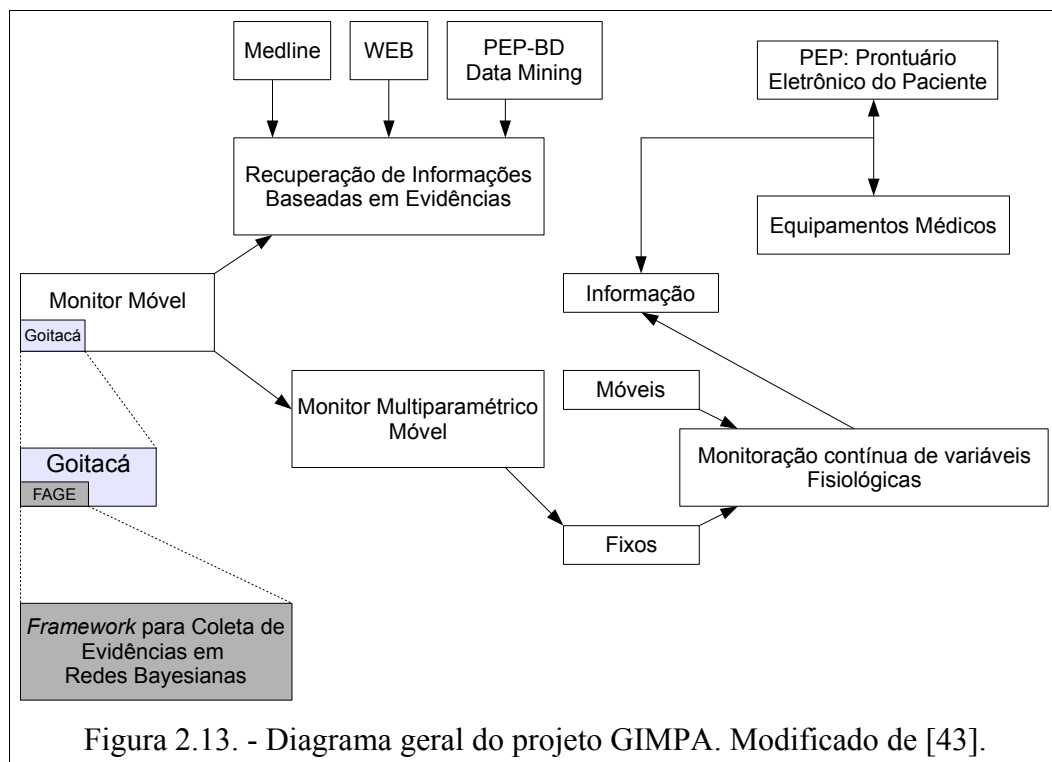


Figura 2.12. - *Wearable Health Monitoring Systems*. Fonte:[41]

Procurando unir sistemas móveis de monitoração (Computadores Vestíveis), Sistema de Informação na Web (WWW e Bibliotecas Digitais) e Prontuário Eletrônico do Paciente surgiu o projeto GIMPA [42]. O projeto é dividido em módulos, com visto na Figura 2.13.

No módulo de monitor multiparamétrico móvel, Carvalho[42] defende o uso de vários tipos de sensores assim como informações provenientes do próprio paciente.

O monitor móvel pode ser um sistema vestível ou mesmo uma BSN, assim tal módulo composto por uma Rede de Sensores Sem Fio para o Corpo Humano pode interagir de maneira dinâmica com os sistemas de informações de Saúde Eletrônicos, por exemplo solicitando uma ambulância em caso de problemas graves, atualizando constantemente sinais fisiológicos dentro do prontuário, gravando sintomas evidenciados pelos usuários, etc.



As RSSF possuem uma certa complexidade de programação de seus nós sensores, normalmente utilizando linguagens como C. Para resolver problemas referentes a reprogramação de BSN a distancia e facilitar a vida dos profissionais da área médica na reconfiguração de tais redes há o projeto de reconfiguração d uma rede de sensores do corpo humano à distancia por meio de uma interface gráfica [43, 44]. Neste sub-projeto do projeto *Body-worn Sensor Network* [14], tem-se um ambiente de programação dinâmica e visual desenvolvido para interação dos profissionais de saúde com os códigos sendo executados nos nodos sensores. A Figura 2.14 mostra parte da interface e arquitetura de reprogramação do nós sensores através do uso de um compilador inteligente.

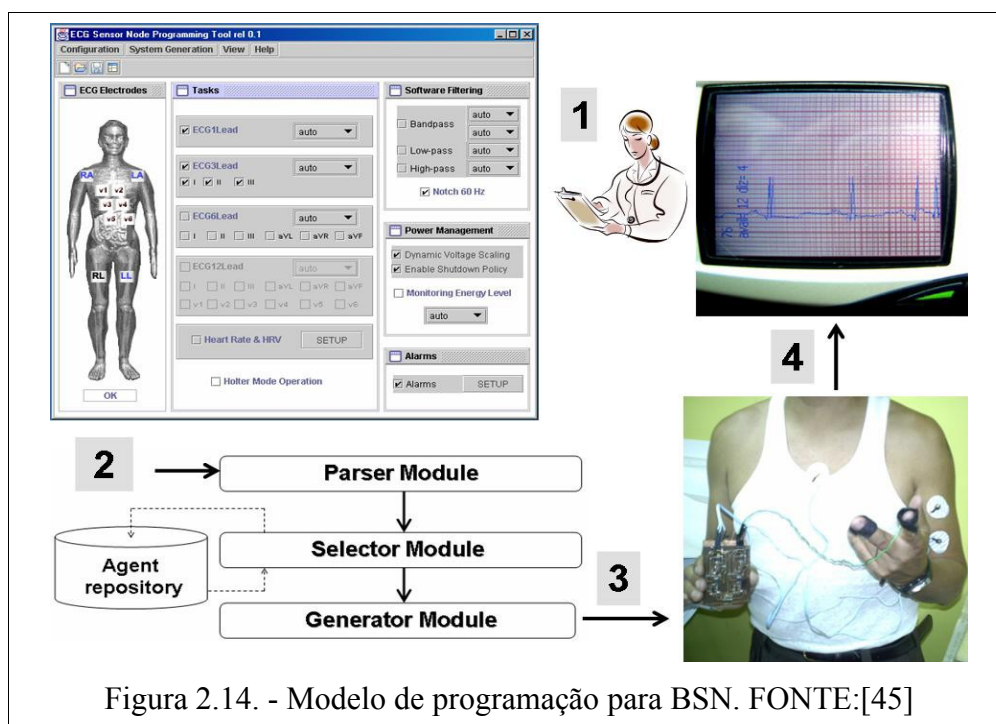


Figura 2.14. - Modelo de programação para BSN. FONTE:[45]

2.1- Redes Bayesianas

A teoria de decisão Bayesiana é usada para gerar um modelo probabilístico, ela permite determinar probabilidades condicionais de uma evidência *a priori*, já as probabilidades revisadas são chamadas probabilidades *a posteriori*. Redes Bayesianas podem ser criadas para lidar com problemas que possuem algum nível de incerteza, desde o estabelecimento de diagnósticos de doenças, seleção de tratamentos otimizado até na prescrição de tratamentos médicos [13].

Uma Rede Bayesiana pode ser representada através de uma estrutura gráfica associada a uma distribuição de probabilidade, ou seja, graficamente pode ser representada por um grafo acíclico e dirigido, onde os nodos do grafo estariam representando as variáveis do problema modelado e os arcos entre os nodos definem as relações de dependência probabilística entre as variáveis. Tais tipos de redes funcionam como módulos em sistemas de apoio a decisão, e são no caso de uma construção manual da rede, totalmente

dependentes de conhecimentos prévios de especialistas humanos no problema modelado. Uma construção automatizada da rede precisaria apenas de uma base de dados e algoritmos capazes de criar uma rede através dos dados [13]. A desvantagem da rede construída manualmente é o tempo gasto em entrevistas com especialistas para coleta de informações, mas resulta em uma rede de maior qualidade. Já uma rede automatizada pode dispensar o especialista, mas precisa de uma base de dados para análise e normalmente gera uma rede de qualidade inferior a manual.

2.2- Formalização das Redes Bayesianas

Formalmente um o grafo que representa a rede Bayesiana é assim definido:

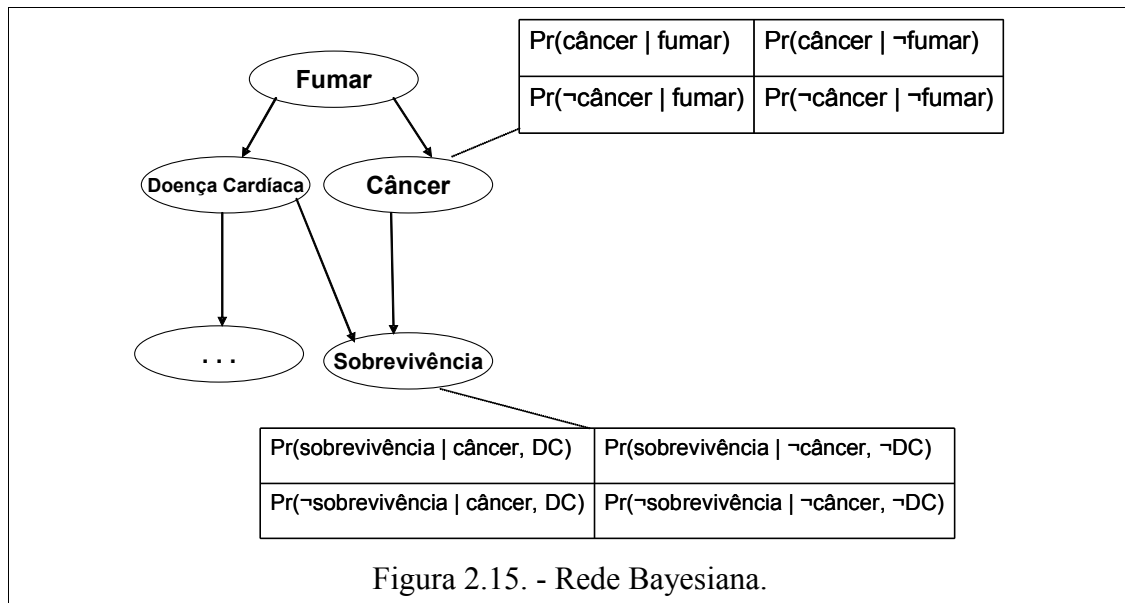
$$B=(Pr,G) \quad (1)$$

Onde em (1) temos:

$G=(V(G), A(G))$ definição do grafo G com nodos $V(G)=\{V_1, \dots, V_n\}, n \geq 1$ e arcos $A(G) \subseteq V(G) \times V(G)$.

Pr é a distribuição de probabilidade que atua sobre o grafo, onde para cada nodo V_i (variável) é atribuída um conjunto de distribuição de probabilidades condicionais, ou seja, $Pr(V_i | \pi(V_i))$. Onde $\pi(V_i)$ representa a combinação das probabilidades dos predecessores (pais) de V_i .

Na Figura 2.15, pode-se observar a estrutura geral de uma rede Bayesiana, vale ressaltar o conteúdo da variável Sobrevivência, na tabela de probabilidade atribuída ao nodo, nota-se a combinação de todas as probabilidades relacionadas aos acontecimentos dos eventos dos pais de V_i , ou seja $V_i = (\text{Sobrevivência ou } \neg\text{Sobrevivência})$ e os $\pi(V_i) = (\text{câncer, DC ou } \neg\text{câncer, } \neg\text{DC})$ e, sendo que as mesmas não representam, para efeitos de simplificação, os valores de probabilidade (0 até 1), somente termos lógicos referentes aos eventos.



2.3- Etapas para criação manual de Redes Bayesianas

Em [13] tem-se a enumeração e descrição de cinco etapas para a criação de uma Rede Bayesiana relacionada à medicina e cuidados médicos, sendo que tais etapas são listadas abaixo:

1. **SELEÇÃO DE VARIÁVEIS RELEVANTES:** como cada nó do grafo representa uma variável relevante do processo sobre consideração, faz-se necessário fazer um levantamento de todas as possíveis variáveis que compõem o processo em questão. Para tal costuma-se realizar entrevistas com especialistas na área do processo, levantando assim todas as possíveis variáveis.
2. **IDENTIFICAÇÃO DOS RELACIONAMENTOS ENTRE AS VARIÁVEIS:** após a identificação das variáveis que fazem parte do processo, é necessário analisar como tais variáveis se relacionam, ou seja, quais são as casualidades presentes em cada variável que fazem com que determinadas variáveis tenham ligações através de arcos e outras não. Normalmente essas casualidades estão relacionadas com o conhecimento do especialista sobre o acontecimento de determinados eventos, perguntas do tipo “*O que causa tal efeito?*” e “*Que*

manifestação essa causa pode ter?” são mostradas em [13] como boas opções para se extrair tais tipos de relacionamentos.

3. IDENTIFICAÇÃO DE PROBABILIDADES QUALITATIVAS E RESTRIÇÕES LÓGICAS: identificar o tipo de distribuição de probabilidade que rege o processo pode ajudar a avaliar e verificar as probabilidades requeridas para a construção da rede. Já as restrições lógicas ajudam a limitar o universo de probabilidades que tem de ser avaliadas pela rede, exemplo, se na Figura 2.15 as chances de sobrevivência só pudessem ser avaliadas se a pessoa \neg câncer e \neg DC, então a tabela de probabilidade associada ao nodo sobrevivência teria a metade do seu tamanho.
4. AVALIAÇÃO DAS PROBABILIDADES: nesse estágio a distribuição de probabilidade $\Pr(V_i|\pi(V_i))$ é atribuída a cada nó do grafo. Tais probabilidades são obtidas na maioria das vezes dos especialistas no processo sobre questão ou então podem ser levantadas a partir de um conjunto de dados sobre o processo.
5. ANÁLISE DE SENSIBILIDADE E AVALIAÇÃO: no quarto estágio a rede já está totalmente montada, o que falta então é verificar sua validade. Pode-se através de dados reais de pacientes fazer um teste de sensibilidade da rede para verificar se as distribuições de probabilidades subjacentes possuem alguma inexatidão. A avaliação pode ser feita utilizando-se dados de um paciente na rede recém criada, então as mesmas informações são submetidas a outro sistema probabilístico, comparando-se os resultados.

3 - METODOLOGIA

No início deste trabalho, que foi uma extensão do trabalho de [15], a idéia básica era construir uma aplicação capaz de executar o Interrogatório Sintomatológico de uma maneira mais eficiente no consumo de energia, pois quanto menor for o tempo gasto do usuário respondendo perguntas maior o tempo de vida da aplicação. Pois trata-se aqui de software para sistemas com limitação devido ao uso de baterias[33].

Entretanto no decorrer deste processo, notou-se que na criação de *software* para execução de diversos tipos de questionários, não somente referente ao Interrogatório Sintomatológico, seria interessante dividir as tarefas: o programador cria o sistema de informação que usa o *framework* e um profissional, da saúde por exemplo, verdadeiro conhecedor das relações entre perguntas e possíveis respostas modelaria a Rede Bayesiana. Assim surge o *Framework* para Coleta Automatizada de Evidências, FAGE[34, 45].

3.1- Fage e sua arquitetura

O *framework* desenvolvido neste trabalho possui a capacidade de manipular Redes Bayesianas que representem um conjunto de questões definidas de maneira hierárquica. A Figura 3.1 representa a arquitetura do *framework*, que é basicamente dividida em três componentes chaves.

A Rede Bayesiana normalmente construída por um especialista em alguma área do conhecimento, é usada como uma espécie de arquivo de configuração, que dirá ao software quais perguntas haverá no sistema, assim como suas possíveis respostas. Tal item é definido como domínio do especialista.

O bloco que engloba todas as classes do *framework* é chamado de FAGE, ele é responsável por prover toda a abstração na execução dos questionários em dispositivos móveis.

Dentro do bloco FAGE há uma classe chamada de *Bsearch*, que pode ser estendida pelo

Na Figura 3.2 pode-se ver a representação de tais cenários de uso. Observe que o celular pode funcionar como um sorvedouro de dados, *sink*, e posteriormente enviar os dados via *Internet* a alguma aplicação.

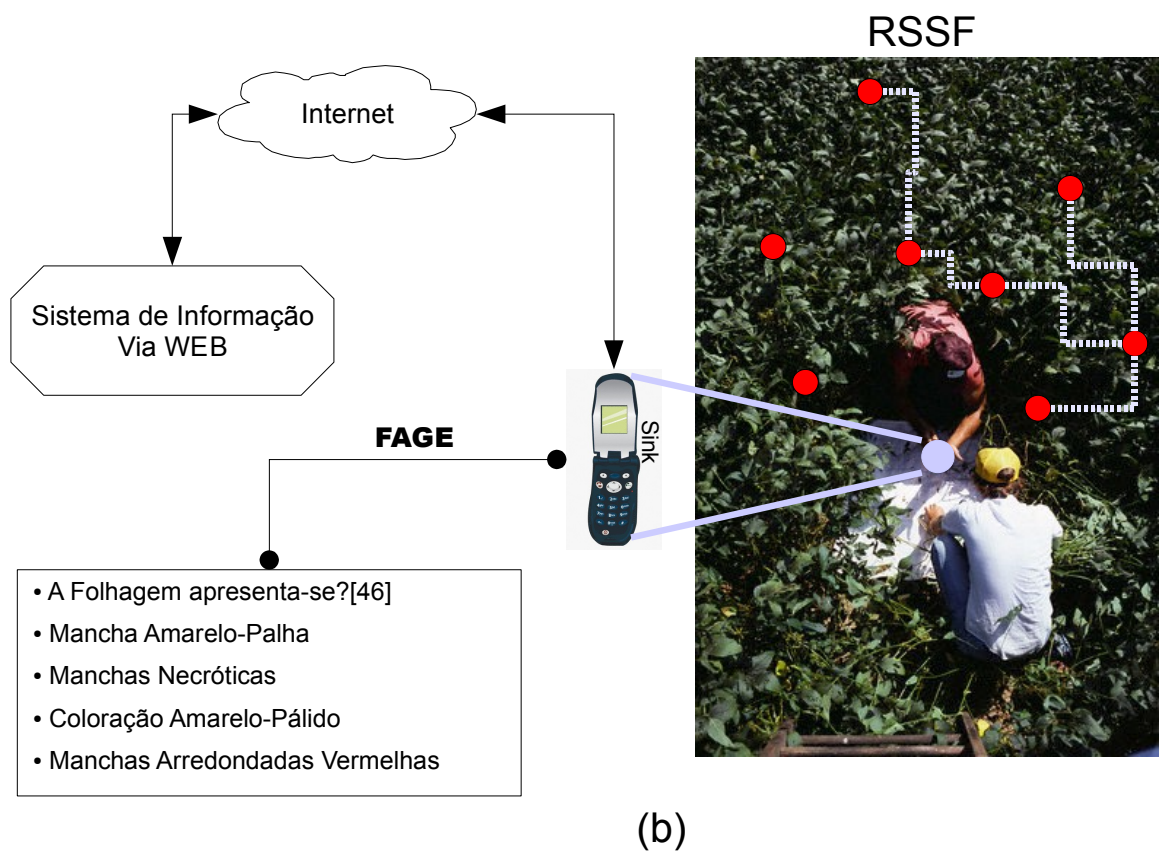
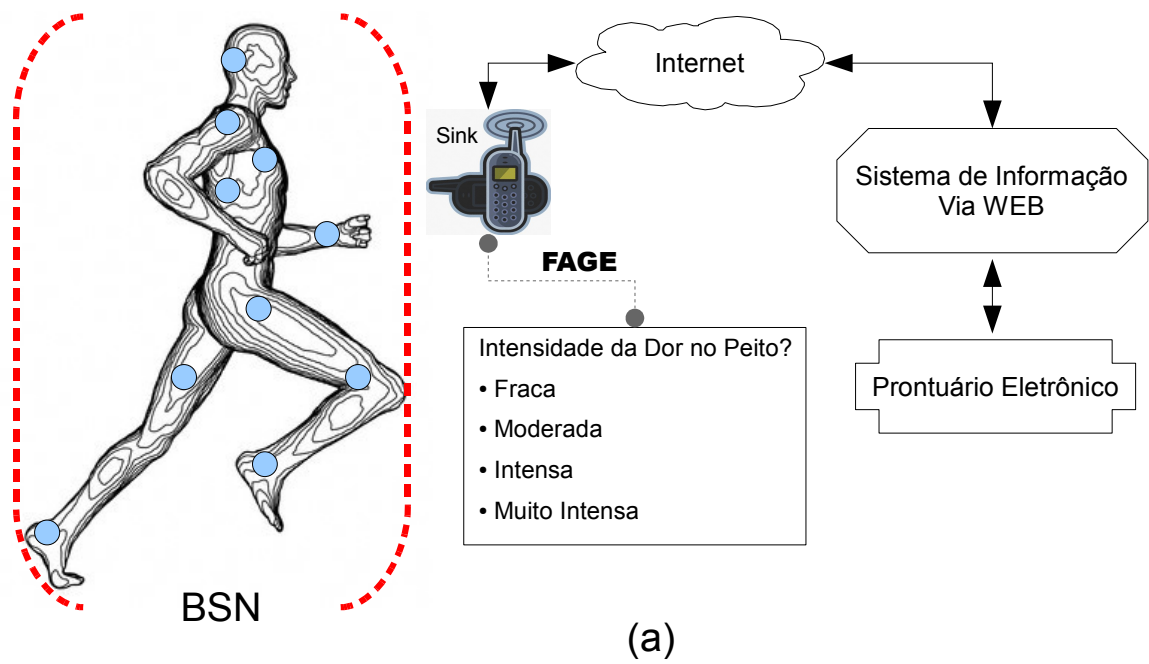


Figura 3.2. - Uso genérico do framework: Cenário para BSN (a); Cenário agrícola (b).

3.2- A Escolha das Tecnologias: Linguagens de programação

Quando se fala em desenvolvimento para dispositivos móveis tipo telefone celular, PDA, Smartphones, etc. Poucas são as tecnologias que podem alcançar uma grande gama destes equipamento. Basicamente tem-se: J2ME, C++(Symbian), Python, Flash Lite, Perl e Ruby. Não se pretende aqui realizar uma descrição detalhada de cada uma das linguagens listadas, maiores informações sobre cada uma das tecnologias pode ser encontrado consultando as referências indicadas.

Python é uma linguagem de programação interpretada, orientada a objetos, com tipagem dinâmica e forte. Possui código fonte aberto, permitindo sua inclusão mesmo em projetos comerciais. Atualmente só há disponibilidade de execução em telefones celulares da plataforma S60 da Nokia, o que é um problema, pois restringe a abrangência da linguagem[17].

Perl foi criada em 1987 é uma linguagem considerada estável, é multiplataforma e fundamentada em *scripts*. Muito utilizada em aplicações de bioinformática para detectar padrões de dados e especialmente seqüências de caracteres de textos [11]. É bastante eficiente do ponto de vista do programador, pois seu tempo de aprendizado é considerado baixo. Para seu uso em dispositivos móveis, é necessário usar um *port* da linguagem feito para rodar no sistema operacional (SO) *Symbian* ou WinCE [18]. Infelizmente o *Symbian* SO e o WinCe só estão disponíveis em uma classe de dispositivos definidos como *SmartPhones*, que ainda são poucos e com alto custo.

Ruby é uma linguagem de programação totalmente orientada a objetos, até os tipos primitivos são objetos, é interpretada, seu foco principal é a simplicidade do código possibilitando uma maior produtividade [19]. Também é dependente do SO *Symbian*, para uso em telefones celulares, restringindo assim seu alcance [20].

A maior vantagem de uso da linguagem *Flash Lite* é na criação de ricas interfaces com o usuário de uma maneira mais simples para o programador. Depende do SO *Symbian* ou *Windows Mobile* para ser executado, além de suas ferramentas de desenvolvimento serem pagas [21].

A linguagem C++ possui como maior qualidade, tratando aqui de dispositivos móveis, sua

capacidade de acessar diretamente todos os recursos do dispositivo, como câmera, rádios *bluetooth*, etc. O tempo de aprendizado é alto e a construção de aplicações também demanda mais tempo de trabalho [22]. Como o nome sugere é dependente do SO *Symbian* para ser executada.

*Java Micro Edition*³ é uma linguagem de programação projetada para lidar com os requisitos especiais dos dispositivos móveis, pois deve garantir que os programas feitos usando *Java* rodem em ambientes com pouca memória, tela limitante e restrições de energia. Basicamente possui as mesmas características da linguagem *JAVA* original, orientada a objetos, geração de *bytecodes* independentes de arquitetura, execução segura de código e dependência de uma máquina virtual para execução [24]. Comercialmente é a linguagem mais utilizada em dispositivos móveis, tal fato fez com que a maioria dos fabricantes de telefones inclua em seus aparelhos uma implementação da máquina virtual conhecida como Kilobyte Virtual Machine (KVM). Em [23] pode ser consultada uma lista com os dispositivos compatíveis com a KVM, onde praticamente todos os fabricantes possuem modelos compatíveis com J2ME.

A grande disponibilidade de aparelhos capazes de rodar os *bytecodes* garante uma maior disponibilidade do framework aqui proposto, assim a escolha de tecnologia de desenvolvimento recai sobre a tecnologia J2ME.

Linguagem	<i>Modelo de Execução</i>	<i>Paradigma</i>	<i>Pervasividade</i>
Python	Interpretação	Orientada a Objetos	Baixa
Perl	Interpretação	Orientada a Objetos, Funcional e Procedural	Baixa
Ruby	Interpretação	Orientada a Objetos	Baixa
Flash Lite	-	-	Baixa
C++ (Symbian)	Compilação	Orientada a Objetos	Baixa
J2ME	Compilação e Interpretação	Orientada a Objetos	Alta

Tabela 1: Linguagens de programação e pervasividade.

3 O termo J2ME será usado em todo o texto pois atualmente há uma transição do nome para JME, entretanto a maioria das publicações ainda adota J2ME.

3.3- Ferramenta de Modelagem

Dependendo do cenário que se deseja modelar, uma Rede Bayesiana pode ser criada com dezenas ou até centenas de nodos. Sendo assim para facilitar a representação gráfica de tais estruturas existem ferramentas computacionais capazes de auxiliar no projeto e validação das Redes Bayesianas. A seguir tem-se a descrição da ferramenta utilizada para a criação da rede que representa o Interrogatório Sintomatológico. Neste trabalho foi adotada a abordagem de construção manual da rede, utilizou-se o *software Microsoft Belief Networks* (MSBNx) [9] para construção, sendo que a ferramenta é livre para uso não comercial.

3.3.1 - MSBNx

É um *software* para a plataforma *Windows* que dá suporte a criação, manipulação e avaliação de Redes Bayesianas. O *software* representa a rede graficamente através de um diagrama, onde as variáveis são exibidas como elipses, os nodos, e as dependências condicionais são exibidas como arcos entre as variáveis, Figura 3.3.

Cada nodo pode ser selecionado e então atribuído ao mesmo a sua tabela de probabilidades assim como definidas as evidências que podem ser observadas no cenário que se está modelando.

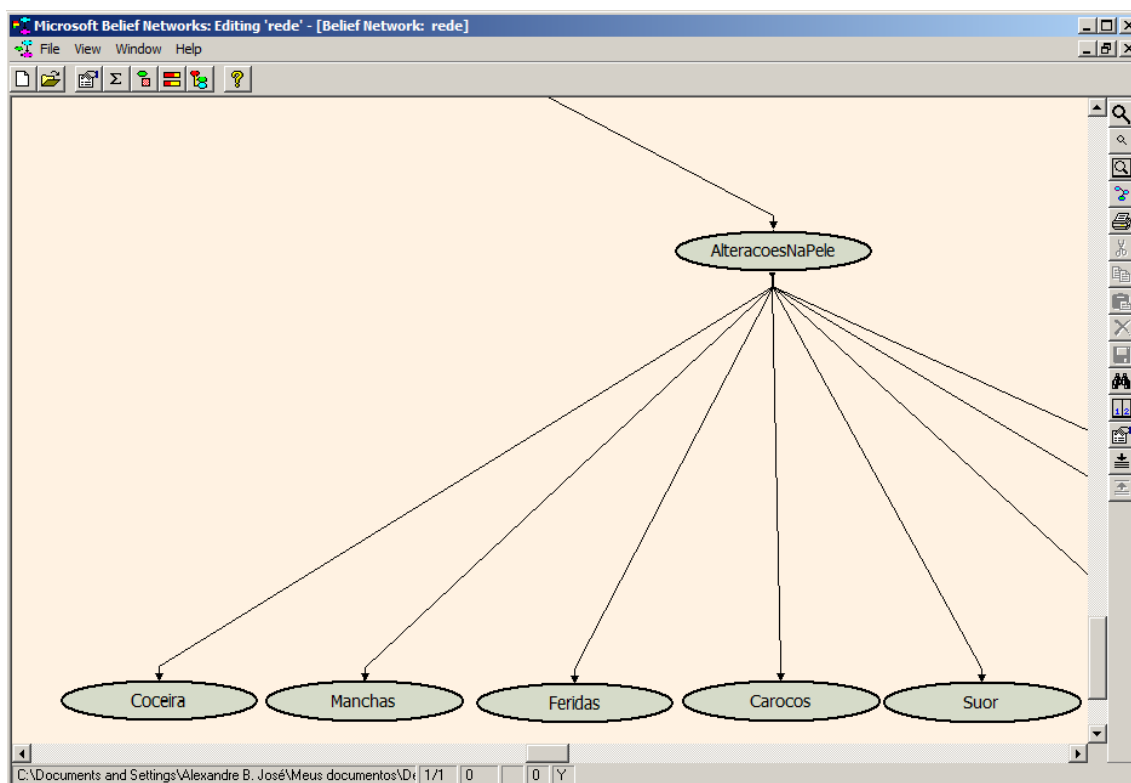


Figura 3.3. - Interface da ferramenta MSBNx.

O *Microsoft Belief Networks* permite que a estrutura gráfica gerada seja armazenada em um arquivo em disco com a opção de escolha do tipo de formato que será dado ao arquivo gerado. Para a execução deste trabalho foi escolhido o formato dsc, um formato de texto estruturado que pode ser visualizado em seus componentes principais (Figura 3.4 e Figura 3.5). A versão utilizada do *software* foi a 1.4.2.

Como principais vantagens do *Microsoft Belief Networks* temos a sua facilidade para trabalhar com vários nodos, permitindo a manipulação gráfica da maioria de suas funções, a possibilidade de se realizar inferências sobre novas evidências inseridas no modelo, pois o mesmo utiliza o algoritmo de propagação de evidências árvore de junção. Outra vantagem é que se trata de um *software* que pode ser utilizado livremente para fins não comerciais.

```

node AchouQueOCoracaoParouDeBater
{
    name = "Achou que o Coracao Parou de Bater";
    type = discrete[2]
    {
        "Yes",
        "No"
    };
    position = (570789, -73999);
}

```

Figura 3.4. - Representação do nodo no arquivo dsc.

```

probability(AchouQueOCoracaoParouDeBater|Peito)
{
    (0) = 0.5, 0.5;
    (1) = 0.5, 0.5;
}

```

Figura 3.5. - Representação da tabela de probabilidade do nodo.

Uma das desvantagens dessa ferramenta é que a mesma não gera o modelo em forma de estrutura de dados para nenhuma linguagem específica, usa um formato proprietário no arquivo com extensão dsc.

3.4- Modelando Interrogatórios

Interrogatórios que seguem um padrão hierárquico entre as questões podem ser modelados em uma Rede Bayesiana, onde as questões principais podem ser os nodos e as possíveis respostas serão as evidências que podem ser observadas. Nesta seção discute-se a criação de tais estruturas, usando-se como exemplo prático o Interrogatório Sintomatológico, apresentado na próxima seção.

Um interrogatório hierárquico é definido a partir de uma relação de precedência entre as questões, normalmente são criados usando-se estruturas de tópicos e sub-tópicos, tais estruturas podem ser representadas como grafos, ou seja, um grafo $G(V,E)$ é um conjunto finito não vazio V e um conjunto E de pares não-orientados de elementos distintos de V , onde:

- V é um conjunto de vértices
- E é um conjunto de arestas

Normalmente a representação geométrica de tais estruturas é mais utilizada, sendo descrita na Figura 3.6. A definição foi relaxada para permitir elementos não distintos de V no conjunto E.

$G(V,E)$

$G: V=\{1,2,3,4,5\}; E=\{(1,1),(2,1),(2,3),(3,4),(4,1)\}$

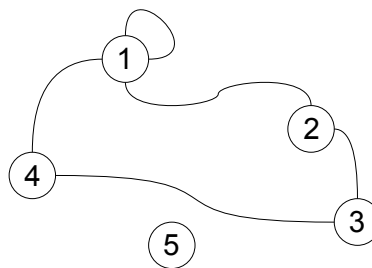


Figura 3.6. - Representação geométrica do grafo.

Um exemplo genérico de questionário hierárquico é dado e em seguida, na Figura 3.7 tem-se sua representação geométrica.

Questionário:

1. O que está sendo analisado?
 - 1.1. Item A.
 - 1.1.1. Item A possui?
 - 1.1.1.1. Característica X.
 - 1.1.1.2. Característica Y.
 - 1.2. Item B.
 - 1.3. Item C.
 - 1.4. O aspecto geral dos itens?
 - 1.4.1. Bom
 - 1.4.2. Regular

1.4.3. Ruim

No pequeno exemplo de questionário acima, pode-se observar sua estrutura de tópicos e sub-tópicos. E que quanto mais se desce na estrutura, mais específico se tornam as evidências.

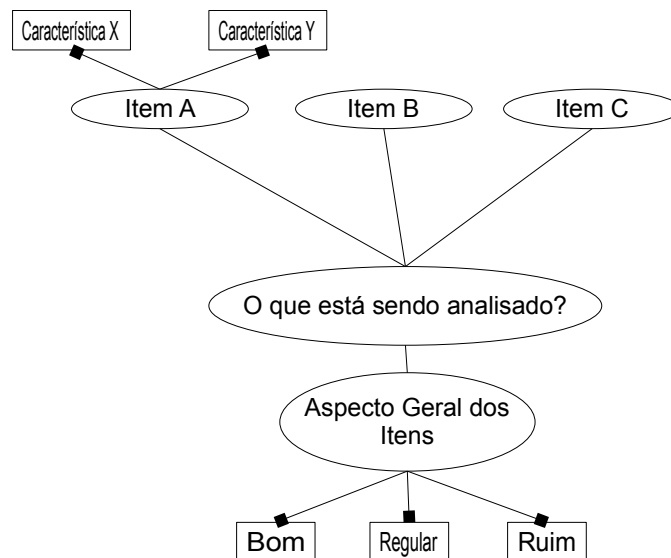


Figura 3.7. - Representação geométrica do grafo das questões.

Na Figura 3.7 há uma representação geométrica do grafo do exemplo acima. Vê-se que as perguntas com níveis superiores são modeladas em um nodo, e as perguntas mais internas são modeladas de maneira que possuam em sua composição seus sub-itens, indicados pela ligação terminada em quadrado. O exemplo acima foi propositalmente construído com um único nível superior, no caso nível 1. No entanto a maioria dos questionário possuem vários itens com um mesmo nível superior igual a 1. Acrescentando-se no exemplo acima mais um item no nível 1, teríamos mais um nodo.

Questionário:

1. O que está sendo analisado?

1.1. Item A.

1.1.1. Item A possui?

1.1.1.1. Característica X.

1.1.1.2. Característica Y.

- 1.2. Item B.
- 1.3. Item C.
- 1.4. O aspecto geral dos itens?
 - 1.4.1. Bom
 - 1.4.2. Regular
 - 1.4.3. Ruim

2. A análise foi feita usando?

- 2.1. Técnica A.
- 2.2. Técnica B.

Acima há o acréscimo de mais um item no primeiro nível, item em negrito. Na Figura 3.8 é apresentada o grafo com o novo elemento. Note que, como é um elemento de um nível superior ele possui seu próprio nodo que é totalmente independente dos outros, ou seja, não há um vértice que liga este nodo ao restante do grafo. Isso ocasiona dois problemas. O primeiro é que uma Rede Bayesiana deve ser definida através de um grafo onde nenhum nó esteja desconectado. Segundo mesmo se fosse possível criar uma rede com nodos desconectados, algoritmos clássicos de busca em grafos, como busca em profundidade e largura, não conseguiriam atingir todo o universo de possibilidades.

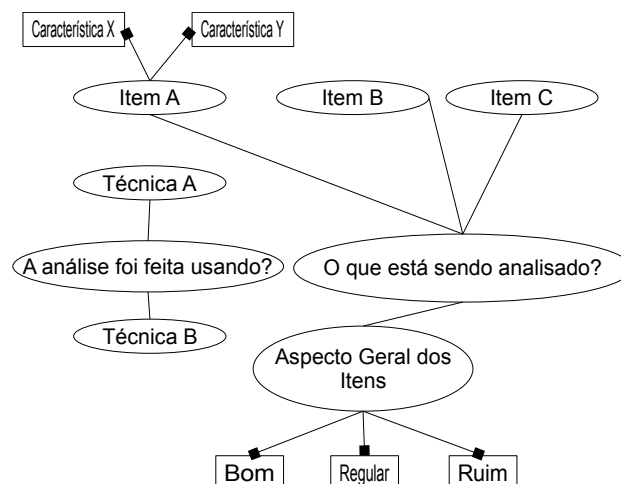


Figura 3.8. - Grafo com nova questão de nível 1.

A solução é obtida através do acréscimo de um “super nível”, no conjunto original de

questões, que será na hierarquia o pai de todos os outros níveis, ou seja, no grafo será o nó raiz. Agora com um nó raiz é possível pensar no uso dos algoritmos clássicos de busca em grafos, pois intuitivamente toda busca começa pela raiz. A Figura 3.9 mostra o novo grafo com o nó raiz, possibilitando sua montagem de forma mais hierárquica.

Questionário com “super nível”:

1. Início

1.1. O que está sendo analisado?

1.1.1. Item A.

1.1.1.1. Item A possui?

1.1.1.1.1. Característica X.

1.1.1.1.2. Característica Y.

1.1.2. Item B.

1.1.3. Item C.

1.1.4. O aspecto geral dos itens?

1.1.4.1. Bom

1.1.4.2. Regular

1.1.4.3. Ruim

1.2. A análise foi feita usando?

1.2.1. Técnica A.

1.2.2. Técnica B.

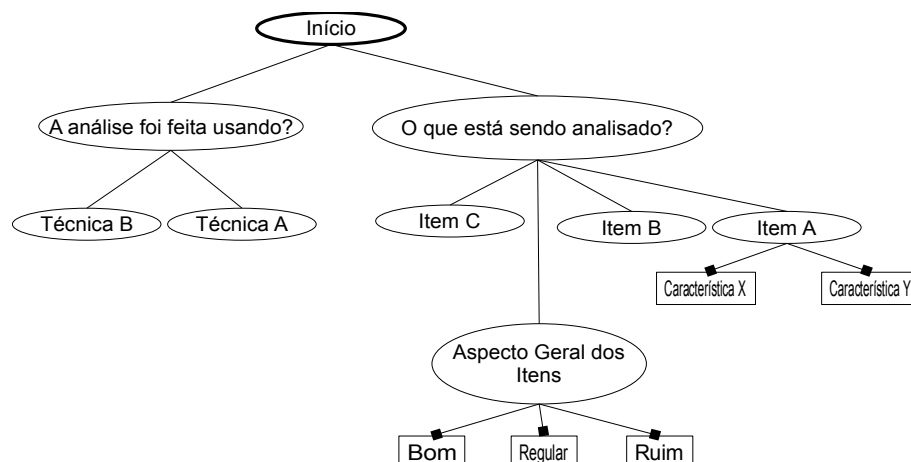


Figura 3.9. - Grafo do questionário com raiz.

4 - IMPLEMENTAÇÃO DO *FRAMEWORK*

Uma Rede Bayesiana como definida no item 4.1 é um grafo acíclico e dirigido, sendo então possível gerar algoritmos capazes de percorrer o grafo seguindo algum critério de busca. Durante o trajeto pelo grafo é possível realizar a coleta de evidências fornecidas pelo usuário do sistema, tais grupos de evidências podem fornecer a base para a geração de um raciocínio evidencial nos módulos de decisão.

4.1- Coleta de Evidências: Algoritmos em Grafos

Levando-se em conta que a rede assume a forma de um grafo acíclico, pode-se então pensar em usar algoritmos para caminhamentos em árvores. Para este trabalho foram avaliados dois algoritmos: um algoritmo de busca em profundidade e um outro que utiliza o caminhamento em largura [5, 1].

Definidos os algoritmos que serão analisados, basta agora definir qual será o parâmetro que definirá qual terá o “melhor” ou “pior” desempenho no caminhamento pelo grafo. Uma característica dos interrogatório hierárquicos é que os mesmos assumem uma visão de alto nível de possíveis características que podem ser observadas e então vão refinando tal visão através de perguntas cada vez mais específicas. Portanto, o parâmetro de desempenho será qual dos algoritmos consegue chegar até as perguntas mais específicas de forma mais rápida sendo que a constatação é a velocidade de convergência que será dada pela notação *big-Oh*, ou seja, qual a função limitante superior do algoritmo.

4.1.1 - Busca em Profundidade

A busca em profundidade consiste em visitar todos os nós de um ramo até atingir os nós terminais, repetindo o processo em todos os ramos [5, 1].

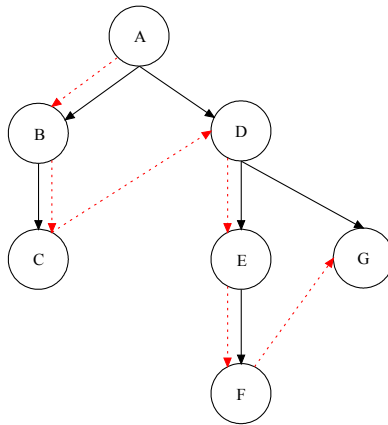


Figura 4.1. - Busca em profundidade.

4.1.2 - Busca em Largura

Na busca em largura todos os vértices adjacentes são examinados. A busca segue até que não reste mais nodos para serem examinados ou a chave de busca encontrada [5, 1].

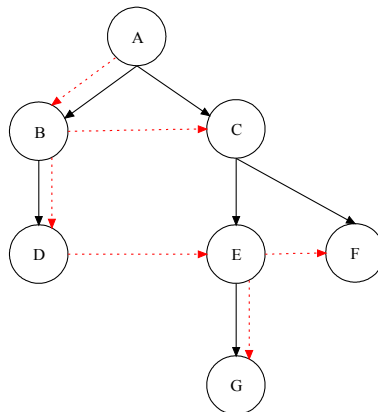


Figura 4.2. - Busca em largura.

4.1.3 - Algoritmo híbrido de caminhamento

Os algoritmos clássicos mostrados acima, não podem ser diretamente utilizados no grafo do interrogatório sintomatológico, pois não sabe-se qual é a chave de busca, ou conjunto de chaves. Tal informação só é obtida ao final do interrogatório. Por exemplo, faz-se necessário a execução de perguntas referentes aos nodos imediatamente posteriores a pergunta anterior. Ex: Caso a pergunta D da Figura 4.1 tenha sido feita, seria necessário realizar a pergunta referente ao nodo E e G, pois o algoritmo original não prevê qualquer tipo de heurística para escolher a pergunta mais apropriada.

O algoritmo híbrido utilizado neste trabalho, tem o objetivo de atingir um nodo folha, ou seja, um nodo terminal do grafo através da interação do usuário nas bifurcações do grafo. Assim quando um caminho com múltiplas escolhas é encontrado, o algoritmo depende de uma escolha do usuário para decidir por qual aresta irá seguir.

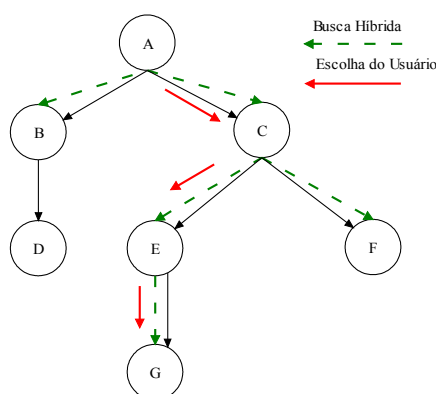


Figura 4.3. - Busca híbrida.

A complexidade computacional do algoritmo de busca híbrida usada neste trabalho, especificamente como mostrado na Figura 4.3 é $O(n)$, pois o número de iterações do algoritmo fica restrito ao número de nodos encontrados em seu trajeto, sendo que o grafo utilizado para o caminhamento não possui arestas cíclicas. No exemplo da Figura 4.3 os nodos visitados seriam: A, C, E, G, logo temos quatro iterações até o nodo terminal.

Caso o trajeto percorrido do grafo não contemple todas as possibilidades vivenciadas pelos usuários, ou seja, há ainda uma questão que não foi abordada pelo algoritmo, o mesmo provê a opção de retorno a raiz para uma nova rodada do algoritmo.

4.2- AS CLASSES DO *FRAMEWORK*

Os algoritmos responsáveis pelo funcionamento do *framework* foram implementados utilizando-se a linguagem Java. Os arquivos que representam as classes criadas são mostrados abaixo, seguidos de uma breve descrição:

1. `TradutorDSC.java` : Classe que implementa o tradutor.
2. `Nodo.java` : Classe que implementa o `Nodo`.
3. `GAD.java` : Classe que implementa o Grafo Acíclico e Dirigido.
4. `Bsearch.java`: Classe que implementa o algoritmo de híbrido de caminhamento.

4.2.1 - A Classe `TradutorDSC`

A classe que implementa o tradutor é responsável por portar a estrutura gerada pelo *Microsoft Belief Networks* para uma estrutura de dados dinâmica, permitindo assim sua manipulação pelos algoritmos.

O algoritmo supõe que há no mesmo diretório de sua execução um arquivo chamado `rede.dsc`, que possui a implementação da Rede Bayesiana modelada. Como principal característica técnica do `TradutorDSC` tem-se o uso da classe `My_StringTokenizer` que é responsável por separar as linhas lidas do arquivo `rede.dsc` em *tokens* que podem ser analisados, tornando possível então montar a estrutura de dados que representa o grafo em memória para ser manipulado pelos algoritmos.

Note no diagrama de classes reproduzido novamente na Figura 4.4, que cada vez que uma nova instância do `TradutorDSC` é criada, o mesmo se encarrega de criar um novo objeto da classe `GAD`. Assim o modelo dá liberdade de se criar em memória vários objetos `GAD`, sendo que cada um desses objetos pode representar uma Rede Bayesiana com características práticas diferentes.

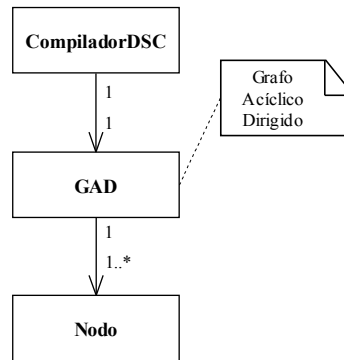


Figura 4.4. - Diagrama de classes.

A classe TradutorDSC possui um único método chamado `ListarArquivo()` que é responsável por iniciar toda a execução da leitura do arquivo `rede.dsc` e montagem das estruturas de dados. A listagem do código fonte do TradutorDSC pode ser vista no Apêndice A.

A classe `Nodo` é responsável por gerar todas as estruturas de dados e métodos para a criação e manutenção dos nodos em memória. Sua estrutura de dados pode ser vista na Figura 4.5.

4.2.2 - A classe nodo

O item de dado crucial para o funcionamento da estrutura mostrada acima é o campo `NodosFilhos` onde o mesmo armazena referências para os filhos do nodo em questão. Sendo que `NodosFilhos` é um campo do tipo *Vector*, uma estrutura de dados presente na linguagem *Java* que permite um crescimento dinâmico da estrutura.

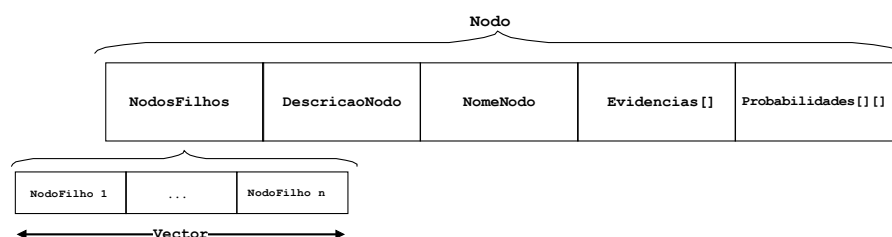


Figura 4.5. - Estruturas de dados da classe nodo.

Os métodos da classe nodo são descritos a seguir:

```
public Nodo(String Desc,String NomeN)
```

Método construtor do Nodo, recebe como parâmetros de construção duas *Strings* uma Desc (Descrição) e um NomeN (Nome do Nodo) que é a chave para pesquisa o grafo.

```
public ChoiceGroup imprimeEvidencias()
```

Método que percorre a lista de evidências do nodo em questão gerando um objeto do tipo ChoiceGroup contendo todas as evidências do nodo.

```
public int quantasEvidencias()
```

Método que retorna um inteiro com o número de evidência contidas no nodo.

```
public int quantosFilhos()
```

Método que retorna um inteiro com o número de filhos presentes no nodo.

```
public boolean temFilhos()
```

Método que retorna um *boolean* indicando se o nodo possui ou não filhos.

```
public void recebeFilho(Nodo ref)
```

Método que anexa um novo filho ao nodo em questão. Uma referência para o novo filho deve ser passada como parâmetro de entrada em *ref* que é do tipo Nodo.

```
public void removeFilho(Nodo ref)
```

Método para remover um filho do nodo em questão. Uma referência para o nodo que será removido deve ser passada com parâmetro de entrada em *ref* que é do tipo Nodo.

```
public int ondeEsta(Nodo ref)
```

Método que retorna um inteiro indicando a posição do nodo passado com referência no vetor de filhos.

```
public void fixarEvidencias(String evi[])
```

Método que atribui evidências ao nodo. Um vetor de *String* com as evidências é passado como parâmetro de entrada.

```
public void fixarProbabilidades(float p[][])
```

Método que atribui probabilidades ao nodo. Uma matriz de *Float* é passada com as probabilidades como parâmetro de entrada.

```
public ChoiceGroup listarFilhos()
```

Método que retorna um objeto do tipo *ChoiceGroup* contendo todos os filhos do nodo em questão assim como suas posições no vetor de nodos filhos.

A classe *Nodo* juntamente com seus métodos é uma classe basicamente utilizada pela classe *TradutorDSC*. Mas note que todos os métodos foram definidos como públicos, sendo assim em qualquer lugar do programa onde se tenha uma referência para um objeto do tipo *Nodo* os métodos podem ser invocados. Por exemplo: a qualquer referência de *Nodo* pode ser feita uma requisição para pedir a lista de seus filhos. Ex:

```
nodoInstanciado.listarFilhos();
```

Sendo que é retornado um objeto do tipo *ChoiceGroup* que deve ser armazenado em um objeto do mesmo tipo ou mandado diretamente para um método de exibição em tela que suporte a classe de objeto utilizada, no caso em *J2ME Screen*. A listagem de todo o código da classe *Nodo* pode ser vista no Apêndice B.

4.2.3 - A classe GAD

Responsável pela manutenção e criação de Grafos Acíclicos e Dirigidos na memória, é basicamente uma classe que assimila referencias para objetos do tipo *Nodo*, criando assim uma coleção de nodos que juntos formam um grafo.

Sua estrutura de dados pode ser observada na Figura 4.6. Onde vale destacar a capacidade de auto-ajuste do campo *Nodo* definido como um tipo *Vector*.

Como as classes *TradutorDSC*, *GAD* e *Nodo* são utilizadas de maneira que o

TradutorDSC gere instâncias das duas últimas classes respectivamente, observe a Figura 4.4, é possível realizar um agrupamento hierárquico de tais informações para seu uso nos algoritmos.

Ex:

```
TradutorDSC.Grafo.acharNo("Problema");
```

Onde tem-se a indicação do pacote TradutorDSC, seguido de uma indicação que se deseja acessar o membro Grafo, um objeto do tipo GAD criado dentro do próprio TradutorDSC. E em seguida é invocado o método acharNo do objeto Grafo, sendo que o método acharNo recebe como parâmetro uma *String*, no caso o “Problema” que é a raiz do grafo, de acordo com o que foi modelado no Interrogatório Sintomatológico e discutido na seção 4.4.

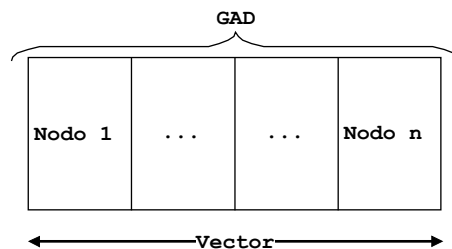


Figura 4.6. - Estruturas de dados da classe GAD.

Os métodos que dão funcionalidades a classe GAD são descritos a seguir:

```
public GAD()
```

Método construtor da classe GAD.

```
public void Insere(Nodo ref)
```

Método para inserir um novo nodo ao grafo.

```
public int ContaNos()
```

Método que retorna um inteiro com o número de nodos contidos no grafo.

```
public void imprimeGAD()
```

Método para criar uma listagem com todos os nodos contidos no grafo.

```
public Nodo acharNo(String chave)
```

Método para procurar por um determinado nodo no grafo, sendo o mesmo definido por uma *String* de busca, no caso do Interrogatório Sintomatológico usa-se o nome do nodo (NomeNodo) como chave de busca. A listagem de todo o código fonte da classe GAD pode ser vista no Apêndice C.

5 - O SISTEMA GOITACÁ

O sistema Goitacá desenvolvido neste trabalho tem como objetivo principal realizar a coleta de sintomas, ou seja, monitorar os usuários, através da execução do Interrogatório Sintomatológico provido pelo uso do *framework* FAGE.

5.1.1 - O Interrogatório Sintomatológico

A técnica de anamnese, ou seja, levantamento do histórico dos antecedentes de uma doença, pode ser feita de maneira automatizada. Celmo Celeno em seu livro sobre Semiologia Médica [7] diz o seguinte:

“A anamnese, que era considerada reduto inatingível pela tecnologia, já foi por ela alcançada. Estão sendo criados, em número crescente, aparelhos capazes de manipular os dados do ‘interrogatório sintomatológico’, anotados até pelo próprio paciente, inclusive sem a presença ou participação do médico.”

O interrogatório é uma ferramenta médica que auxilia o médico a determinar um quadro sobre seu paciente e assim gerar hipóteses sobre o estado de saúde do paciente. O interrogatório é formado por questões que abrangem desde a condição física quanto psíquica da pessoa.

O Interrogatório Sintomatológico é um conjunto de questões pertinentes à saúde do paciente, que tem como objetivo auxiliar o médico na realização da anamnese. As perguntas que compõem o interrogatório são organizadas de forma hierárquica, ou seja, permitem uma abordagem do tipo *top-down* do problema vivido pelo paciente. O mais importante neste tipo de hierarquia é que a mesma pode ser facilmente representada sob a forma de um grafo.

A seguir tem-se um pequeno trecho do Interrogatório Sintomatológico, ilustrando como o mesmo pode ser transformado em um grafo, sendo que o interrogatório completo pode ser visto no Apêndice D. O Interrogatório Sintomatológico utilizado foi obtido em [16].

3. Cabeça
- 3.1. Onde fica? no rosto (inclui a testa e o ouvido) resto da cabeça
- 3.1.1. Tem dor de cabeça? Sim Não
- 3.1.1.1. Começou: de repente gradualmente
- 3.1.1.2. Intensidade: fraca moderada intensa muito intensa
- 3.1.1.3. Dor: em queimação em pontadas pulsátil em aperto fisgada em peso contínua
- 3.1.1.4. Há quanto tempo? Menos de 12 horas entre 12 e 24 horas há mais de 1 dia
- 3.1.1.5. Onde dói? Toda cabeça na testa no lado esquerdo no lado direito na nuca
- 3.1.1.6. Irradiação para: testa parte de trás da cabeça nuca boca
- 3.1.1.7. Tem relação com: ansiedade menstruação leitura tosse exercício posição e/ou movimentos nenhuma alternativa anterior
- 3.1.1.8. Está associado com: distúrbios visuais ânsia de vômito vômito

Figura 5.1. - Parte do Interrogatório Sintomatológico

Na Figura 5.1 pode-se observar a estrutura de tópicos do interrogatório, evidenciando assim sua natureza hierárquica, vale lembrar que os itens em sublinhado são as evidências que podem ou não ser observadas pelo paciente. Assim o grafo só representa as perguntas, sendo que as evidências ficam inseridas dentro de cada nodo do grafo.

Abaixo se encontra a estrutura gráfica da Figura 5.1. Note no entanto que não há qualquer tipo de exibição das evidências nos nodos do grafo, tais evidências ficam armazenadas na estrutura de cada nodo (campo *State*), como pode ser visto na Figura 5.3 as evidências relativa a pergunta 3.1.1.1. Para simplificação da Figura 5.2 temos somente o item 3.1.1.1 e o item 3.1.1.8 como nodos folhas.

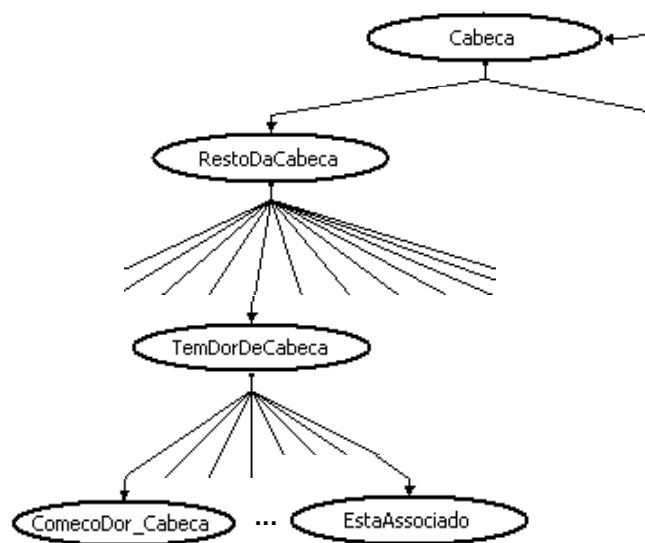


Figura 5.2. - Representação gráfica de parte do Interrogatório Sintomatológico.

Uma característica importante do interrogatório é que o mesmo só gera um antecessor para cada nodo, ou seja, cada nodo filho ou conjunto de nodos filhos no mesmo nível irá possuir apenas um nodo predecessor (pai). Garantindo assim um único trajeto possível até um nodo folha.

Node		Model	Property Types
Attribute	Value		
Name	ComecoDor_Cabeca		
Description	Comeco da Dor		
States	Name		
State(0)	Repentinamente		
State(1)	Gradualmente		
State(2)	<unassigned>		
Properties	Value		
MS_Asym	<unassigned>		

Figura 5.3. - Propriedades do Nodo e Exibição das Evidências.

O sistema foi concebido para se conectar com um sistema de gerenciamento de saúde implantado atualmente no HUB, o GSWeb. Assim é possível que o usuário possa

participar de maneira ativa na manutenção dos dados de seu próprio prontuário eletrônico. Item fundamental para permitir a troca de paradigma que visa tornar o indivíduo um elemento ativo no processo de manutenção e atualização dos dados de seu prontuário.

Além do objetivo descrito acima, o Goitacá também é capaz de executar a sincronização de sua base de dados interna com dados pertinentes no prontuário eletrônico do paciente, como por exemplo a lista de medicamentos. Podendo ser usado como auxiliar no uso de medicamentos com horários rígidos.

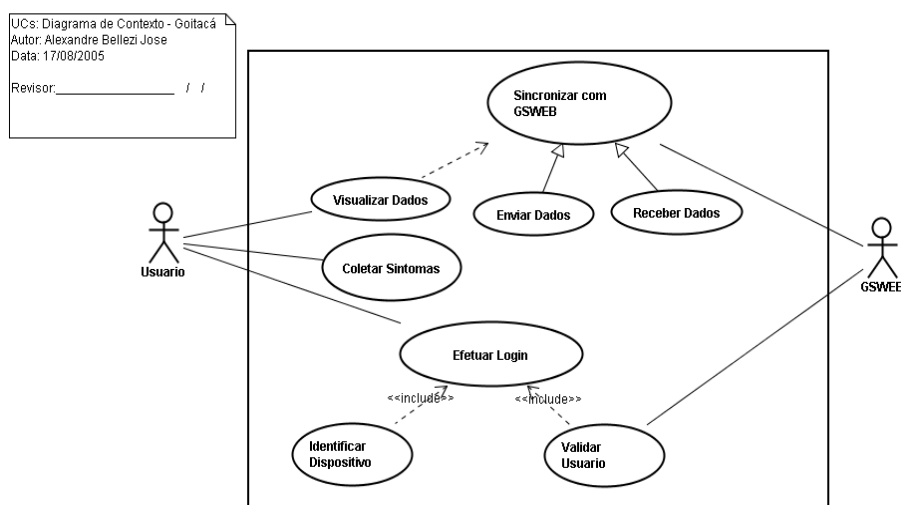


Figura 5.4. - Caso de Uso contexto geral.

O caso de uso da Figura 5.4 mostra de maneira geral o contexto de utilização do Goitacá. No primeiro acesso do usuário ao sistema Goitacá é realizado o *download* das informações do paciente para o telefone celular. O reconhecimento do usuário que está utilizando o sistema Goitacá é realizado através de um *login* e senha, previamente cadastrados no sistema de gerenciamento de informações de saúde, GSWeb, Figura 5.5. Ainda no primeiro acesso do usuário ao sistema Goitacá é iniciado o processo de criação de uma base dados local, ou seja, um armazenamento não volátil. Assim é possível guardar no dispositivo informações mesmo quando ele está desligado, melhorando também a capacidade de sincronização entre o sistema Goitacá e o GSWeb, pois nem sempre uma conexão de dados estará presente.

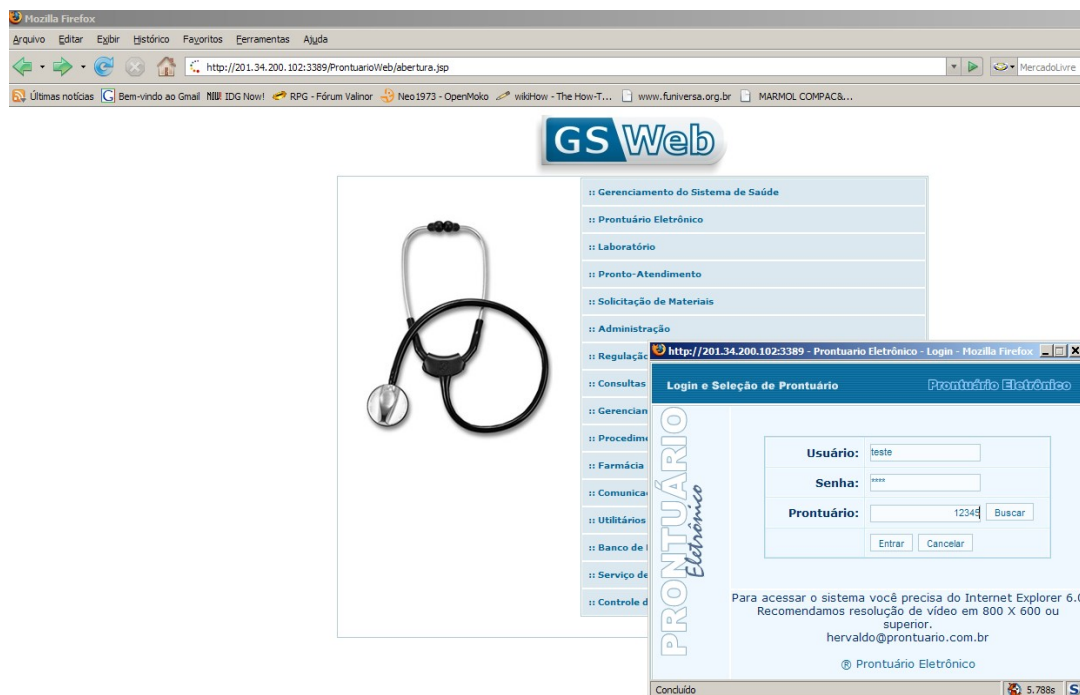


Figura 5.5. - Sistema GSWeb.

Nos usos subsequentes do Goitacá, o mesmo não mais irá pedir ao usuário um *login* de acesso, somente uma senha, pois como trata-se de um sistema pessoal de monitoramento não é interessante permitir em um mesmo telefone vários usuários. A Figura 5.6 mostra o caso de uso onde é contemplado a sincronização do sistema.

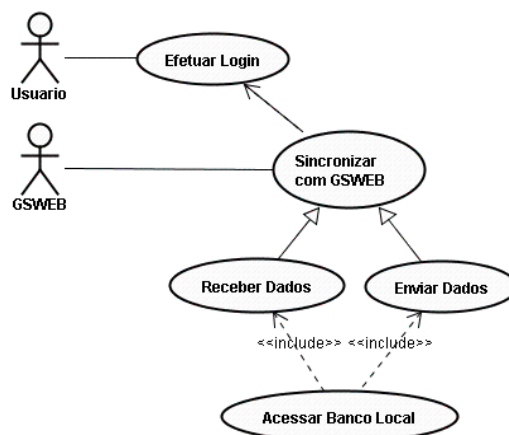


Figura 5.6. - Caso de Uso sincronização.

A documentação e modelagem do *software* Goitacá foi feita em UML, e pode ser vista no Apêndice E.

5.2- A COLETA DE SINTOMAS E SINAIS NO GOITACÁ

O sistema Goitacá realiza a coleta de sintomas utilizando o *framework* FAGE, para isso foi modelada uma Rede Bayesiana que representa o interrogatório sintomatológico, como descrito no item 6.1.1. A *interface* com o usuário faz uso de opções de seleção, que de acordo com o dispositivo em execução apresenta leves mudanças no *layout* da tela. A Figura 5.7 mostra o sistema de coleta de Goitacá executando o módulo de coleta de sintomas em um telefone celular e em um *Palm*. Observe que os menus de controle básicos, com por exemplo selecionar e voltar, ficam no idioma definido pelo sistema hospedeiro.

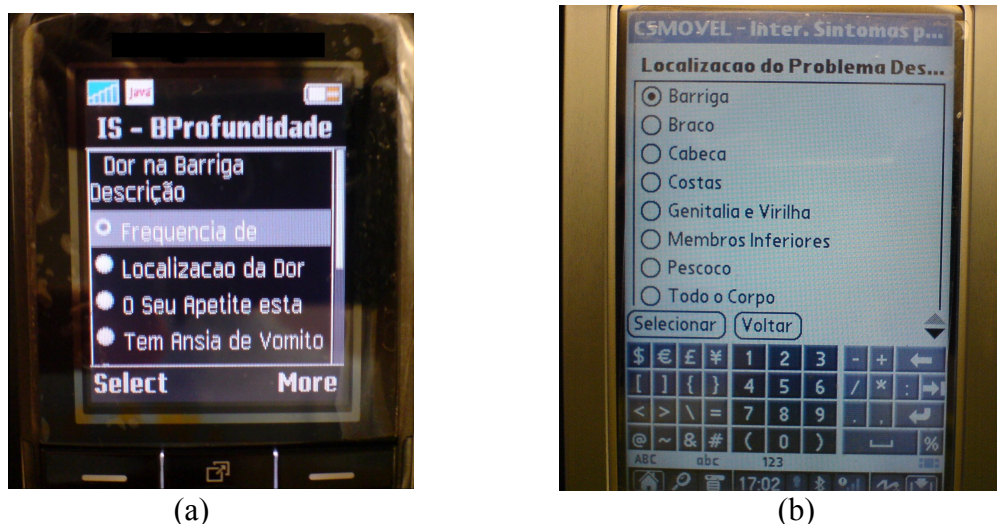


Figura 5.7. - Exemplo de Interface. (a) Telefone Celular (b) Palm

Após a interação do usuário com o conjunto de questões, o sistema solicita se as evidências coletadas devem ser armazenadas na base de dados local do dispositivo, assim mesmo que o dispositivo em questão seja desligado, as informações estarão salvas.

5.3- A SINCRONIZAÇÃO DE DADOS

De tempos em tempos o usuário do sistema Goitacá poderá sincronizar sua base de dados

local com uma base de dados remota do GSWeb. Assim os profissionais de saúde que têm acesso ao prontuário eletrônico do paciente, podem visualizar uma opção chamada diário do paciente, Figura 5.8.

No diário do paciente consta as informações que foram coletadas pelo *framework* FAGE utilizado no Goitacá.

Data e Hora	Descrição	Evidência
2007-09-04 11:33:12	FrequenciaDeEvacuacao	Normal
2007-09-04 11:43:35	FrequenciaDeEvacuacao	Normal
2007-09-04 08:42:58	PerdaDeMemoria	FatosRecentes
2007-09-04 08:48:39	TemFeridas_SexoFeminino	Tem Feridas
2007-09-04 08:48:39	DorDeGarganta	Dor de Garganta
2007-09-04 08:48:39	DeQueLado	LadoEsquerdo
2007-09-04 10:03:37	DeQueLado	Toda
2007-09-04 10:03:37	TemCoceira	Tem Coceira
2007-09-04 10:03:02	TemDificuldeParaMexer_Bracos	Tem Dificulde para Mexer
2007-09-04 11:21:00	InchacoNosMembroInferiores	Inchaco nos Membro Inferio
2007-09-04 11:22:23	OlhoEstaSeco	Olho esta Seco
2007-09-04 00:00:52	TemRegurgitacaoDeAlimentos	Tem Regurgitacao de Alime
2007-09-04 00:00:52	OSeuApetiteEstaNormal	Diminuido
2007-09-04 00:00:52	TemAnsiaDeVomito	Tem Ansia de Vomito
2007-09-04 00:00:52	TemPirose	Tem Pirose

Figura 5.8. - Tela do Diário de Sintomas.

Os dados transmitidos pelo sistema Goitacá são codificados em arquivos usando *eXtensible Markup Language* (XML), facilitando assim a interconexão do sistema Goitacá com outros sistemas de informação. A estrutura do arquivo XML que representa a base de dados local do dispositivo móvel pode ser vista na Figura 5.9. É valido ressaltar que o banco de dados local do dispositivo não é armazenado em formato XML, é usada uma representação binária, assim há uma economia do espaço necessário pela aplicação. A conversão para XML é feita no momento da transmissão das informações.

```
<?xml version="1.0" encoding="UTF-8"?>
<caminho>
  <nodo nome="Barriga" evidencia="Barriga">
  </nodo>
  <nodo nome="Dor_Barriga" evidencia="Dor na Barriga">
  </nodo>
  <nodo nome="TemAnsiaDeVomito" evidencia="Tem Ansia de
  Vomito">
  </nodo>
  <dataHora>2007-4-2 2:41:18</dataHora>
  <usuario dataHora="2007-4-2 2:41:18" id="1"></usuario>
</caminho>
```

Figura 5.9. - XML de Representação dos Sintomas e Sinais.

6 - CONCLUSÃO

Durante este trabalho viu-se como é possível transformar questionários estruturados de modo hierárquico em estruturas gráficas chamadas grafos, criados de modo acíclico e dirigido, permitindo assim que os mesmos possam ser transformados em Redes Bayesianas.

A principal contribuição deste trabalho foi a criação de um *framework* capaz de manipular Redes Bayesianas que representem um conjunto de questões definidas de maneira hierárquica, utilizando tecnologia J2ME o que aumenta sua pervasividade para uso em sistemas vestíveis. Como características e aplicações do *framework* desenvolvido tem-se:

1. Capacidade de reimplementação da classe de busca, ou seja, navegação pelas questões.
2. Rede Bayesiana independente do *framework* torna possível o uso de diferentes redes para diferentes contexto de utilização.
3. Integração com outros sistemas de informação facilitada pelo uso de arquivos XML para a representação do processamento do *framework*.

Na concepção deste trabalho ficou claro que a definição e construção de uma solução computacional dita genérica traz também alguns conceitos intrínsecos da orientação por objetos, afinal quando uma aplicação criada a partir de um *framework* é montada, percebe-se que a mesma deixou de ser genérica e se tornou específica para um determinado propósito, mesmo que o *framework* em qual foi baseada se mantenha genérico. Ou seja tem-se o mecanismo básico de especialização atuando. O *framework* dita apenas como deve ser feito e não o que deve ser feito.

Utilizando-se também o sistema Goitacá como mediador mostrou-se que é possível tornar usuários de prontuários eletrônicos agentes ativos na manutenção de seus próprios dados. Com a atual cobertura de redes de transmissão de dados via telefonia celular tipo GPRS é relativamente fácil enviar os dados via *Internet*. Um possível impedimento para uso de tais facilidades pode ser o alto custo de tais serviços. No entanto já nota-se uma diminuição nos preços, o que pode ser um fator instigante ao uso de tais sistemas on-line melhorando assim em muito a pervasividade de tais sistemas.

6.1- DIFICULDADES E LIMITAÇÕES DO PROJETO

Um dos problemas encontrados na elaboração deste projeto foi o fato de que a estrutura da Rede Bayesiana deve ser montada por um especialista para que seja atingida uma boa qualidade no funcionamento do *framework*. O problema está que na maioria das vezes um especialista não sabe necessariamente como é uma Rede Bayesiana ou como construir uma, mesmo utilizando softwares gráficos.

O *framework* proposto possui em sua composição capacidade de armazenar informações das tabelas de probabilidade de cada nodo do GAD. Entretanto tais informações não foram utilizadas pelo simples fato de não se conhecer as probabilidades que regem as dependências entre os nós do grafo. Tal fato gera um interessante desafio para um trabalho futuro descrito na próxima seção.

6.2- TRABALHOS FUTUROS

Em muitos cenários a princípio não se conhecem as probabilidades que regem as dependências entre os nós do grafo, assim fica inviável acrescentar tais informações nas Redes Bayesianas. Futuramente pode-se através de processamento da informações coletadas pelo *framework* que ficam armazenadas em banco de dados, no caso do sistema Goitacá, montar as tabelas de probabilidades através de análise dos dados e de suas frequências.

Assim poderia-se ter com o passar do tempo e conseqüente uso do sistema dados que podem gerar as informações probabilísticas. Ou seja, se durante um ano de uso sistema, 90% das pessoas que se queixaram de dor de cabeça e a classificaram como dor pulsátil, pode-se em uma nova versão da Rede Bayesiana já agregar esta informações probabilística e assim utilizar funções de heurística para melhorar a interação do sistema com o usuário, como por exemplo reordenado as opções de seleção dos menus de forma que as opções com maiores probabilidades fiquem nas primeiras posições. Ainda no exemplo anterior, a partir da seleção da opção de dor de cabeça, o sistema vestível já poderia antever uma

maior disponibilidade de recursos para sensores de pressão arterial ou mesmo trazer o nó sensor de um estado de dormência para atividade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Cormen, T.; Leiserson, C.; Rivest, R.; e Stein, C. (2002), Algoritmos - Teoria e Prática, (Tradução da 2ª Edição Americana.), 936pgs.
- [2] Disponível em <http://www.csee.ltu.se/~jench/mulle.html> acessado em 26/06/07.
- [3] Disponível em <http://www.intel.com/research/exploratory/motes.htm> acessado 26/06/07.
- [4] Disponível em <http://www.bluetooth.com/Bluetooth/Learn/Technology/lowpower.htm> acessado em 26/06/07.
- [5] Forbellone, André L. V.; Eberspächer, Henri F., Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados, Ed. Makron Books.
- [6] Disponível em <http://unbbayes.sourceforge.net/> acessado em 27/06/07.
- [7] Porto, Celmo C.(1997) Semiologia Médica. Editora Guanabara Koogan, Terceira Edição, 1187pgs.
- [8] Korhonen, I; Bardram, J. (2004), “Guest Editorial Introduction to the Especial Section on Pervasive Healthcare”, In: *IEEE Transactions on Information Technology in Biomedicine*, Vol. 8, N° 3, Setembro.
- [9] Disponível em <http://research.microsoft.com/adapt/MSBNx> acessado em 27/06/07.
- [10] Disponível em <http://www.uml.org/> acessado em 27/06/07.
- [11] Gilbas, C.; Jambeck, P. (2002). “Desenvolvendo a Bioinformática”, Editora Campus, Rio de Janeiro.
- [12] Devaul, R.; Sung, M.; Gips, J.; Pentland, A. “S.”. (2003) MIThril 2003: Applications and Architecture. Media Laboratory, Massachusetts Institute of Technology.
- [13] Lucas, P. (2004) “Bayesian networks in biomedicine and health-care.” Volume 30, Issue 3, 2001-214.
- [14] Carvalho, S. H. (2005). Data Fusion Implementation in Sensor Networks applied to Health Monitoring, Tese de Doutorado, Universidade Federal de Minas Gerais, UFMG, Brasil.
- [15] Sena, S. L. (2006). HANDMED – Um Sistema Móvel para Captura Automática de Sintomas e Monitoração da Saúde do Indivíduo. Dissertação de mestrado. UNB.

- [16]Barbosa, M. G. A. T.; Sene, Jr. G. I.; Castro, S. S. L.; Branisso, J. P. H.; Figueredo, C. É.; Carvalho, S. H.; Rocha, F. A.; Nascimento, A. O. F. (2004) Sistema Pessoal Móvel de Monitoração da Saúde: Algoritmo para Captura Inteligente de Sintomas. In: *IX Congresso Brasileiro de Informática na Saúde*.
- [17]Disponível em <http://www.python.org/> acessado em 03/07/07.
- [18]Disponível em <http://www.perl.org/> acessado em 03/07/07.
- [19]Disponível em <http://www.ruby-lang.org/en/> acessado em 03/07/07.
- [20]Disponível em <http://developer.symbian.com/main/tools/opensrc/ruby/part2.jsp> acessado em 03/07/07.
- [21]Disponível em <http://www.adobe.com/products/flashlite/> acessado em 03/07/07.
- [22]Disponível em http://wiki.forum.nokia.com/index.php/Diferen%C3%A7as_Symbian_C++_e_C/ acessado em 03/07/07.
- [23]Disponível em <http://developers.sun.com/mobility/device/device> acessado em 03/07/07.
- [24]Schildt, H. (2005). The Complete Reference Java J2SE 5 Edition. MacGraw-Hill/Osborne, 1000pgs.
- [25]Oliver, N.; Flores-Mangas, F. HealthGear: A Real-time Wearable System for Monitoring and Analyzing Physiological Signals. Microsoft Research, Redmond.
- [26]McDaniel, T. L.; Panchanathan, S. (2006). “A Visio-Haptic Wearable System for Assisting Individuals Who Are Blind”. In: *Acessibility and Computing*, Number 86.
- [27] Sena, S. L. (2005). “Automatização na Captura de Sintomas – Handmed” In: *XV Congresso Argentino de Bioengenharia*.
- [28]Disponível em <http://www.wearcomp.org/wearcompdef.html>
- [29]Araujo, R. B. (2003). “Computação Ubíqua: Principios, Tecnologias e Desafios.” In: *XXI Simpósio Brasileiro de Redes de Computadores*. Natal, CE.
- [30]Disponível em <http://www.futurehealth.rochester.edu/> acessado em 20/02/2008.
- [31]Devaul, R.; Schwartz, S. J.; Pentland, A. “S.”. (2001). MIThril: context-aware

computing for daily life. The Media Laboratory, Massachusetts Institute of Technology.

[32]Conway, J.C.D.; Fernandes, A.O.; Coelho, C.J.N. Jr.; Andrade, L.C.G.; da Silva, D.C., Jr.; Carvalho, H.S.(2000) “Wearable Computer as a Multi-parametric Monitor for Physiological Signals.” In: *Bio-Informatics and Biomedical Engineering*, 2000. Proceedings. IEEE International Symposium. 236-242.

[33]José, A. B.; Barbosa, T. M. G. A.; Sene JR, I. G.; Carvalho, H. S.; Rocha, A. F.; Castro, L. S. S. (2005). “Implementação de Revisão Sistemática de Sintomas em Sistemas Móveis Usando Redes Bayesianas.” In: *IV Simpósio Brasileiro de Qualidade de Software - V Workshop de Informática Médica*, Porto Alegre. Anais do IV Simpósio Brasileiro de Qualidade de Software, 2005. v. 1. p. 1-9.

[34]José, A. B.; Sene JR, I. G.; Barbosa, T. M. G. A.; Castro, L. S. S.; Carvalho, H. S.; Rocha, A. F.; Nascimento, F. A. O.; Carvalho, J.L.A. (2007). “A Framework for Automated Evidence Gathering with Mobile Systems Using Bayesian Networks.” In: *Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC)*, Lyon. Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), 2007. v. 29.

[35]SENE JR, I. G.; BARBOSA, T. M. G. A.; CARVALHO, H. S.; ROCHA, A. F.; NASCIMENTO, F. A. O. (2006). “Monitoração da temperatura corporal baseada em uma rede de sensores sem fios.” In: *CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE (CBIS)*. Florianópolis.

[36]Carvalho, S. H.; Heinzelmann, B. W.; Murphy, A. L.; Coelho, C. J. N. (2003). “A General Data Fusion Architecture.” In: *IEEE Information Fusion 2003, Cairns. Proceedings Of the 6th International Conference of Information Fusion*, v. 2. p. 1465-1472.

[37]Wald, L. (1999) “Definitions and Terms of Reference in Data Fusion.” In: *IAPRS*, Vol. 32, Part 7-4-3 W6, Valladolid, Spain, pp. 2-6.

[38]Disponível em <http://www.glucowatch.com/> acessado em 20/11/2006

[39]Mann, S.; Fung, J.; Aimone, C.; Sehgal, A.; Chen, D. (2005). “Designing EyeTap Digital Eyeglasses for Continuous Lifelong Capture and Sharing of Personal Experiences” In: *Proceedings of ALT.CHI 2005*. Portland, ACM Press.

[40]Carvalho, H. S.; Rocha, A. F.; Nascimento, F. A. O.; Ortis, R. S.; Coelho JR, C. J. N.

(2004). “Monitorização de Sinais Biomédicos em Assistentes Pessoais Digitais (PDA based Biomedical signals monitoring.)” In: *SBIS 2004*, Ribeirão Preto-SP. Anais SBIS 2004, 2004.

[41]Disponível em <http://www.ece.uah.edu/~jovanov/whrms/> acessado em 12/08/2007.

[42]Carvalho, H. S.; Coelho JR, C. J. N.; Heinzelmann, B. W. (2002). “Gerenciamento de Informações Médicas do Paciente(GIMPA)”, In: CBIS.

[43]Barbosa, T. M. G. A. ; Sene JR, I. G. ; Carvalho, H. S. ; Rocha, A. F. ; Nascimento, F. A. O. (2006). “Programação de uma rede de sensores para o corpo humano por meio de uma interface gráfica.” In: *CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE (CBIS)*, Florianópolis. CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE (CBIS), 2006.

[44]Barbosa, T. M. G. A. ; Sene JR, I. G. ; Carvalho, H. S. ; Rocha, A. F. ; Nascimento, F. A. O. (2007). “A New Model for Programming Software in Body Sensor Networks.” In: *Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC)*, Lyon. Annual International Conference IEEE Engineering in Medicine and Biology Society (EMBC), v. 29.

[45]Disponível em <http://sourceforge.net/projects/fage/> acessado em 15/10/2005.

[46]Disponível em <http://sistemasdeproducao.cnptia.embrapa.br/FontesHTML/Pessegueiro/CultivadoPessegueiro/cap04.htm> acessado em 27/11/2007.

APÊNDICE A

```

package fage;

import java.util.Vector;

import utilidades.My_File;
import utilidades.My_StringTokenizer;

/**
 * @author Alexandre Bellezi José
 */
public class TradutorDSC {

    // private InputStream NomeArquivo;
    private My_File entrada;

    // private InputStreamReader entrada;
    private String texto = new String();

    private My_StringTokenizer tokens;

    private String tk = new String();

    // Nodos, evidencias e probabilidades sao variaveis de depuracao
    private int nodos;

    // Nodos, evidencias e probabilidades sao variaveis de depuracao
    private int evidencias;

    // Nodos, evidencias e probabilidades sao variaveis de depuracao
    private int probabilidades;

    // Nodos, evidencias e probabilidades sao variaveis de depuracao
    private int aux;

    private boolean PontoEVirgula = false;

    private String Pai;

    private String Filho;

    private String Raiz = new String();

    private Vector evi = new Vector();

    private String e[];

    private float p[];

    public Nodo raiz;

    public Nodo pai = null;

```

```

public Nodo filho;

public/* static */GAD Grafo = new GAD();

//public Thread t = null;

/**
 * @param caminho
 *         representa o arquivo que será traduzido pelo TradutorDSC.
 */

public TradutorDSC(String caminho) {
    super();
    this.entrada = new My_File(caminho);

    //t = new Thread(this);
    //t.start();
}

public void iniciar() {
    while ((texto = entrada.my_readLine()) != null) {
        // System.out.println(texto);

        tokens = new My_StringTokenizer(texto, ' ');
        // IMPORTANTE: A organizacao do arquivo .dsc garante q os nomes dos
        // nodos
        // serao os primeiros a serem processados.
        while (tokens.my_hasMoreTokens()) {
            tk = tokens.my_nextToken();
            // Para descoberta de variaveis e suas evidencias
            if (tk.equals("node")) {
                // System.out.println("ACHEI UM NODO");

                // Tem-se o nome do nodo em tk
                tk = tokens.my_nextToken();

                // System.out.println("Token: " + tk);
                // Pula uma linha inutil
                texto = entrada.my_readLine();
                // Para na linha com a descricao do nodo

                texto = entrada.my_readLine();
                // Tira as aspas (34) da string
                String str = texto.substring(texto.indexOf(34) + 1, texto
                    .indexOf(34, texto.indexOf(34) + 1));

                Nodo no = new Nodo(str, tk, pai);
                Grafo.insere(no);
                nodos++;

                // Pula as linha 'inuteis'

                texto = entrada.my_readLine();
            }
        }
    }
}

```

```

texto = entrada.my_readLine();
// Para na linha da primeira evidencia:
texto = entrada.my_readLine();

// Problemas no my_readLine() obriga a dobrar sua quantidade
// texto = entrada.my_readLine();
// texto = entrada.my_readLine();
// texto = entrada.my_readLine();
// texto = entrada.my_readLine();
// texto = entrada.my_readLine();

// System.out.println("Linha da Evidência: " + texto);

// Pega todas a evidencias relacionadas a um nodo
while (texto.endsWith(";") != true) {
    texto = texto.trim();

    // Acha a Aspas e retira da string

    texto = texto.substring(texto.indexOf(34) + 1, texto
        .indexOf(34, texto.indexOf(34) + 1));

    // System.out.println("Evidência limpa: " +
    // texto.trim());

    evi.addElement(texto.trim());
    texto = entrada.my_readLine();

    // Problemas no my_readLine() obriga a dobrar sua
    // quantidade
    // texto = entrada.my_readLine();

    evidencias++;
}

e = new String[evidencias]; // Cria um novo array com o
// tamanho certo

// evi.toArray(e); Transforma o Vector em array
// pois não tem toArray no J2ME assim uso o for

// System.out.println("Vetor e[0]: " + e[0]);

for (int i = 0; i < evi.size(); i++) {
    // System.out.println("Evidência no X Vector: " +
    // evi.elementAt(i));
    e[i] = evi.elementAt(i).toString();
    // System.out.println("Evidência no Vetor e[" + i + "]
    // :"+ e[i]);
}

// Aqui as evidencias entram no nodo
Grafo.acharNo(tk).fixarEvidencias(e);
// Zerar o vetor de evidencias para receber novas evidencias
evi.removeAllElements();
evidencias = 0; // Zera o contador de evidencias

```

```

} else {
/*
 * LEMBRAR:Pensar em como colocar as probabilidades no Nodo
 * lembrando-se q precisamos achar o nodo correto no Grafo
 * Montar um Matriz com as probabilidades e passar para o
 * Nodo A matriz sera Nx2 sendo q N = numero de evidencias
 * no vetor de evidencias! Obs: Nx2 vale para o dominio do
 * interrogatorio pois so ha um pai por filho
 */

// Para descoberta das probabilidades das variaveis
aux = tk.indexOf("probability");// -1 se nao achou
if (aux != -1) // Achou
{
/*
 * FUNCIONA POIS POR ENQUANTO SO TEMOS UM PAI POR FILHO
 * QUE E UMA CARACTERISTICA DA REDE DO INTERROGATORIO
 */
// Para achar o(s) Filho(s)
// aux agora tem o inicio do Pai
aux = tk.indexOf("|");

// Se tiver pai
if (aux != -1) {
Filho = tk.substring(12, aux);
// Para achar o Pai
// Sempre e o ultimo item na 2-upla
// (x|pai)

Pai = tk.substring(aux + 1, tk.indexOf(" "));
pai = Grafo.acharNo(Pai);
Grafo.acharNo(Filho).setPai(pai);

// Acha o Pai no vetor de Nodos e coloca seu filho
Grafo.acharNo(Pai)
.recebeFilho(Grafo.acharNo(Filho));
// System.out.println("Filho do Nodo " + Pai + " é "
// + Filho);
} else { // E a raiz e nao tem pai!

Raiz = tk.substring(12, tk.indexOf(" "));
Pai = null;
pai = null;
Grafo.acharNo(Raiz).setPai(null);
// Temos a raiz do Grafo!
}

//System.out.println("Pai String: " + Pai);
//System.out.println("Pai Nodo: " + pai.NomeNodo);

// LEMBRAR!: Nesse ponto ja posso localizar o nodo no
// Grafo (Vetor)
// probability(xx|yy) xx representa o nome do nodo
// (SearchKey)
// xx e o Filho

```



```

// Pula as linha 'inuteis'
texto = entrada.my_readLine();
texto = entrada.my_readLine();

// Problemas no my_readLine() obriga a dobrar sua
// quantidade
// texto = entrada.my_readLine();
// texto = entrada.my_readLine();

// IMPORTANTE: assume-se que as probabilidades não
// passam de um linha no texto
// SENA0: gera erro no token
if (texto.startsWith("\tdefault")) {
texto = entrada.my_readLine(); // O default e
// desnecessario
// Agora estamos na primeira linha de probabilidades

// Problemas no my_readLine() obriga a dobrar sua
// quantidade
// texto = entrada.my_readLine();
}

while (texto.endsWith("}") != true) {
tokens = new My_StringTokenizer(texto, ' ');
while (tokens.hasMoreTokens()) {
// Como e sem pai nao preciso pular as linhas
// pois elas nao existem mesmo
if (Pai != null) {
tokens.my_nextToken(); // (x) são
// descartados
tokens.my_nextToken(); // = são descartados
}
// Aqui comecam os valores
tk = tokens.my_nextToken();
while (PontoEVirgula == false) {
if (tk.endsWith(";") == false) {
tk = tk.substring(0, tk.indexOf(","));
// LEMBRAR: Fazer o set no nodo do Grafo
PontoEVirgula = false;
tk = tokens.my_nextToken(); // Ainda tem
// tokens
} else {
PontoEVirgula = true;
tk = tk.substring(0, tk.indexOf(";"));
// LEMBRAR: Fazer o set no nodo do Grafo
}
}
PontoEVirgula = false;
}
// Alimenta nova linha
texto = entrada.my_readLine();

// Problemas no my_readLine() obriga a dobrar sua
// quantidade
// texto = entrada.my_readLine();
}

```

```
}  
probabilidades++;  
}  
}  
} // Fim do While tiver texto no buffer  
// System.out.println("Achei " + nodos + " Variaveis");  
// return (true);  
} // Fim do Public Listar  
}
```

APÊNDICE B

```

package fage;

import java.util.Vector;

import javax.microedition.lcdui.ChoiceGroup;
/**
 *
 * @author Alexandre Bellezi José
 * <p>
 * Classe responsável pela manutenção de nós que podem compor um novo
 * grafo.
 */
public class Nodo
{
    /**
     * Vetor de Ponteiros para os Filhos do Nodo.
     */
    public Vector NodosFilhos;
    /**
     * Descrição textual do Nodo.
     */
    public String DescricaoNodo;
    /**
     * Nome para identificação do Nodo,
     * funciona como chave de busca.
     */
    public String NomeNodo;

    /**
     * Vetor de evidências de um nodo.
     */
    public String Evidencias[];
    /**
     * Matriz de probabilidades de um nodo.
     */
    public float Probabilidades[][];
    private Nodo pai = null;

    /**
     * @param Desc Recebe como parâmetro uma String representando
     * a descrição do nó.
     * @param NomeN Recebe como parâmetro uma String representando
     * o nome do nó, onde o nome é a chave de busca pelo grafo.
     */
    public Nodo(String Desc,String NomeN, Nodo pai)
    {
        this.DescricaoNodo = new String(Desc);
        this.NomeNodo = new String(NomeN);
        this.NodosFilhos = new Vector(1);
        this.pai = pai;
    }
    /**

```

```

    * Método para gerar uma listagem de evidência. Tal listagem
    * já retorna configurado um objeto do tipo ChoiceGroup. Facilitando
    * assim o reaproveitamento de código. Os itens do ChoiceGroup podem
    * ser usados como (seus índices) ponteiros diretos para novos nodos do
    * grafo.
    * @return Um objeto ChoiceGroupo contendo todas as evidências de um
    nó.
    */
    public ChoiceGroup imprimeEvidencias()
    {

        ChoiceGroup vetorEvidencias = new ChoiceGroup(" " + this.DescricaoNodo
        +
        "\nDescrição",ChoiceGroup.EXCLUSIVE);

        for(int aux = 0; aux < this.Evidencias.length; aux++)
        {
            //System.out.println("Evidencia S/N " + this.Evidencias[aux]);
            vetorEvidencias.append(ConsertaTexto.consertaEspaco(this.Evidencias[a
            ux]), null);//Sem imagem
        }

        //for(int aux = 0; aux < this.Evidencias.length; aux++)
        //vetorEvidencias.setFont(aux,Font.getFont(Font.FACE_PROPORTIONAL,Font.ST
        YLE_PLAIN,Font.SIZE_LARGE));

        return(vetorEvidencias);
    }

    /**
    * Método usado para saber a quantidade de evidência em um nó.
    * @return Número de evidências contidas no nodo.
    */
    public int quantasEvidencias()
    {
        return(this.Evidencias.length);
    }

    /**
    * Método usado para saber a quantidade de filhos de um nó.
    * @return Número de filhos anexados ao nodo.
    */

    public int quantosFilhos()
    {
        return(this.NodosFilhos.size());
    }

    /**
    * Método usado para saber se o nó possui filhos.
    * @return true se o nó tiver filhos
    * false caso contrário.
    */

    public boolean temFilhos()
    {

```

```

return(!(this.NodosFilhos.isEmpty()));
}

/**
 * Método responsável por anexar um novo filho ao nodo.
 * @param ref Uma referência a um objeto do tipo Nodo, ou seja,
 * o novo filho do nó.
 */
public void recebeFilho(Nodo ref) //Referencia do nodo filho
{
this.NodosFilhos.addElement(ref);
}

/**
 * Método responsável por remover um filho do nodo.
 * @param ref Uma referência a um objeto do tipo Nodo, ou seja,
 * o nó que será removido do vetor de filhos (NodosFilhos).
 */
public void removeFilho(Nodo ref)
{
this.NodosFilhos.removeElement(ref);
}

/**
 * Identifica a posição do nó no vetor de nodos filhos.
 * @param ref O nó que se deseja saber a posição no vetor de nodos
 * filhos (NodosFilhos).
 * @return Um inteiro representado o índice do objeto no vetor.
 */
public int ondeEsta(Nodo ref) //Referencia do nodo filho
{
return(this.NodosFilhos.indexOf(ref));
}

/**
 * Grava (seta) evidências a um nó.
 * @param evi Vetor de String que representa as evidência de um
 * determinado nó.
 */
public void fixarEvidencias(String evi[])
{
this.Evidencias = evi;
}

/**
 * Grava (seta) as probabilidade a um determinado nó.
 * @param p Uma matriz de float que representando as
 * probabilidades do nodo.
 */
public void fixarProbabilidades(float p[][])
{
this.Probabilidades = p;
}

/**
 * Lista todos os filhos do nodo. A listagem é retornanda

```

```

* dentro de um objeto do tipo ChoiceGroup, assim pode ser usada
* de maneira direta na interface com o usuário. Idem ao método
* imprimeEvidencia().
* @return Objeto ChoiceGroup contendo a listagem de todos
* os filhos do nó.
*/
public ChoiceGroup listarFilhos()
{
    Nodo x;

    ChoiceGroup filhosNodo = new ChoiceGroup(" " + this.DescricaoNodo +
"\nDescrição",ChoiceGroup.EXCLUSIVE);

    for(int aux = 0; aux < this.NodosFilhos.size(); aux++)
    {
        x = (Nodo) this.NodosFilhos.elementAt(aux);
        filhosNodo.append(x.DescricaoNodo, null);
        //System.out.println("Filhos do Nodo " + this.NomeNodo + "é filho: "
        //+ x.NomeNodo);
    }
    //for(int aux = 0; aux < this.NodosFilhos.size(); aux++)
    //filhosNodo.setFont(aux,Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_P
    LAIN,Font.SIZE_LARGE));

    return(filhosNodo);
}

public Nodo getPai(){
    return this.pai;
}
public void setPai(Nodo pai){
    this.pai = pai;
}
}

```

APÊNDICE C


```

package fage;

import java.util.Vector;

/**
 * @author usuario
 */
public class GAD //Grafo Aciclico Dirigido
{
    /**
     * Vetor de Nodos para armazenar o grafo.
     */
    private Vector Nodos;

    /**
     * Atributo auxiliar para funcionamento interno da classe GAD
     */
    private Nodo x;

    /**
     * Construtor da classe GAD não necessita
     * de parâmetro.
     */
    public GAD()
    {
        this.Nodos = new Vector(1);
    }

    /**
     * @param ref recebe um objeto do tipo Nodo para inserir no grafo
     */
    public void insere(Nodo ref)
    {
        this.Nodos.addElement(ref);
    }

    /**
     * @return Retorna o número de nós contidos no grafo.
     */
    public int contaNos()
    {
        return(this.Nodos.size());
    }

    /**
     * Método para depuração do sistema. Somente imprime uma listagem
     * do conteúdo do grafo no console.
     */
    public void imprimeGAD()
    {
        for (int aux = 0; aux < this.Nodos.size();aux++)
        {
            //IMPORTANTE: Forcar o tipo (Nodo) +- 4 Horas desp. p. descobrir isso!!!
            x = (Nodo) this.Nodos.elementAt(aux);

```

```

System.out.println(x.NomeNode);
}
}
/**
 * @param chave Objeto do tipo String que representa a chave de busca
 * pelo grafo.
 * @return Retorna um objeto do tipo Node se a chave de busca
 * for localizada, caso contrario retorna null.
 */
public Node acharNo(String chave)
{
    int aux = 0;

    do
    {
        x = (Node) this.Nodos.elementAt(aux);
        aux++;
    } while (aux < this.Nodos.size() && (x.NomeNode.equals(chave) != true));

    if (this.x.NomeNode.equals(chave) == false)
        return null;
    else
        return x;
    }
}

```

APÊNDICE D

1. Você está sentindo alguma coisa diferente?
ou tem algo que te incomoda?
ou tem algum problema? Sim Não
2. Qual a localização?
3. Todo o corpo
4. Cabeça
5. Pescoço
6. Tronco
7. Braço
8. Pernas
3. Todo o corpo
 - 3.1. Tem febre? Sim Não
 - 3.1.1. Qual a temperatura? 37oC 37,5oC 38oC 38,5oC 39oC 39,5oC 40oC 40,5oC 41oC
 - 3.1.2. Começou há: 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
 - 3.1.3. Qual o horário em que aparece mais? O dia inteiro manhã tarde noite madrugada
 - 3.2. Tem dor? Sim Não
 - 3.2.1. Começou: de repente gradualmente
 - 3.2.2. Intensidade: fraca moderada intensa muito intensa
 - 3.2.3. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
 - 3.2.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
 - 3.2.5. Irradiação
 - 3.2.6. Começou com esforço físico ? Sim Não
 - 3.2.6.1. pequenos esforços médios esforços grandes esforços
 - 3.3. Tem fraqueza? Sim Não
 - 3.3.1. Sentiu tontura? Sim Não
 - 3.3.1.1. Teve necessidade de se apoiar? Sim Não
 - 3.3.1.2. Caiu sem perder a consciência? Sim Não
 - 3.3.1.3. Chegou a perder a consciência? Sim Não
 - 3.4. Tem tremores? Sim Não
 - 3.4.1. São: leves fortes muito forte
 - 3.5. Tem alterações da pele? Sim Não
 - 3.5.1. Tem coceira? Sim Não

- 3.5.2. Tem manchas? Sim Não
- 3.5.2.1. São: Vermelhas azuis brancas amarelas verdes escuras
- 3.5.3. Tem feridas? Sim Não
- 3.5.3.1. Sangram? Sim Não
- 3.5.3.2. Tem crostas? Sim Não
- 3.5.3.3. Tem pus? Sim Não
- 3.5.3.4. Tem descamação? Sim Não
- 3.5.4. Tem caroços? Sim Não
- 3.5.5. Sua muito (anormal)? Sim Não
- 3.5.6. Tem alterações nos pelos? Sim Não
- 3.5.6.1. Tem: queda de pelos excesso de pelos
- 3.5.7. Tem alterações nas unhas? Sim Não
- 3.5.8. Tem veias visíveis? Sim Não
- 3.5.8.1. deitado? Em pé?
- 4. Cabeça
- 4.1. Onde fica? na face (inclui o ouvido) .2 resto da cabeça .1
- 4.1.1. Tem dor de cabeça? Sim Não
- 4.1.1.1. Começou: de repente gradualmente
- 4.1.1.2. Intensidade: fraca moderada intensa muito intensa
- 4.1.1.3. Dor: em queimação em pontadas pulsátil em aperto fisgada em peso contínua
- 4.1.1.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
- 4.1.1.5. Onde dói? Toda cabeça na testa no lado esquerdo no lado direito na nuca
- 4.1.1.6. Irradiação
- 4.1.1.7. Tem relação com: ansiedade menstruação leitura tosse exercício posição e/ou movimentos
- 4.1.1.8. Está associado com: distúrbios visuais ânsia de vômito vômito
- 4.1.2. Tem tontura? Sim Não
- 4.1.3. Tem vertigens? Sim Não
- 4.1.4. Bateu a cabeça? Sim Não
- 4.1.5. Tem perda de memória? Sim Não
- 4.1.5.1. de fatos antigos de fatos recentes
- 4.1.6. Tem coceira? Sim Não
- 4.1.7. Tem insônia? Sim Não
- 4.1.8. Tem sonolência? Sim Não
- 4.1.9. Tem alguma ferida na cabeça? Sim Não
- 4.1.9.1. Sai sangue? Sim Não

- 4.1.9.2. Sai pus? Sim Não
- 4.1.9.3. Há perda de cabelo no local? Sim Não
- 4.1.9.4. Tem descamação? Sim Não
- 4.1.10. Há perda excessiva de cabelo? Sim Não
- 4.1.11. Tem algum caroço? Sim Não
- 4.1.11.1. é doloroso? Sim Não
- 4.1.12. Você compreende bem o sentido das palavras? Sim Não
- 4.2. Na face (inclui o ouvido):
- 4.2.1. Fica: na pele (.1) nos olhos (.2) nos ouvidos (.3) no nariz (.4) na boca (.5)
(alternativas não excludentes)
- 4.2.1.1. Na pele:
- 4.2.1.1.1. Tem inchaço? Sim Não
- 4.2.1.1.2. Tem coceira? Sim Não
- 4.2.1.1.3. Tem manchas? Sim Não
- 4.2.1.1.3.1. São: Vermelhas azuis brancas amarelas verdes escuras
- 4.2.1.1.4. Tem feridas? Sim Não
- 4.2.1.1.4.1. Sangram? Sim Não
- 4.2.1.1.4.2. Tem crostas? Sim Não
- 4.2.1.1.4.3. Tem pus? Sim Não
- 4.2.1.1.4.4. Tem descamação? Sim Não
- 4.2.1.1.5. Tem caroços? Sim Não
- 4.2.1.1.6. Sua pele é: seca normal oleosa
- 4.2.1.1.7. Tem alterações nos pelos? Sim Não
- 4.2.1.1.7.1. Tem: queda de pelos excesso de pelos
- 4.2.1.2. Nos olhos:
- 4.2.1.2.1. Está com dificuldade de enxergar? Sim Não
- 4.2.1.2.1.1. de longe de perto
- 4.2.1.2.2. Sua vista está embaçada? Sim Não
- 4.2.1.2.3. Vê pontos na visão? Sim Não
- 4.2.1.2.4. Tem visão das coisas? Sim Não
- 4.2.1.2.5. Tem inchaço? Sim Não
- 4.2.1.2.6. Tem vermelhidão? Sim Não
- 4.2.1.2.7. Tem coceira? Sim Não
- 4.2.1.2.8. A parte branca do seu olho está amarela? Sim Não
- 4.2.1.2.9. Tem hipersensibilidade à luz? Sim Não
- 4.2.1.2.10. Tem secreção ocular? Sim Não
- 4.2.1.2.11. Seu olho está seco? Sim Não

- 4.2.1.2.12. Tem dor? Sim Não
- 4.2.1.2.12.1. Começou: de repente gradualmente
- 4.2.1.2.12.2. Intensidade: fraca moderada intensa muito intensa
- 4.2.1.2.12.3. Dor: em queimação em pontadas pulsátil em aperto fisgada em peso contínua
- 4.2.1.2.12.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
- 4.2.1.2.12.5. Irradiação
- 4.2.1.2.12.6. Tem relação com: ansiedade menstruação leitura tosse exercício posição e/ou movimentos
- 4.2.1.2.13. Tem alucinações visuais? Sim Não
- 4.2.1.3. Nos ouvidos:
- 4.2.1.3.1. Tem dor? Sim Não
- 4.2.1.3.1.1. Começou: de repente gradualmente
- 4.2.1.3.1.2. Intensidade: fraca moderada intensa muito intensa
- 4.2.1.3.1.3. Dor: em queimação em pontadas pulsátil em aperto fisgada em peso contínua
- 4.2.1.3.1.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
- 4.2.1.3.1.5. Irradiação
- 4.2.1.3.1.6. Tem relação com: ansiedade menstruação leitura tosse exercício posição e/ou movimentos (pergunta tipo marcar x)
- 4.2.1.3.2. Tem zumbido? Sim Não
- 4.2.1.3.2.1. Está associado a exposição ao barulho? Sim não
- 4.2.1.3.3. Tem secreção? Sim Não
- 4.2.1.3.3.1. É: cera líquida sangue pus
- 4.2.1.3.4. Você está ouvindo bem? (ou as pessoas estão tendo dificuldade em se comunicar com você?) Sim Não
- 4.2.1.3.5. Você compreende bem o sentido das palavras? Sim Não
- 4.2.1.4. No nariz:
- 4.2.1.4.1. Tem dor ao lado do nariz? Sim Não
- 4.2.1.4.2. Tem sangramento nasal? Sim Não
- 4.2.1.4.3. Tem secreção nasal? Sim Não
- 4.2.1.4.3.1. É: líquida catarro
- 4.2.1.4.4. Tem coceira? Sim Não
- 4.2.1.4.5. Está entupido? Sim Não
- 4.2.1.5. Na boca (inclui os dentes):
- 4.2.1.5.1. Tem dor? Sim Não
- 4.2.1.5.1.1. de dente na gengiva em outro lugar da boca

- 4.2.1.5.2. Tem sangramento? Sim Não
- 4.2.1.5.3. Tem feridas? Sim Não
- 4.2.1.5.4. Tem manchas? Sim Não
- 4.2.1.5.4.1. São: Vermelhas brancas amarelas escuras
- 4.2.1.5.5. Tem boca seca? Sim Não
- 4.2.1.5.6. Tem salivação em excesso? Sim Não
- 4.2.1.5.7. Tem mau hálito constante? Sim Não
- 4.2.1.5.8. Tem alteração da sensibilidade da língua? Sim Não
- 4.2.1.5.9. Está com dificuldade na fala? Sim Não
- 4.2.1.5.10. Tem dor de garganta? Sim Não (testar!)
- 4.2.1.5.11. Tem dificuldade para engolir? Sim Não
- 4.2.1.5.12. Tem rouquidão? Sim Não
- 4.2.1.5.13. Tem muita sede? Sim Não
- 4.2.1.5.14. Está com dificuldade para falar? Sim Não
5. Pescoço:
- 5.1. Tem dor de garganta? Sim Não
- 5.2. Tem nódulos? Sim Não
- 5.3. Tem aumento de volume do pescoço? Sim Não
- 5.4. Tem limitação dos movimentos? Sim Não (ou rigidez?)
- 5.5. Tem rouquidão? Sim Não
- 5.6. Tem dificuldade para engolir? Sim Não
- 5.7. Tem ânsia de vômitos? Sim Não
6. Tronco:
- 6.1. O que você sente está: no peito (acima das costelas)(.1) na barriga(abaixo das costelas)(.2) nas costas(.3) na região da genitália e virilha(.4)
- 6.1.1. No peito:
- 6.1.1.1. Você está sentindo dor? Sim Não
- 6.1.1.1.1. Essa dor varia com a respiração? Sim Não
- 6.1.1.1.2. Onde se localiza a dor ????
- 6.1.1.1.3. Começou: de repente gradualmente
- 6.1.1.1.4. Intensidade: fraca moderada intensa muito intensa
- 6.1.1.1.5. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
- 6.1.1.1.6. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
- 6.1.1.1.7. Irradia: Sim Não
- 6.1.1.1.7.1. Irradia para: o pescoço o ombro esquerdo o ombro direito braço esquerdo braço direito barriga costas cabeça

- 6.1.1.1.8. Começou com esforço físico ? Sim Não
- 6.1.1.1.8.1. pequenos esforços médios esforços grandes esforços
- 6.1.1.2. Você já sentiu que o seu coração disparou de repente? Sim Não
- 6.1.1.3. Você já achou que seu coração tinha parado de bater? Sim não
- 6.1.1.4. Você tem tosse ? Sim não
- 6.1.1.4.1. A tosse: é seca tem escarro branco tem escarro amarelo
- 6.1.1.4.2. No escarro tem sangue? Sim não
- 6.1.1.5. Tem falta de ar? Sim não
- 6.1.1.5.1. Essa falta de ar ocorre: em repouso ao esforço mínimo ao esforço moderado ao esforço grande
- 6.1.1.5.2. Tem falta de ar quando deita? Sim não
- 6.1.1.5.3. Acorda à noite com falta de ar? Sim não
- 6.1.1.6. O seu peito chia durante a respiração? Sim não
- 6.1.1.7. A ponta dos seus dedos está azul? Sim não
- 6.1.1.8. Começa a mancar durante a caminhada ? Sim não
- 6.1.1.9. Tem varizes nas pernas? Sim não
- 6.1.1.10. Tem desmaiado? Sim não (ver cabeça?)
- 6.1.1.11. Sente tonturas quando levanta? Sim não
- 6.1.1.11.1. Sente tontura minutos depois de se levantar? Sim não
- 6.1.1.12. Tem sudorese noturna? Sim não
- 6.1.1.13. Passou muito tempo sentada ? Sim não
- 6.1.1.14. tem alguma alteração da pele nessa região? Sim não
- 6.1.2. Na barriga:
- 6.1.2.1. Tem dor ? Sim não
- 6.1.2.1.1. Onde se localiza essa dor?
- | | | |
|----------------------------|---------------------|----------------------|
| ou na boca do estômago | do seu lado direito | do seu lado esquerdo |
| na região do umbigo | do seu lado direito | do seu lado esquerdo |
| na região abaixo do umbigo | do seu lado direito | do seu lado esquerdo |
- 6.1.2.1.2. Começou: de repente gradualmente
- 6.1.2.1.3. Intensidade: fraca moderada intensa muito intensa
- 6.1.2.1.4. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
- 6.1.2.1.5. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
- 6.1.2.1.6. Irradiação (????)
- 6.1.2.1.7. Começou com esforço físico ? Sim Não
- 6.1.2.1.7.1. pequenos esforços médios esforços grandes esforços

- 6.1.2.2. Tem pirose ? (sensação de queimação da garganta à boca do estômago ou tem azia)
- 6.1.2.3. O seu apetite está: normal aumentado diminuído
- 6.1.2.4. Tem ânsia de vômito? Sim não
- 6.1.2.5. Tem vômitos? Sim não
- 6.1.2.5.1. O vômito tem sangue? Sim não
- 6.1.2.5.2. O vômito tem resto de alimentos? Sim não
- 6.1.2.6. Tem regurgitação de alimentos? Sim não
- 6.1.2.7. A frequência com que está evacuando está: normal aumentada (diarréia)
diminuída (constipação)
- 6.1.2.8. Tem hemorróidas? Sim não
- 6.1.2.9. Tem hérnias? Sim Não
- 6.1.3. Nas costas:
- 6.1.3.1. Tem dor? Sim Não
- 6.1.3.1.1. De que lado? Na coluna lado direito lado esquerdo toda as costas
- 6.1.3.1.2. qual a parte? Da metade pra cima da metade pra baixo
- 6.1.3.1.3. Começou: de repente gradualmente
- 6.1.3.1.4. Intensidade: fraca moderada intensa muito intensa
- 6.1.3.1.5. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
- 6.1.3.1.6. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias
mais de 2 dias
- 6.1.3.1.7. Irradiação
- 6.1.3.1.8. Começou com esforço físico ? Sim Não
- 6.1.3.1.8.1. Pequenos esforços médios esforços grandes esforços
- 6.1.4. Na região da genitália e virilha:
- 6.1.4.1. Tem dor pra urinar? Sim Não
- 6.1.4.2. A quantidade da urina é: normal aumentada diminuída
- 6.1.4.3. A cor da urina está: amarelo clara amarelo escura avermelhada branca
preta (cor de coca-cola)
- 6.1.4.4. Acorda à noite para urinar? Sim Não
- 6.1.4.5. Tem urgência urinária? Sim Não
- 6.1.4.6. Elimina pedras na urina? Sim Não

Sexo feminino:

- 6.1.4.7. A menstruação está regular? Sim Não
- 6.1.4.8. Quando foi a última menstruação? Dd/mm/aaaa não sei
- 6.1.4.9. A duração é: menor que 2 dias de 3 a 7 dias maior que 8 dias

- 6.1.4.10. A relação sexual com seu parceiro é dolorosa? Sim Não
- 6.1.4.11. Está com corrimento? Sim Não
- 6.1.4.11.1. branco amarelado com sangue
- 6.1.4.12. Tem coceira na vagina? Sim Não
- 6.1.4.13. Tem alteração da libido? Sim Não
- 6.1.4.14. Tem feridas? Sim Não

Sexo masculino:

- 6.1.4.15. Tem alteração da libido? Sim Não
- 6.1.4.16. Tem dificuldade de ereção? Sim Não
- 6.1.4.17. Tem dor nos testículos? Sim Não
- 6.1.4.18. Tem caroços nos testículos? Sim Não
- 6.1.4.19. Tem corrimentos no pênis? Sim Não
- 6.1.4.20. Tem ejaculação: normal precoce dolorosa
- 6.1.4.21. Tem feridas? Sim Não
- 7. Nos braços:
 - 7.1. Tem inchaço? Sim Não
 - 7.2. Tem dor? Sim Não
 - 7.2.1. Começou: de repente gradualmente
 - 7.2.2. Intensidade: fraca moderada intensa muito intensa
 - 7.2.3. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
 - 7.2.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
 - 7.2.5. Irradiação
 - 7.2.6. Começou com esforço físico ? Sim Não
 - 7.2.6.1. pequenos esforços médios esforços grandes esforços
 - 7.2.7. Localização: ombro braço cotovelo antebraço punho mão dedos
 - 7.2.8. Sofreu alguma pancada? Sim Não
 - 7.3. Tem tremores? Sim Não
 - 7.4. Tem dificuldade para mexer? Sim Não
- 8. Nos membros inferiores:
 - 8.1. Tem inchaço? Sim Não
 - 8.2. Tem dor? Sim Não
 - 8.2.1. Começou: de repente gradualmente
 - 8.2.2. Intensidade: fraca moderada intensa muito intensa
 - 8.2.3. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua

8.2.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias

8.2.5. Irradiação

8.2.6. Começou com esforço físico ? Sim Não

8.2.6.1. pequenos esforços médios esforços grandes esforços

8.2.7. Localização: coxa joelho batata da perna pé tornozelo

8.2.8. Sofreu alguma pancada? Sim Não

8.3. Tem tremores? Sim Não

8.4. Tem dificuldade para andar? Sim Não

8.5. Começa a mancar durante a caminhada ? Sim não

8.6. Tem fraqueza nas pernas? Sim não

8.7. A panturrilha dói quando você aperta? Sim Não

8.8. A panturrilha está solta? Sim não

8.9. Consegue dobrar o pé para frente e para trás sem dor? Sim Não

8.10. Tem dormência constante? Sim Não

8.11. Tem varizes? Sim Não

Finalmente, como você está se sentindo? Alegre triste Disposto cansado ansioso

APÊNDICE E

Coleta de Sintomas Via Dispositivos Móveis

RT 001-2005

Especificação dos Requisitos do Software

Projeto Goitacá Versão 1.0

Autor: Alexandre Bellezi José

Brasília - DF

Agosto de 2005

Aprovação

Aprovamos o documento de Especificação dos Requisitos 1.0 do projeto GIMPA-GC.

Hervaldo Sampaio Carvalho

Data

Revisões anteriores

Não aplicável

Especificação dos Requisitos do Software Projeto Goitacá Versão 1.0

Especificação dos Requisitos do Software

<u>Projeto Goitacá Versão 1.0.....</u>	<u>2</u>
<u>1 Introdução.....</u>	<u>5</u>
<u>1.1 Objetivos deste documento.....</u>	<u>5</u>
<u>1.2 Escopo do produto.....</u>	<u>5</u>
<u>1.2.1 Nome do produto e de seus componentes principais.....</u>	<u>5</u>
<u>1.2.2 Missão do produto</u>	<u>5</u>
<u>1.2.3 Limites do produto.....</u>	<u>5</u>
<u>1.2.4 Benefícios do produto.....</u>	<u>5</u>
<u>1.3 Materiais de referência</u>	<u>6</u>
<u>1.4 Definições e siglas</u>	<u>6</u>
<u>1.5 Visão geral deste documento.....</u>	<u>6</u>
<u>2 Descrição geral do produto.....</u>	<u>6</u>

2.1	Perspectiva do produto.....	7
2.1.1	Diagrama de contexto.....	7
2.1.2	Interfaces de usuário.....	7
2.1.3	Interfaces de hardware.....	8
2.1.4	Interface de software	8
2.1.5	Interfaces de comunicação.....	8
2.1.6	Restrições de memória.....	8
2.1.7	Modos de operação.....	8
2.1.8	Requisitos de adaptação ao ambiente.....	8
2.2	Funções do produto.....	8
2.3	Usuários.....	9
2.3.1	Descrição.....	9
2.3.2	Características.....	9
2.4	Restrições.....	10
2.5	Hipóteses de trabalho.....	11
2.6	Requisitos adiados.....	11
3	Requisitos específicos.....	12
3.1	Requisitos de interface externa.....	12
3.1.1	Interfaces de usuário.....	12
3.1.1.1	Interface de usuário Tela de Primeiro Acesso.....	12
3.1.1.2	Interface de usuário Tela do Menu Principal.....	13
3.1.1.3	Interface de usuário Tela do Interrogatório Sintomas e Sinais.....	15
3.1.1.4	Interface de usuário Tela do Interrogatório Sintomas por Doenças.....	16
3.1.1.5	Interface de usuário Tela Informativa de Medicações.....	19
3.1.2	Interfaces de hardware.....	20
3.1.3	Interfaces de software.....	20
	Não Aplicável.....	20
3.1.3.1	Interface de usuário Tela de Diagnósticos.....	20
3.1.4	Interfaces de hardware.....	21
3.1.5	Interfaces de software.....	21
3.1.5.1	Interface de usuário Tela de Problemas.....	21
3.1.6	Interfaces de hardware.....	22
3.1.7	Interfaces de software.....	22
3.1.7.1	Interface de usuário Tela de Sincronização.....	23
3.1.8	Interfaces de hardware.....	24
3.1.9	Interfaces de software.....	24
3.1.9.1	Comunicação com GSWEB via protocolo HTTP.....	24
3.1.9.2	Interface de usuário Tela Parar Aplicação.....	24
3.1.10	Interfaces de hardware.....	25
3.1.11	Interfaces de software.....	25
3.1.11.1	Comunicação com GSWEB via protocolo HTTP.....	25
3.2	Requisitos funcionais.....	26
3.2.1	Diagramas de casos de uso.....	26
3.2.1.1	Cenário - Primeiro Acesso.....	26
	Não aplicável.....	27
3.2.1.2	Cenário - Executando Interrogatórios.....	27
	Não aplicável.....	28
3.2.1.3	Cenário - Verificando Medicações.....	28

Não aplicável.....	29
3.2.1.4 Cenário - Verificando Diagnósticos.....	29
Não aplicável.....	29
3.2.1.5 Cenário - Verificando Problemas.....	30
Não aplicável.....	30
3.2.1.6 Cenário – Sincronizar.....	31
Não aplicável.....	31
3.2.1.7	31
3.2.1.8 Cenário – Parar Aplicação.....	32
Não aplicável.....	32
3.3 Requisitos não funcionais.....	32
3.3.1 Requisitos de desempenho.....	32
3.3.1.1 Requisito de desempenho Tempo de resposta.....	32
3.3.2 Requisitos de dados persistentes.....	33
3.3.2.1 Diagrama de classes persistentes.....	33
3.3.3 Restrições ao desenho.....	33
3.3.4 Atributos da qualidade.....	33
3.3.4.1 Atributo da qualidade Segurança do Acesso.....	33
3.3.4.2 Atributo da qualidade Apreensibilidade.....	33
4 Informação de suporte.....	34
4.1 Modelo de análise.....	34
4.1.1 Diagramas de classes.....	34
4.1.1.1 Diagrama de classes Principal.....	34
4.1.1.2 Diagrama de classes Primeiro Acesso.....	35
4.1.1.3 Diagrama de classes Executando Interrogatórios.....	36
4.1.1.4 Diagrama de classes Verificando Medicações.....	36
4.1.1.5 Diagrama de classes Verificando Problemas.....	37
4.1.1.6 Diagrama de classes Verificando Diagnósticos.....	37
4.1.1.7 Diagrama de classes Parar Aplicação.....	37
4.1.2 Classes.....	38
4.1.2.1 Classe Usuário.....	38
4.1.2.2 Classe Paciente.....	38
4.1.2.3 Classe ProfissionalSaude.....	39
4.1.2.4 Classe Médico.....	39
4.1.2.5 Classe Enfermeira.....	39
4.1.2.6 Classe Fisioterapeuta.....	39
4.1.2.7 Classe AuxEnfermagemTecnico.....	39
4.1.2.8 Classe Fonoaudiologo.....	40
4.1.2.9 Classe Nutricionista.....	40
4.1.2.10 Classe Psicologo.....	40
4.1.2.11 Classe Conectar.....	40
4.1.2.12 Classe Sincronizar.....	41
4.1.2.13 Classe Enviar.....	41
4.1.2.14 Classe Receber.....	41
4.1.2.15 Classe BancoLocal.....	41
4.1.3 Realizações dos casos de uso.....	42
4.1.3.1 Realizações do caso de uso Primeiro Acesso.....	42
4.1.3.2 Realizações do caso de uso Executando Interrogatórios.....	43

4.1.3.3 Realizações do caso de uso Verificando Medicações.....	43
4.1.3.4 Realizações do caso de uso Verificando Problemas.....	43
4.1.3.5 Realizações do caso de uso Verificando Diagnósticos.....	44
4.1.3.6 Realizações do caso de uso Parar Aplicação.....	44
4.2 Cadastro dos requisitos.....	45

1 Introdução

1.1 Objetivos deste documento

Descrever e especificar as necessidades de comunicação, envolvendo a troca de dados, entre o produto GOITACÁ e o sistema GSWEB, bem como definir para os desenvolvedores o produto GOITACÁ a ser feito.

Público alvo: cliente, usuários e desenvolvedores do projeto GOITACÁ e GSWEB.

1.2 Escopo do produto

1.2.1 Nome do produto e de seus componentes principais

GOITACÁ: Sistema móvel para coleta de sintomas e sinais.

1) Gerenciamento de sincronismo entre GOITACÁ e GSWEB.

2) Coletor de sintomas.

3) Gerenciamento de banco de dados local no dispositivo móvel.

4) Controle de perfis de acesso ao sistema.

1.2.2 Missão do produto

Sistema informatizado móvel de coleta de sintomas e sinais, e exibição de dados integrado ao GSWEB.

1.2.3 Limites do produto

O GOITACÁ fará a coleta de sintomas e sinais, sincronização de sua base de dados interna com a base de dados do GSWEB.

Também cabe ao GOITACÁ a tarefa de exibir dados relativos, dependendo do perfil do usuário, por exemplo, profissionais de saúde podem ter acesso a informações do prontuário eletrônico do paciente.

É função do GOITACÁ garantir a persistência dos dados coletados e armazenados no dispositivo móvel.

Não faz parte do GOITACÁ garantir a disponibilidade dos serviços providos pelo GSWEB.

Não cabe ao GOITACÁ o controle de gastos referentes ao uso da rede de comunicação fornecida pela operadora de telefonia celular local.

O acesso à rede de comunicação fica a cargo da operadora de telefonia celular local.

1.2.4 Benefícios do produto

Número de ordem	Benefício	Valor para o Cliente
1	Facilidade na coleta de sintomas e sinais	Essencial
2	Confiabilidade na comunicação entre os sistemas GOITACÁ e GSWEB	Essencial
3	Confiabilidade no armazenamento local	Essencial
4	Padronização da informação	Essencial
5	Facilidade de acesso à informação	Essencial
6	Sincronismo sem a intervenção do usuário	Desejável
7	Outros meios de comunicação, SMS ou E-mail.	Desejável
8	Mobilidade aos usuários do sistema	Essencial

1.3 Materiais de referência

Número de ordem	Tipo do material	Referência bibliográfica
1	Entrevistas	Entrevistas com profissionais de saúde.
2	Anais de Congressos	-Workshop de Informática Médica 2005.

1.4 Definições e siglas

Número de ordem	Sigla	Definição
1	Sintomas	Dado evidenciado pelo paciente.
2	Sinais	Informação medida pelo paciente, ou por profissional de saúde.
3	Perfis	Configurações que definem usuários e seus níveis de acesso.
4	GSWEB	Sistema de Gerenciamento de Informações Médicas do Paciente.
5	Dados	Informação proveniente do paciente.
6	Paciente	Agente solicitante da prestação de serviço em saúde.
7	Profissional de Saúde	Presta serviços de saúde.
8	Cadastramento de usuários	Entrada de usuários no sistema.
9	GOITACÁ	Nome do projeto de software, salvo se dito ao contrário, refere-se à versão 1.0.
10	Usuários	Pessoas que farão uso do sistema.
11	Autenticação	Certificação usuário devidamente cadastrado no GSWEB.
12	Unidade de Saúde	Local da prestação de serviço de saúde. Ex: Hospital Universitário de Brasília, Posto de saúde 10 de Venda Nova, etc.
13	PEP	Prontuário Eletrônico do Paciente

1.5 Visão geral deste documento

De acordo com o Padrão para Especificação de Requisitos de Software, ou seja:

Parte 2: Descrição Geral do Produto GOITACÁ

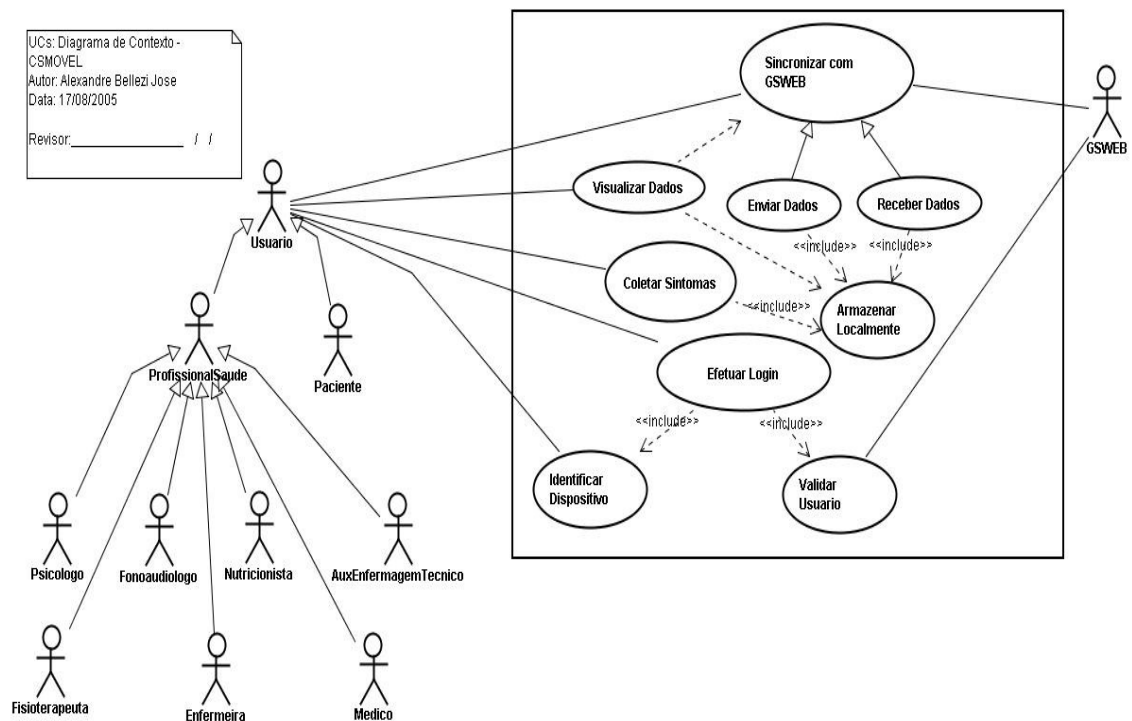
Parte 3: Requisitos Específicos do GOITACÁ

Parte 4: Informação de Suporte: Cadastro dos Requisitos.

2 Descrição geral do produto

2.1 Perspectiva do produto

2.1.1 Diagrama de contexto



2.1.2 Interfaces de usuário

Número de ordem	Nome	Ator	Caso de uso	Descrição
1	Tela de Primeiro Acesso	Usuário	Primeiro Acesso	Tela usada somente no primeiro acesso do usuário ao sistema.
2	Tela de Confirmação dos Dados do Usuário.	Usuário	Checar Dados do Usuário	Tela para confirmar os dados do usuário.
3	Tela do Menu Principal	Usuário	Menu Principal	Tela de acesso às funcionalidades do sistema.

4	Tela do Interrogatório Sintomas e Sinais	Usuário	Interrogatório Sintomas e Sinais	Tela com um conjunto de questões específicas sobre Sintomas e Sinais.
5	Tela do Interrogatório Sintomas por Doenças	Usuário	Interrogatório por Doenças	Tela com um conjunto de questões específicas sobre Sintomas por Doenças.
6	Tela Informativa de Medicações	Usuário	Lembrar Medicações	Tela de lembrete das medicações.
7	Tela de Diagnósticos	Usuário	Mostrar Últimos Diagnósticos	Tela com um conjunto dos últimos diagnósticos.
8	Tela de Problemas	Usuário	Mostra Últimos Problemas	Tela com o conjunto dos últimos problemas.
9	Tela de Sincronização	Usuário	Sincronizar Dados	Tela que permite o sincronismo dos dados do dispositivo móvel com o sistema GSWEB.
10	Tela de Parar Aplicação	Usuário	Sair do GOITACÁ	Tela que possibilita o usuário a encerrar a aplicação ou deixa-la funcionando em segundo plano.

2.1.3 Interfaces de hardware
Não aplicável

2.1.4 Interface de software

Número de ordem	Nome	Atores	Casos de Uso	Sigla, versão e fornecedor	Descrição
1	Sincronizar Dados	GSWEB	Sincronizar com GSWEB; Validar Usuário.		O propósito desta interface é recuperar informações do PEP.

2.1.5 Interfaces de comunicação
Não aplicável.

2.1.6 Restrições de memória

Número de ordem	Tipo de memória	Limites aplicáveis
1	RAM do Dispositivo Móvel	Maior que 500Kb, para aplicação e base de dados.

2.1.7 Modos de operação

Número de ordem	Tipo de operação	Descrição da operação	Detalhes de operação
1	Interativa	Modo de Coleta de Evidências	Modo de operação onde o GOITACÁ realizar os conjuntos de questões pertinentes aos interrogatórios disponíveis.
2	Segundo Plano	Sincronismo com GSWEB	Modo onde o sistema irá sincronizar com o servidor remoto sua base de dados, desde que uma rede esteja disponível.

2.1.8 Requisitos de adaptação ao ambiente
Não aplicável

2.2 Funções do produto

Número de ordem	Caso de uso	Descrição
1	Checar Dados do Usuário	Funcionalidade executado quando o sistema é inicializado pela primeira vez, para que o usuário confirme seus dados.
2	Interrogatório Sintomas e Sinais	Capaz de colher os Sintomas e Sinais e agrupa-os para armazenamento.
3	Interrogatório por Doenças	Capaz de colher os Sintomas e Sinais, classificados de acordo com os problemas do paciente, e agrupa-os para armazenamento.
4	Lembrar Medicações	Função capaz de emitir avisos temporais sobre uso de medicações.
5	Mostrar Últimos Diagnósticos	Exibe uma listagem dos últimos diagnósticos.
6	Mostra Últimos Problemas	Exibe uma listagem dos últimos problemas.
7	Sincronizar Dados	Funcionalidade que pode ser executada a qualquer momento, pelo usuário, para atualizar o banco de dados local com o bando de dados do GSWEB.
8	Parar Aplicação	Capaz de parar definitivamente a aplicação, o que implica no termino dos lembretes sobre medicação se os mesmo existirem.

2.3 Usuários

2.3.1 Descrição

Número de ordem	Ator	Definição
1	Usuário	Usuário genérico do sistema, somente para fins de modelagem Orientada a Objetos.
2	Profissional de Saúde	Responsável pela entrada de dados referentes à avaliação clínica, procedimentos e exames realizados pelo paciente.
3	Médico	Acessa todo o sistema.
4	Enfermeira	Acessa todo o sistema.
5	Auxiliar de Enfermagem	Acessa todo o sistema.
6	Fisioterapeuta	Acessa todo o sistema.
7	Psicólogo	Acessa todo o sistema.
8	Fonoaudiólogo	Acessa todo o sistema.
9	Nutricionista	Acessa todo o sistema.
10	Paciente	Acessa todo o sistema.

2.3.2 Características

Número de ordem	Atores	Permissão de acesso	Frequência de uso	Nível educacional	Proficiência na aplicação	Proficiência em Informática
1	Profissional de Saúde	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	1º Grau	Operacional	Aplicação
2	Médico	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux
3	Enfermeira	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux

4	Auxiliar de Enfermagem	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	2º Grau	Operacional	Aplicação
5	Fisioterapeuta	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux
6	Psicólogo	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux
7	Fonoaudiólogo	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux
8	Nutricionista	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux
9	Paciente	Lê todo o sistema. E executa os Interrogatórios	Diário 24h por dia	3º Grau	Completa	Aplicação Linux

2.4 Restrições

Número de ordem	Restrição	Descrição
1	Ambiente	O ambiente operacional a ser utilizado é o KVM. Máquina virtual para J2ME.
2	Ambiente	O sistema deverá executar em um dispositivo móvel que suporte a tecnologia Java J2ME.
4	Expansibilidade	O produto deve ser desenvolvido levando-se em consideração que poderá ser expandido para mais de uma unidade de saúde.
5	Legal	O produto deverá estar em conformidade com as leis e regulamentos vigentes na época do início do Desenvolvimento.
6	Segurança	O produto deverá restringir o acesso através de senhas individuais para cada usuário.

2.5 Hipóteses de trabalho

Número de ordem	Hipótese	De quem depende
1	Deve ser utilizada a conexão HTTP provida pela operadora de telefonia celular.	O cliente deve arcar com os custos e contrato com a operadora de telefonia celular.

2.6 Requisitos adiados

Não aplicável

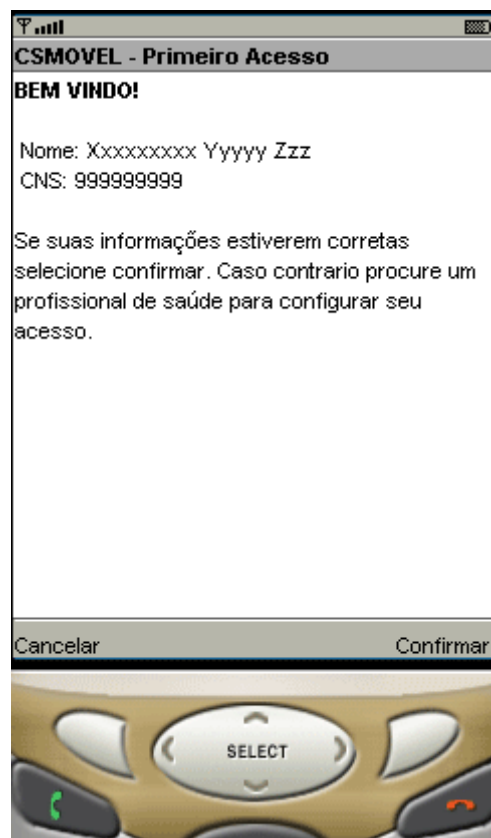
3 Requisitos específicos

3.1 Requisitos de interface externa

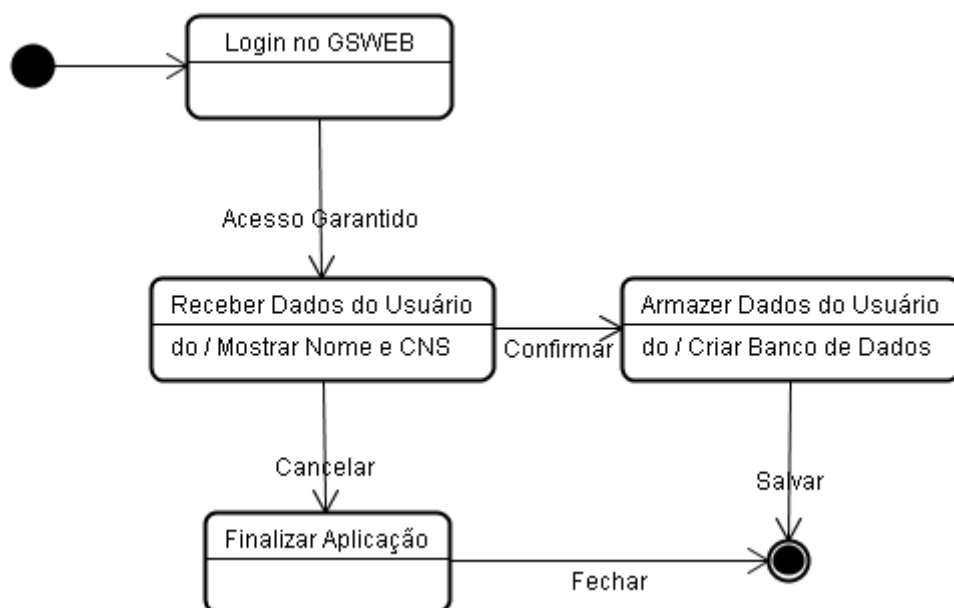
3.1.1 Interfaces de usuário

3.1.1.1 Interface de usuário Tela de Primeiro Acesso

3.1.1.1.1 Layout sugerido



3.1.1.1.2 Diagrama de estados



3.1.1.1.3 Relacionamentos com outras interfaces

O botão Cancelar finaliza a aplicação e retorna ao meu do Sistema Operacional.

O botão Confirmar aciona a interface “Tela do Menu Principal”.

3.1.1.1.4 Campos

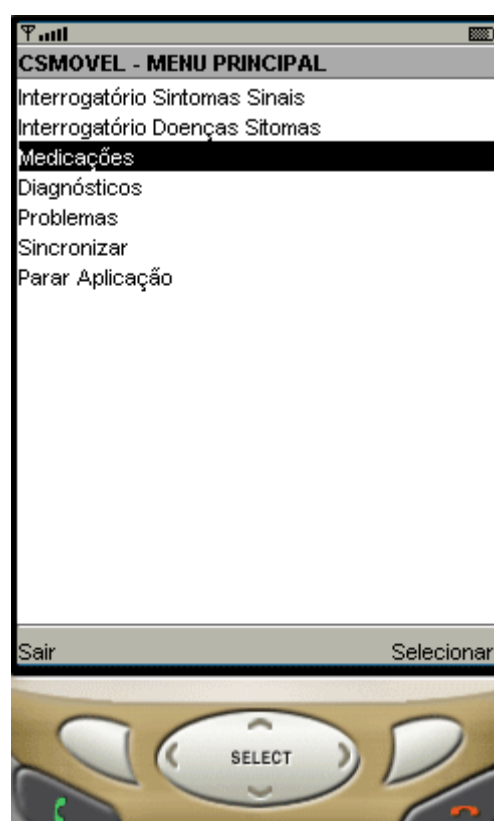
Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Nome	Texto	Até 30 caracteres.	Literal	Somente Leitura
2	CNS	Número	Até 9 dígitos.	Númerico	Somente Leitura

3.1.1.1.5 Comandos

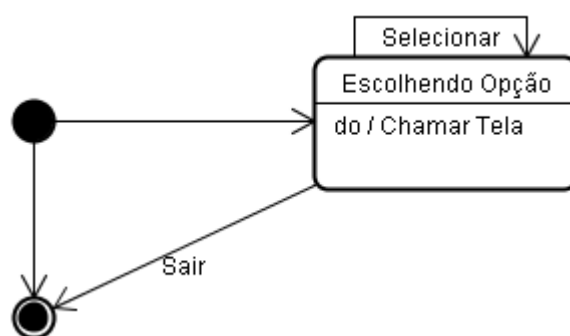
Número	Nome	Ação	Restrições
1	Confirmar	Libera o acesso a Tela do Menu Principal.	Habilitado somente quando o usuário estiver cadastrado e a ID do dispositivo estiver correta.
2	Cancela	Fecha a interface.	Sempre habilitado.

3.1.1.2 Interface de usuário Tela do Menu Principal

3.1.1.2.1 Layout sugerido



3.1.1.2.2 Diagrama de estados



3.1.1.2.3 Relacionamentos com outras interfaces

O botão Sair fecha a aplicação e retorna para o Sistema Operacional. Mas mantém o sistema rodando em segundo plano para possibilitar o lembrete das medicações.

O botão Selecionar ativa a opção correspondente. As opções são listadas abaixo:

A opção Interrogatório Sintomas Sinais inicia o questionário correspondente.

A opção Interrogatório Sintomas por Doenças inicia o questionário correspondente.

A opção Medicação mostra uma lista das medicações em uso atualmente.

A opção Diagnósticos mostra uma lista dos últimos diagnósticos.

A opção Problemas mostra uma lista dos problemas atuais do paciente.

A opção Sincronizar irá conectar-se ao sistema GSWEB e atualizar a base de dados, envio e recebimento de informações.

A opção Parar Aplicação é utilizada para encerrar o aplicativo de maneira definitiva, ou seja, ele não ficará mais sendo executado em segundo plano.

Caso a opção Para Aplicação tenha sido selecionada e confirmada, o Menu Principal passará a exibir no seu local, Iniciar Aplicação.

3.1.1.2.4 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Interrogatório Sintomas e Sinais	Selecionado ou não selecionado.	-	-	Sempre habilitado.
2	Interrogatório Sintomas por Doenças	Selecionado ou não selecionado.	-	-	Sempre habilitado.

3	Medicação	Selecionado ou não selecionado.	-	-	Sempre habilitado.
4	Problemas	Selecionado ou não selecionado.	-	-	Sempre habilitado.
5	Diagnósticos	Selecionado ou não selecionado.	-	-	Sempre habilitado.
6	Sincronizar	Selecionado ou não selecionado.	-	-	Sempre habilitado.
7	Parar Aplicação	Selecionado ou não selecionado.	-	-	Sempre habilitado.

3.1.1.2.5 Comandos

Número	Nome	Ação	Restrições
1	Selecionar	Muda para a interface corresponde na seleção.	Sempre habilitado.
2	Sair	Fecha o sistema e retorna ao Sistema Operacional.	Sempre habilitado.

3.1.1.3 Interface de usuário Tela do Interrogatório Sintomas e Sinais

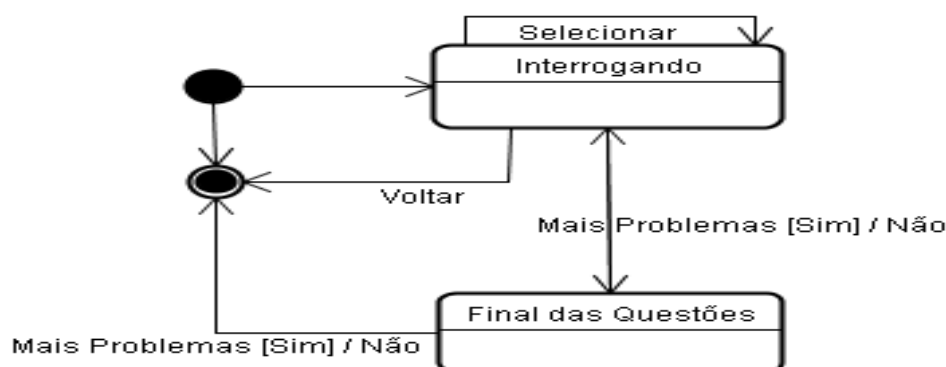
3.1.1.3.1 Descrição

Permite a coleta de Sintomas e Sinais evidenciados pelo Paciente.

3.1.1.3.2 Layout sugerido



3.1.1.3.3 Diagrama de estados



3.1.1.3.4 Relacionamentos com outras interfaces

Se a interface estiver no estado “Questionando” o botão voltar retorna ao menu principal.

3.1.1.3.5 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Pergunta	Texto	-	-	Obrigatório / não alterável.
2	Opções de Resposta	Texto	-	-	Obrigatório / não alterável.

3.1.1.3.6 Comandos

Número	Nome	Ação	Restrições
1	Selecionar	Confirma a alternativa escolhida.	Sempre habilitado.
2	Voltar	Retorna ao Menu Principal	Sempre habilitado.

3.1.1.4 Interface de usuário Tela do Interrogatório Sintomas por Doenças

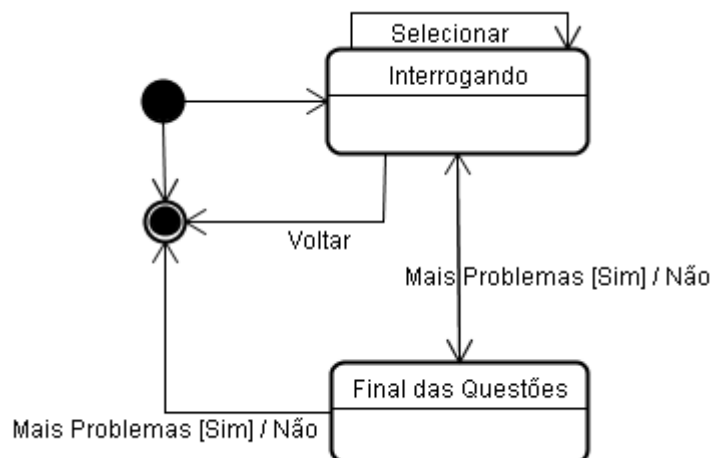
3.1.1.4.1 Descrição

Permite a coleta de Sintomas relacionados à determinada doença evidenciados pelo Paciente

3.1.1.4.2 Layout sugerido



3.1.1.4.3 Diagrama de estados



3.1.1.4.4 Relacionamentos com outras interfaces

Se a interface estiver no estado “Questionando” o botão voltar retorna ao menu principal.

3.1.1.4.5 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Pergunta	Texto	-	-	Obrigatório / não alterável.
2	Opções de Resposta	Texto	-	-	Obrigatório / não alterável.

3.1.1.4.6 Comandos

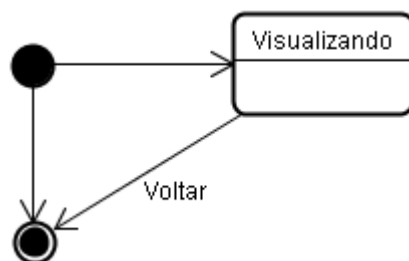
Número	Nome	Ação	Restrições
1	Selecionar	Confirma a alternativa escolhida.	Sempre habilitado.
2	Voltar	Retorna ao Menu Principal	Sempre habilitado.

3.1.1.5 Interface de usuário Tela Informativa de Medicamentos

3.1.1.5.1 Layout sugerido



3.1.1.5.2 Diagrama de estados



3.1.1.5.3 Relacionamentos com outras interfaces

O botão Voltar retorna ao Menu Principal.

3.1.1.5.4 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Medicação	Selecionado ou não selecionado.	-	-	Sempre habilitado.
2	Posologia	Selecionado ou não selecionado.	-	-	Sempre habilitado.
3	Horário	Selecionado ou não selecionado.	-	-	Sempre habilitado.

3.1.1.5.5 Comandos

Número	Nome	Ação	Restrições
2	Voltar	Volta ao Menu Principal	Sempre habilitado.

3.1.2 Interfaces de hardware
Não aplicável.

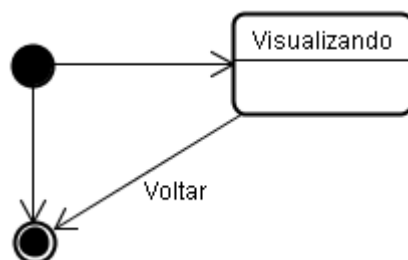
3.1.3 Interfaces de software
Não Aplicável

3.1.3.1 Interface de usuário Tela de Diagnósticos

3.1.3.1.1 Layout sugerido



3.1.3.1.2 Diagrama de estados



3.1.3.1.3 Relacionamentos com outras interfaces

O botão Voltar retorna ao Menu Principal.

3.1.3.1.4 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Diagnóstico	Selecionado ou não selecionado.	-	-	Sempre habilitado.
2	Data	Selecionado ou não selecionado.	-	-	Sempre habilitado.

3.1.3.1.5 Comandos

Número	Nome	Ação	Restrições
2	Voltar	Volta ao Menu Principal	Sempre habilitado.

3.1.4 Interfaces de hardware

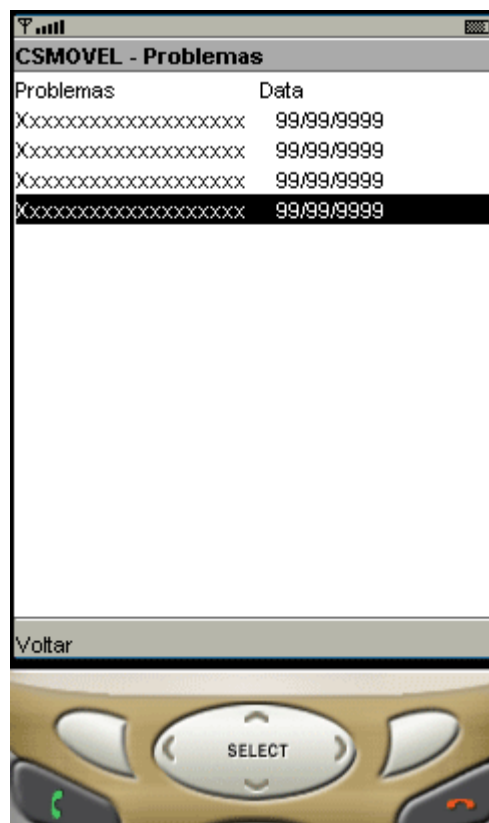
Não aplicável.

3.1.5 Interfaces de software

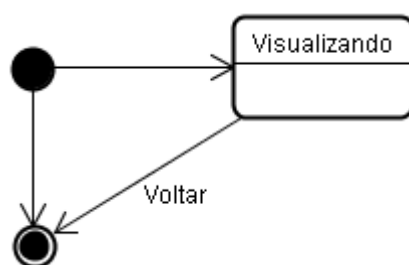
Não aplicável.

3.1.5.1 Interface de usuário Tela de Problemas

3.1.5.1.1 Layout sugerido



3.1.5.1.2 Diagrama de estados



3.1.5.1.3 Relacionamentos com outras interfaces

O botão Voltar retorna ao Menu Principal.

3.1.5.1.4 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Problemas	Selecionado ou não selecionado.	-	-	Sempre habilitado.
2	Data	Selecionado ou não selecionado.	-	-	Sempre habilitado.

3.1.5.1.5 Comandos

Número	Nome	Ação	Restrições
2	Voltar	Volta ao Menu Principal	Sempre habilitado.

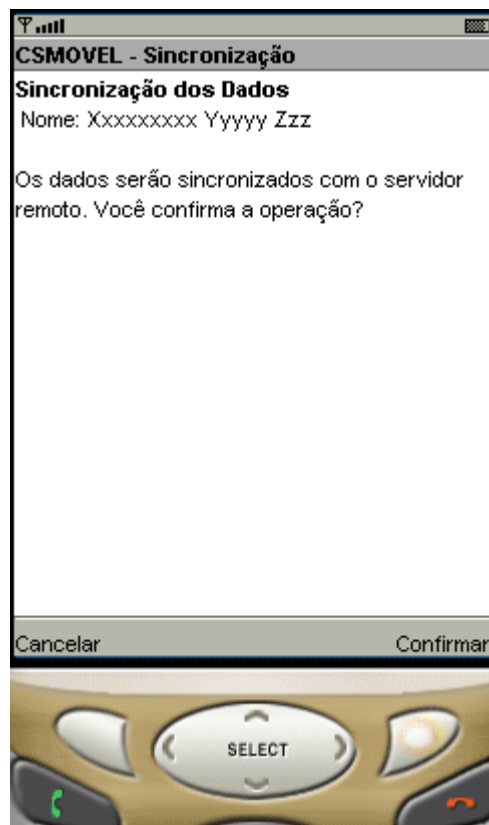
3.1.6 Interfaces de hardware

Não aplicável.

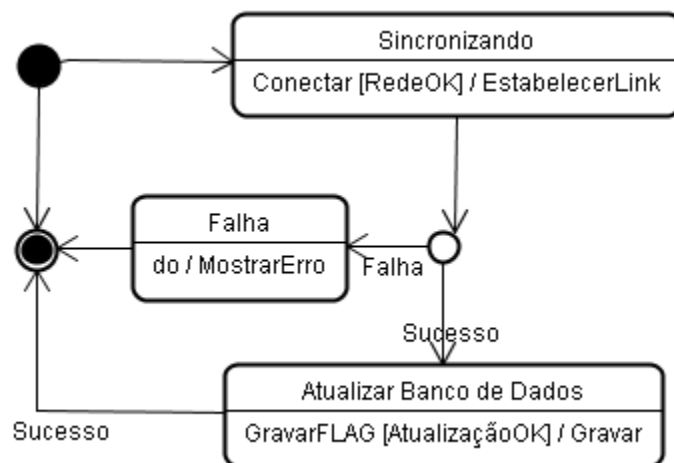
3.1.7 Interfaces de software

Não aplicável.

3.1.7.1 Interface de usuário Tela de Sincronização
3.1.7.1.1 Layout sugerido



3.1.7.1.2 Diagrama de estados



3.1.7.1.3 Relacionamentos com outras interfaces

O botão Cancelar retorna ao Menu Principal.

3.1.7.1.4 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Nome	Não selecionado.	-	-	Sempre habilitado.

3.1.7.1.5 Comandos

Número	Nome	Ação	Restrições
1	Cancelar	Volta ao Menu Principal	Sempre habilitado.
2	Confirmar	Inicia a sincronização com o servidor do GSWEB.	Sempre habilitado.

3.1.8 Interfaces de hardware

Não aplicável.

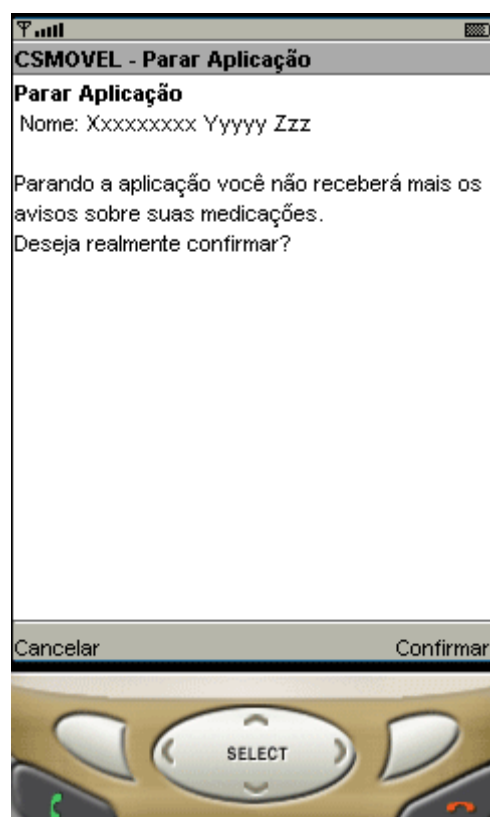
3.1.9 Interfaces de software

3.1.9.1 Comunicação com GSWEB via protocolo HTTP.

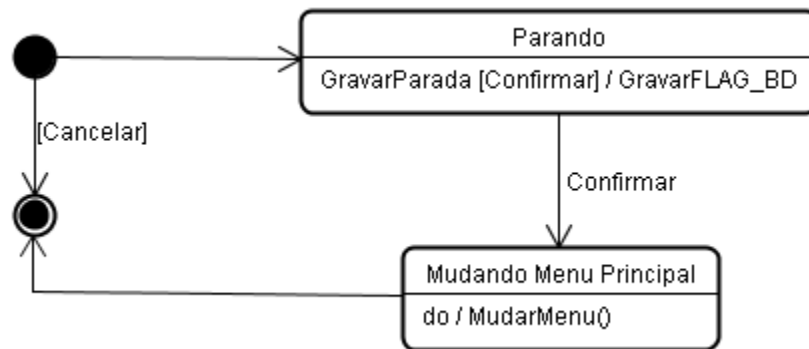
O GOITACÁ estabelecerá uma conexão via HTTP com o GSWEB, para atualização das bases de dados. Segue o padrão do protocolo HTTP para envio e recebimento de informações.

3.1.9.2 Interface de usuário Tela Parar Aplicação

3.1.9.2.1 Layout sugerido



3.1.9.2.2 Diagrama de estados



3.1.9.2.3 Relacionamentos com outras interfaces

O botão Cancelar retorna ao Menu Principal.

O botão Confirmar grava a FLAG de parada no Banco de Dados e muda o menu principal para Iniciar Aplicação.

3.1.9.2.4 Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Nome	Não selecionado.	-	-	Sempre habilitado.

3.1.9.2.5 Comandos

Número	Nome	Ação	Restrições
1	Cancelar	Volta ao Menu Principal	Sempre habilitado.
2	Confirmar	Inicia a sincronização com o servidor do GSWEB.	Sempre habilitado.

3.1.10 Interfaces de hardware

Não aplicável.

3.1.11 Interfaces de software

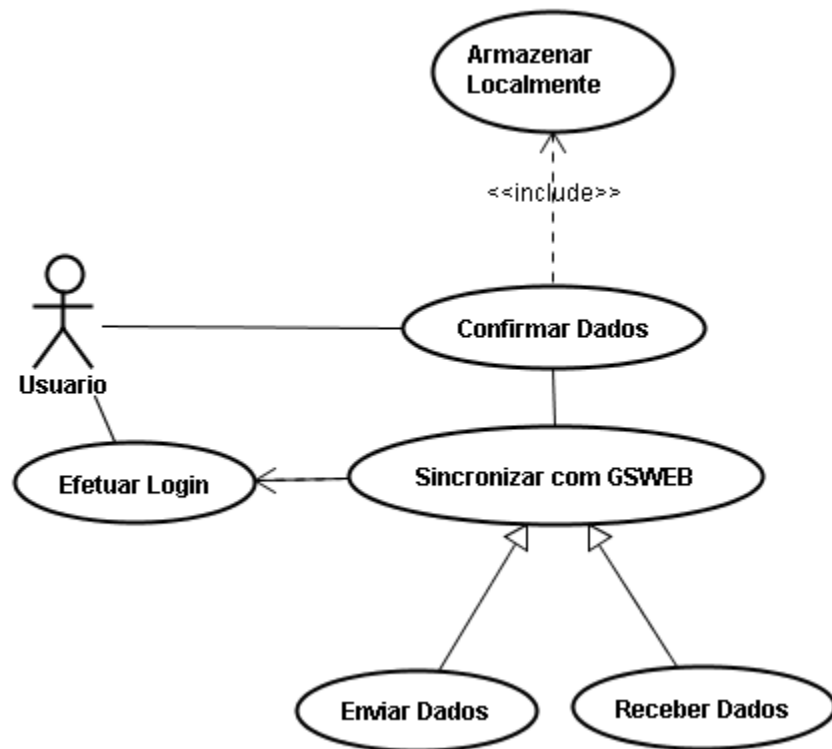
3.1.11.1 Comunicação com GSWEB via protocolo HTTP.

O GOITACÁ estabelecerá uma conexão via HTTP com o GSWEB, para atualização das bases de dados. Segue o padrão do protocolo para envio e recebimento de informações.

3.2 Requisitos funcionais

3.2.1 Diagramas de casos de uso

3.2.1.1 Cenário - Primeiro Acesso



3.2.1.1.1 Precondições

Só ocorre quando o GOITACÁ é inicializado pela primeira vez.

3.2.1.1.2 Fluxo principal

O Usuário inicia o sistema pela primeira vez.

Se houver serviços de rede disponível para o dispositivo móvel o mesmo inicia o processo de autenticação com o servidor GSWEB. Neste primeiro acesso os dados serão recebidos do GSWEB.

Após receber os dados, a identificação do Usuário deve ser exibida para que o mesmo confirme se são realmente seus os dados obtidos no GSWEB.

Se o Usuário confirmar os dados, o sistema desvia para o subfluxo Gravar Dados.

Caso contrário, o Usuário recebe uma tela de mensagem indicando que o mesmo deve procurar um administrador do GSWEB para que seja feito o cadastro de seu dispositivo móvel junto ao sistema.

3.2.1.1.3 Subfluxos

3.2.1.1.3.1 Subfluxo Armazenar Localmente

Se já houver um Banco de Dados no dispositivo móvel, então se deve checar o FLAG de lembrete de medicações e permitir que o Usuário escolha se deseja ou não ativar os lembretes.

Caso contrário seguir fluxo alternativo.

Os dados recebidos do GSWEB são armazenados localmente no dispositivo móvel.

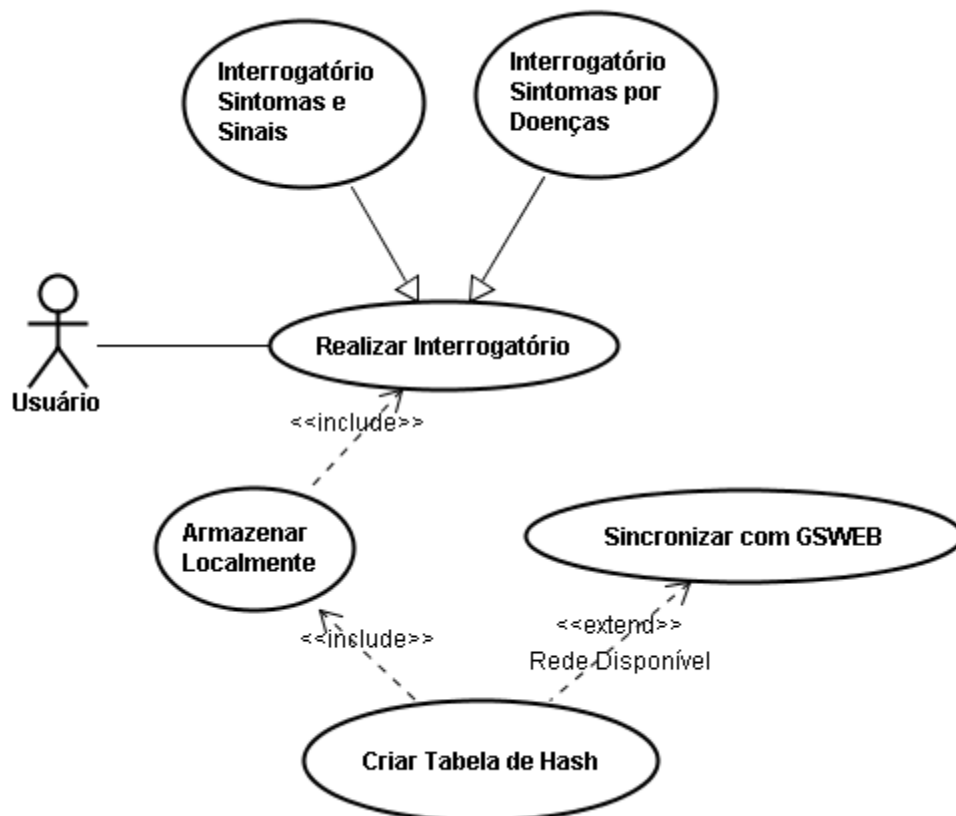
3.2.1.1.4 Fluxos alternativos

Um banco de dados deve ser criado no dispositivo móvel.

3.2.1.1.5 Diagrama de estado / Diagrama de atividade / Fluxograma

Não aplicável.

3.2.1.2 Cenário - Executando Interrogatórios



3.2.1.2.1 Precondições

GOITACÁ deve estar no modo de Interrogatórios.

3.2.1.2.2 Fluxo principal

O Usuário inicia o recurso de interrogatórios.

Se o Usuário selecionar a opção Voltar o sistema retorna ao Menu Principal.

O interrogatório segue até que o Usuário decida que não há mais nenhum problema.

Então o fluxo é desviado para o subfluxo Criar Tabela de Hash.

3.2.1.2.3 Subfluxos

3.2.1.2.3.1 Subfluxo Criar Tabela de Hash

Os dados colhidos pelo interrogatório são armazenados em uma tabela de Hash.

E então enviados ao subfluxo 3.2.1.1.3.1

Se um serviço de conexão de rede estiver disponível, então seguir subfluxo Sincronizar com GSWEB.

3.2.1.2.3.2 Subfluxo Sincronizar

Após o armazenamento local, efetuar a autenticação no GSWEB.

Transmitir os dados do interrogatório para o GSWEB.

Se transmissão bem sucedida.

Realizar a configuração do FLAG de estado no banco de dados local.

Senão manter o FLAG como não sincronizado.

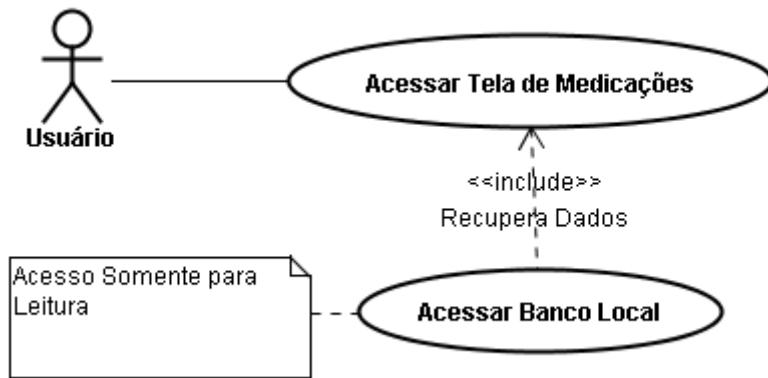
3.2.1.2.4 Fluxos alternativos

Não aplicável.

3.2.1.2.5 Diagrama de estado / Diagrama de atividade / Fluxograma

Não aplicável.

3.2.1.3 Cenário - Verificando Medicções



3.2.1.3.1 Precondições

GOITACÁ deve estar no modo de Verificar Medicacoes.

3.2.1.3.2 Fluxo principal

O Usuário inicia o recurso de Verificação de Medicacoes.

Se o Usuário selecionar a opção Voltar o sistema retorna ao Menu Principal.

O Usuário pode usar a tecla de rolagem para ver a lista de medicacoes ativas.

3.2.1.3.3 Subfluxos

Não aplicável.

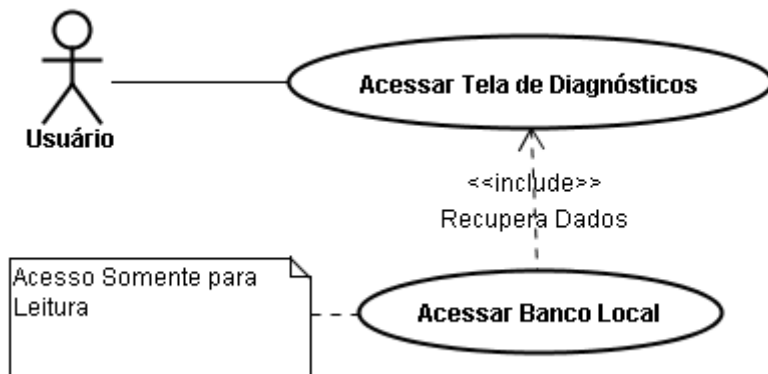
3.2.1.3.4 Fluxos alternativos

Não aplicável.

3.2.1.3.5 Diagrama de estado / Diagrama de atividade / Fluxograma

Não aplicável.

3.2.1.4 Cenário - Verificando Diagnósticos



3.2.1.4.1 Precondições

GOITACÁ deve estar no modo de Verificar Diagnósticos.

3.2.1.4.2 Fluxo principal

O Usuário inicia o recurso de Verificar Diagnósticos.

Se o Usuário selecionar a opção Voltar o sistema retorna ao Menu Principal.

O Usuário pode usar a tecla de rolagem para ver a lista dos últimos diagnósticos.

3.2.1.4.3 Subfluxos

Não aplicável.

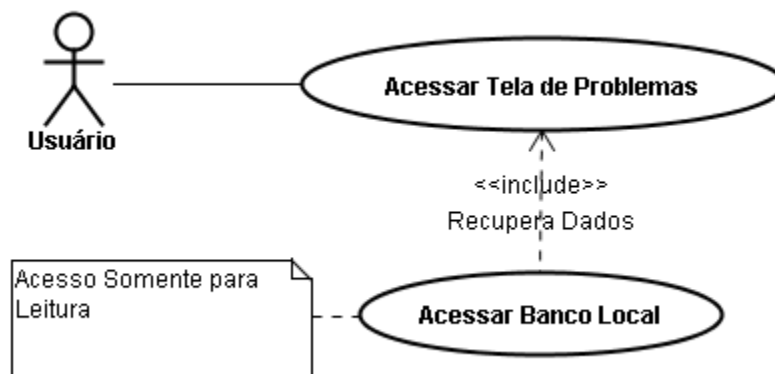
3.2.1.4.4 Fluxos alternativos

Não aplicável.

3.2.1.4.5 Diagrama de estado / Diagrama de atividade / Fluxograma

Não aplicável.

3.2.1.5 Cenário - Verificando Problemas



3.2.1.5.1 Precondições

GOITACÁ deve estar no modo de Verificar Problemas.

3.2.1.5.2 Fluxo principal

O Usuário inicia o recurso de Verificar Problemas.

Se o Usuário selecionar a opção Voltar o sistema retorna ao Menu Principal.

O Usuário pode usar a tecla de rolagem para ver a lista dos últimos problemas.

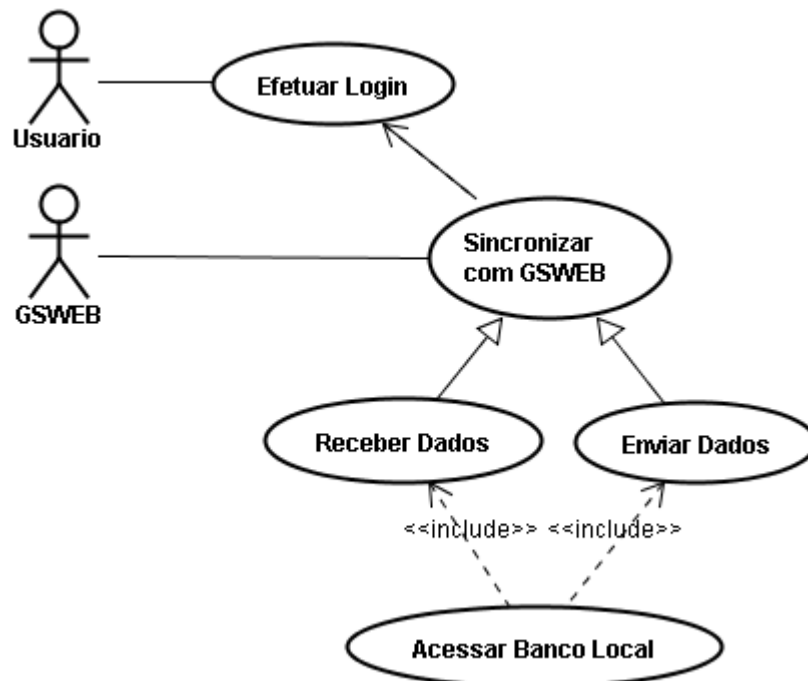
3.2.1.5.3 Subfluxos

Não aplicável.

3.2.1.5.4 Fluxos alternativos
Não aplicável.

3.2.1.5.5 Diagrama de estado / Diagrama de atividade / Fluxograma
Não aplicável.

3.2.1.6 Cenário – Sincronizar



3.2.1.6.1 Precondições

GOITACÁ deve estar no modo de Sincronização ou pode ter detectado uma rede de comunicação e FLAG no Banco de Dados indica que não houve sincronização no último interrogatório, passando assim para o modo de Sincronização.

3.2.1.6.2 Fluxo principal

O Usuário inicia o recurso de Sincronização, ou uma rede é detectada e o FLAG do Banco de Dados indica que não houve sincronização no último interrogatório.

O GOITACÁ inicia o processo de login.

Se conexão bem sucedida. Então acessa Banco de Dados Local e envia os dados ao GSWEB. Muda a FLAG de sincronização do Banco de Dados Local para sincronizado.

3.2.1.6.3 Subfluxos

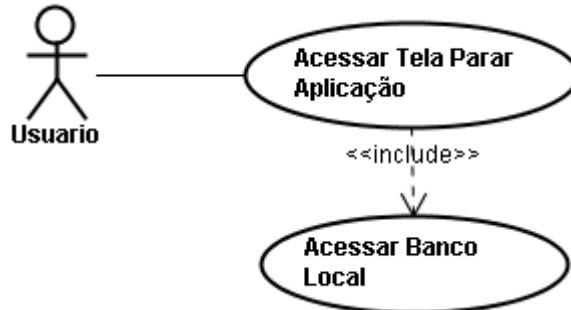
Não aplicável.

3.2.1.6.4 Fluxos alternativos

Não aplicável.

3.2.1.6.5 Diagrama de estado / Diagrama de atividade / Fluxograma
Não aplicável.

3.2.1.7 Cenário – Parar Aplicação



3.2.1.7.1 Precondições

GOITACÁ deve estar no modo de Parar Aplicação.

3.2.1.7.2 Fluxo principal

O Usuário inicia o recurso de Parar Aplicação.

Se o Usuário selecionar a opção Voltar o sistema retorna ao Menu Principal.

Se o Usuário Confirmar, o FLAG no Banco Local sobre os lembretes das medicações é configurado para FALSE e o sistema é descarregado da memória. E a opção Parar Aplicação no Menu Principal fica inacessível.

3.2.1.7.3 Subfluxos

Não aplicável

3.2.1.7.4 Fluxos alternativos

Não aplicável.

3.2.1.7.5 Diagrama de estado / Diagrama de atividade / Fluxograma

Não aplicável.

3.3 Requisitos não funcionais

Não aplicável.

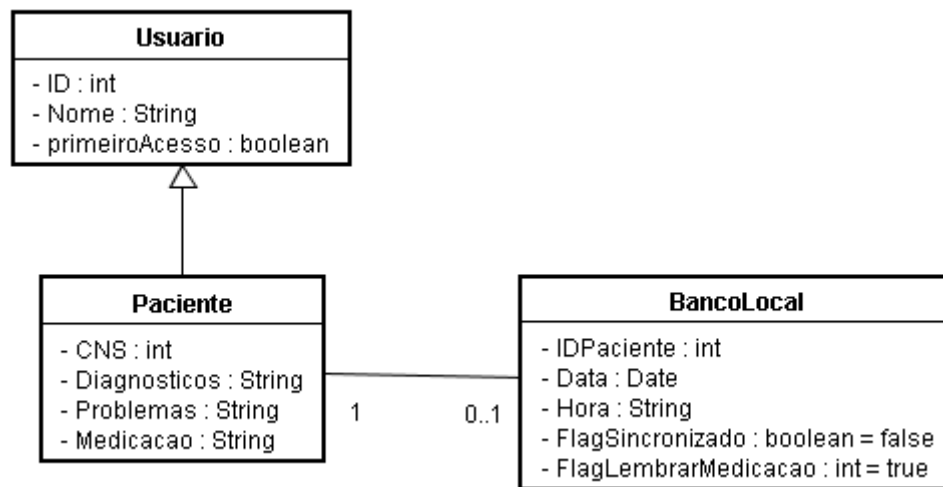
3.3.1 Requisitos de desempenho

3.3.1.1 Requisito de desempenho Tempo de resposta

A sincronização com o GSWEB não deverá durar mais do que 30 segundos.

3.3.2 Requisitos de dados persistentes

3.3.2.1 Diagrama de classes persistentes



3.3.3 Restrições ao desenho

Não aplicável

3.3.4 Atributos da qualidade

3.3.4.1 Atributo da qualidade Segurança do Acesso

O GOITACÁ só permitirá acesso aos usuários que tiverem seus dispositivos móveis devidamente cadastrados no GSWEB.

3.3.4.2 Atributo da qualidade Apreensibilidade

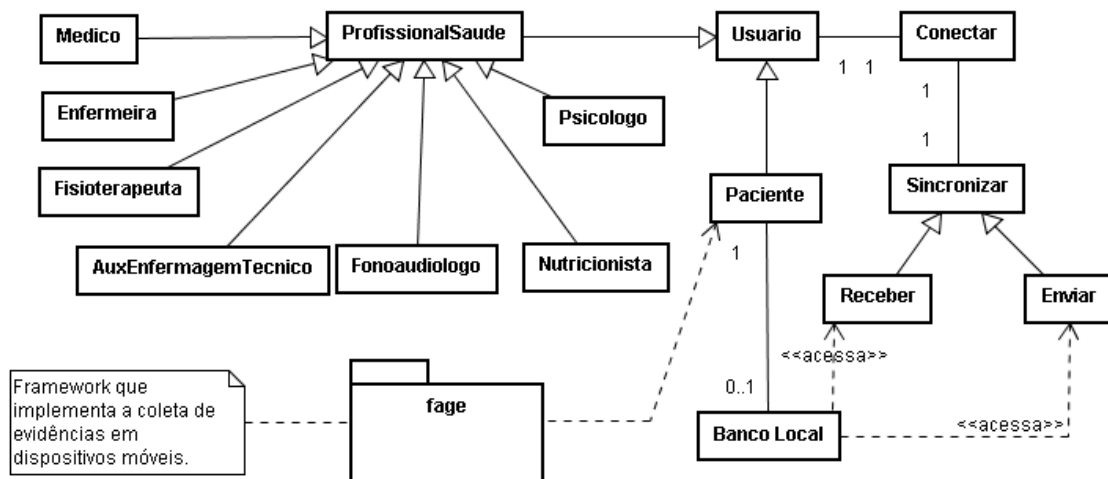
O Usuário deverá estar apto a operar o GOITACÁ em um dia de contato com o sistema.

4 Informação de suporte

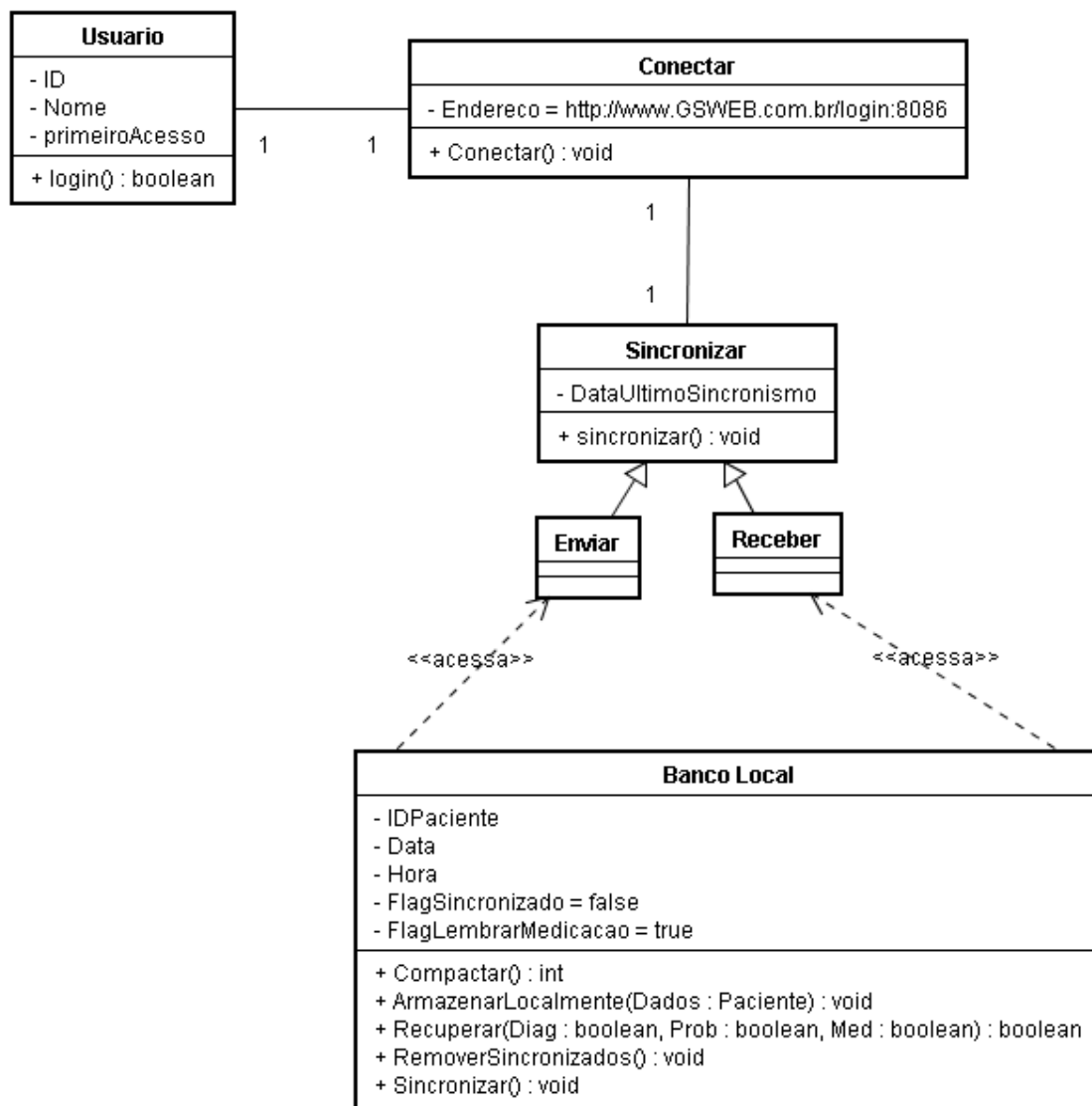
4.1 Modelo de análise

4.1.1 Diagramas de classes

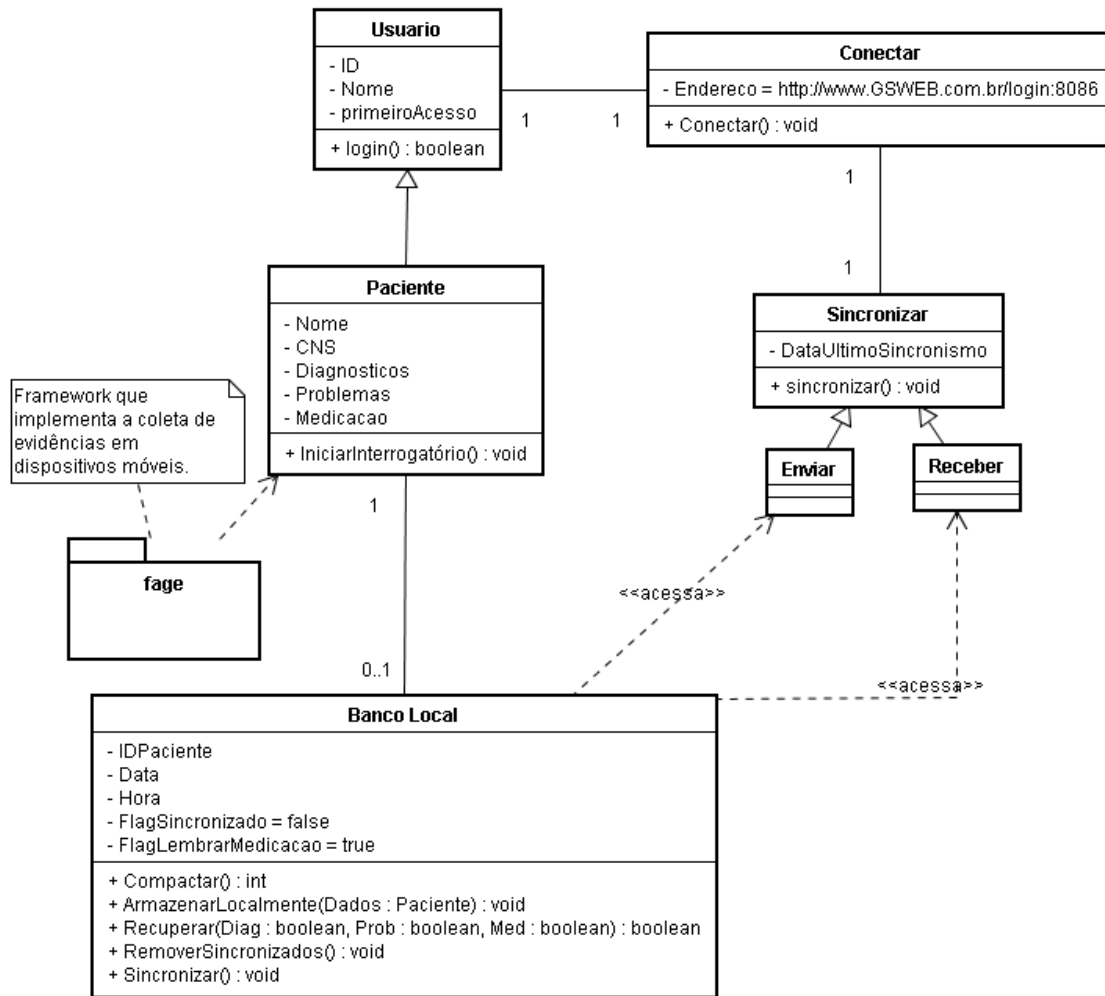
4.1.1.1 Diagrama de classes Principal



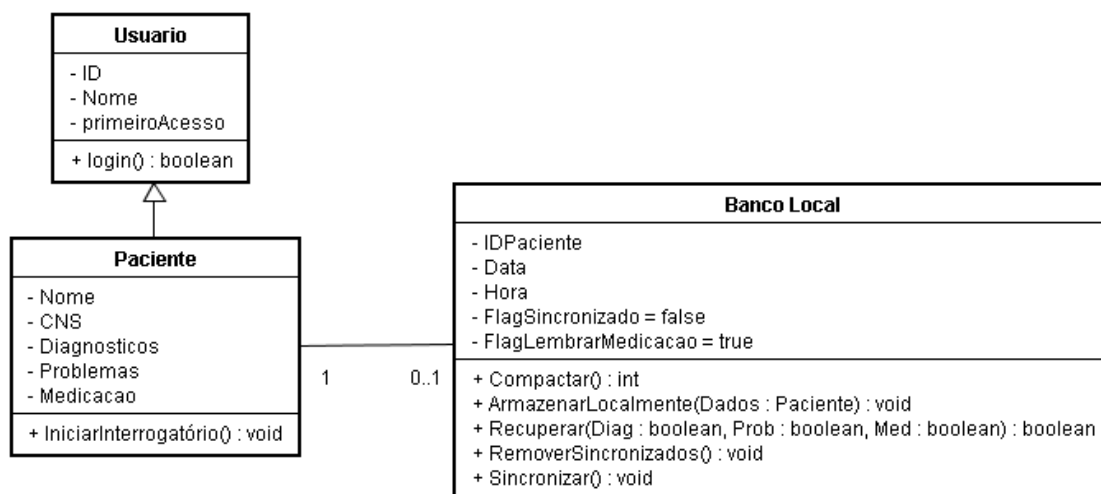
4.1.1.2 Diagrama de classes Primeiro Acesso



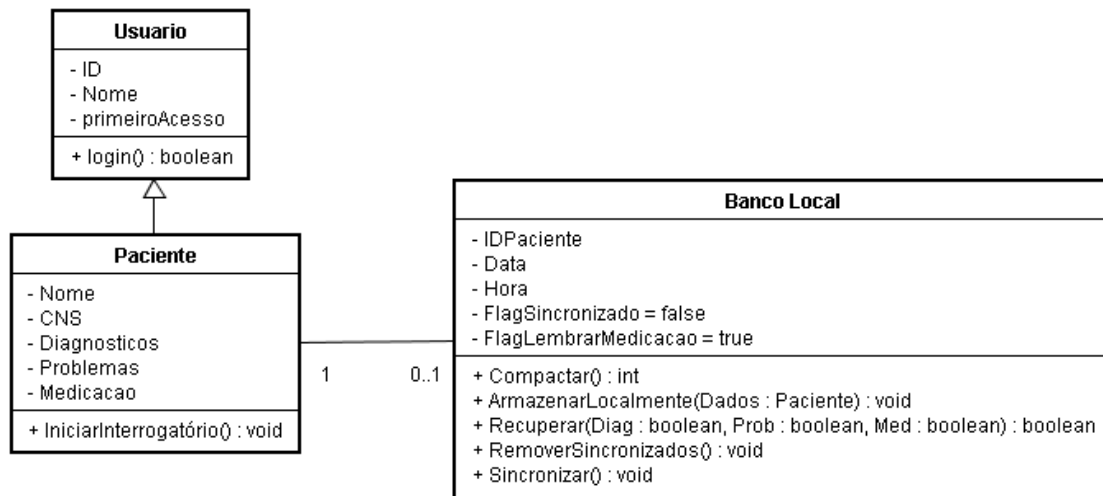
4.1.1.3 Diagrama de classes Executando Interrogatórios



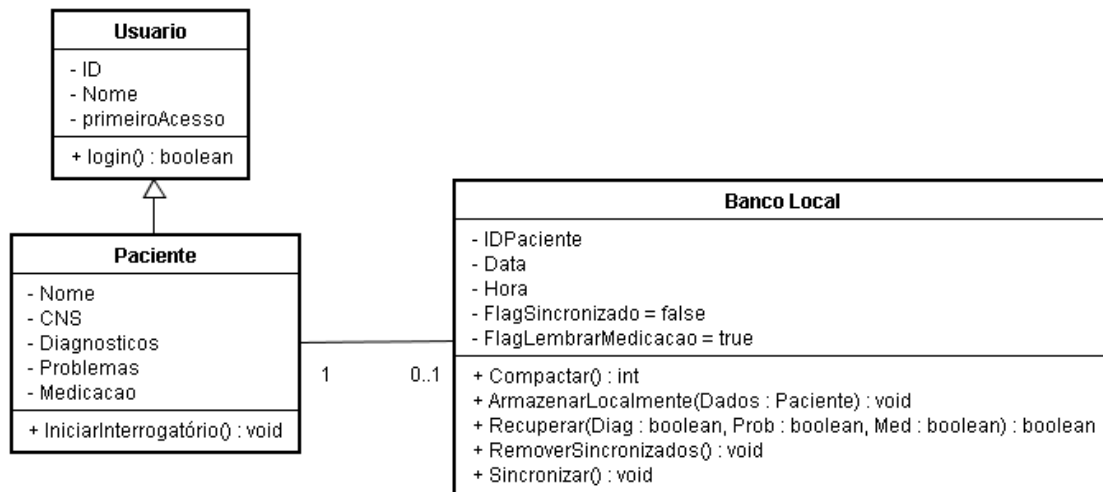
4.1.1.4 Diagrama de classes Verificando Medicações



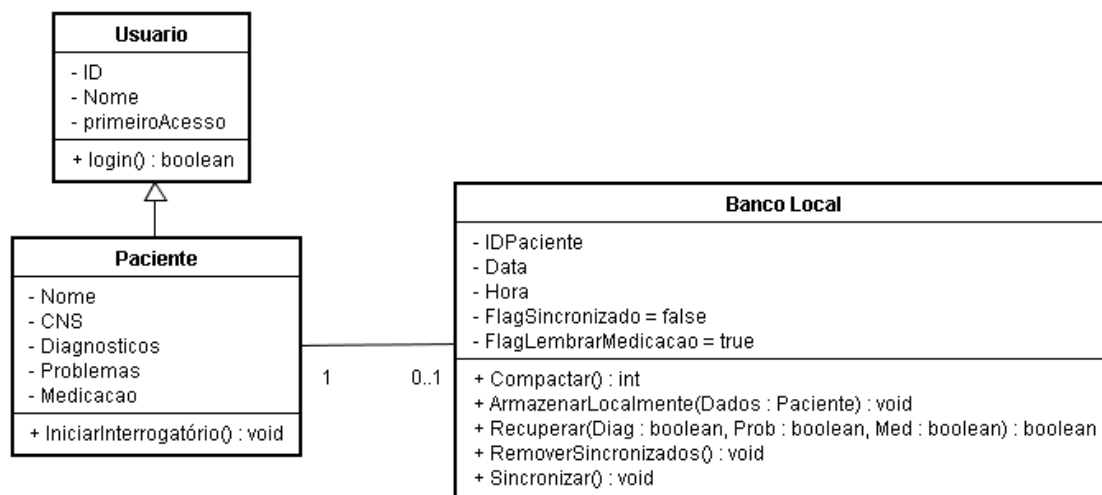
4.1.1.5 Diagrama de classes Verificando Problemas



4.1.1.6 Diagrama de classes Verificando Diagnósticos



4.1.1.7 Diagrama de classes Parar Aplicação



4.1.2 Classes

4.1.2.1 Classe Usuário

4.1.2.1.1 Descrição

Representa um Usuário genérico no GOITACÁ.

4.1.2.1.2 Atributos

Número de ordem	Nome	Descrição (tipo, valor inicial, comentários)
1	ID	Identificador único do usuário no GOITACÁ.
2	Nome	Nome do usuário.
3	primeiroAcesso	FLAG que define se é ou não o primeiro acesso do usuário ao sistema.

4.1.2.1.3 Operações

Número de ordem	Nome	Descrição (assinatura, comentários)
1	Login	Inicia um processo de autenticação com o GSWEB de maneira silenciosa.

4.1.2.2 Classe Paciente

4.1.2.2.1 Descrição

Representa um Paciente no GOITACÁ.

4.1.2.2.2 Atributos

Número de ordem	Nome	Descrição (tipo, valor inicial, comentários)
1	CNS	Cadastro Nacional de Saúde. Número de identificação único do Paciente no Sistema de Saúde Público Federal.
2	Diagnosticos	Conjunto de texto contendo os diagnósticos do paciente.
3	Problemas	Conjunto de texto contendo os problemas do paciente.
4	Medicação	Conjunto de texto contendo as Medicações ativas do paciente.

4.1.2.2.3 Operações

Número de ordem	Nome	Descrição (assinatura, comentários)
1	IniciarInterrogatorio	Inicia o Interrogatório Sintomatológico.

4.1.2.3 Classe ProfissionalSaude**4.1.2.3.1 Descrição**

Representa um Profissional de Saúde no GOITACÁ.

4.1.2.3.2 Atributos

Não definido.

4.1.2.3.3 Operações

Não definido.

4.1.2.4 Classe Médico**4.1.2.4.1 Descrição**

Representa um Médico no GOITACÁ.

4.1.2.4.2 Atributos

Não definido.

4.1.2.4.3 Operações

Não definido.

4.1.2.5 Classe Enfermeira**4.1.2.5.1 Descrição**

Representa um(a) Enfermeiro(a) no GOITACÁ.

4.1.2.5.2 Atributos

Não definido.

4.1.2.5.3 Operações

Não definido.

4.1.2.6 Classe Fisioterapeuta**4.1.2.6.1 Descrição**

Representa um Fisioterapeuta no GOITACÁ.

4.1.2.6.2 Atributos

Não definido.

4.1.2.6.3 Operações

Não definido.

4.1.2.7 Classe AuxEnfermagemTecnico

4.1.2.7.1 Descrição

Representa um Técnico Auxiliar de Enfermagem no GOITACÁ.
--

4.1.2.7.2 Atributos

Não definido.

4.1.2.7.3 Operações

Não definido.

4.1.2.8 Classe Fonoaudiólogo**4.1.2.8.1 Descrição**

Representa um Fonoaudiólogo no GOITACÁ.

4.1.2.8.2 Atributos

Não definido.

4.1.2.8.3 Operações

Não definido.

4.1.2.9 Classe Nutricionista**4.1.2.9.1 Descrição**

Representa um Nutricionista no GOITACÁ.

4.1.2.9.2 Atributos

Não definido.

4.1.2.9.3 Operações

Não definido.

4.1.2.10 Classe Psicólogo**4.1.2.10.1 Descrição**

Representa um Psicólogo no GOITACÁ.

4.1.2.10.2 Atributos

Não definido.

4.1.2.10.3 Operações

Não definido.

4.1.2.11 Classe Conectar**4.1.2.11.1 Descrição**

Representa e efetua uma conexão do GOITACÁ com o GSWEB.

4.1.2.11.2 Atributos

Número de ordem	Nome	Descrição (tipo, valor inicial, comentários)
------------------------	-------------	---

1	Endereço	Endereço seguindo o padrão HTTP para conexão com a porta de autenticação do GSWEB.
---	----------	--

4.1.2.11.3 Operações

Número de ordem	Nome	Descrição (assinatura, comentários)
1	Conectar	Inicia o processo de conexão com o servidor remoto.

4.1.2.12 Classe Sincronizar

4.1.2.12.1 Descrição

Representa uma tentativa de sincronizar o Banco de Dados do GOITACÁ com o GSWEB.

4.1.2.12.2 Atributos

Número de ordem	Nome	Descrição (tipo, valor inicial, comentários)
1	DataUltimoSincronismo	Data representando o último sincronismo com sucesso.

4.1.2.12.3 Operações

Número de ordem	Nome	Descrição (assinatura, comentários)
1	Sincronizar	Transfere os dados do GOITACÁ para o GSWEB.

4.1.2.13 Classe Enviar

4.1.2.13.1 Descrição

Representa uma subclasse da classe sincronizar, sendo usada para mostrar a origem (GOITACÁ) e o destino (GSWEB) dos dados sincronizados..

4.1.2.13.2 Atributos

Herda da superclasse.

4.1.2.13.3 Operações

Herda da superclasse.

4.1.2.14 Classe Receber

4.1.2.14.1 Descrição

Representa uma subclasse da classe sincronizar, sendo usada para mostrar a origem (GSWEB) e o destino (GOITACÁ) dos dados sincronizados..

4.1.2.14.2 Atributos

Herda da superclasse.

4.1.2.14.3 Operações

Herda da superclasse.

4.1.2.15 Classe BancoLocal

4.1.2.15.1 Descrição

Representa um Banco de Dados local e de armazenamento permanente.

4.1.2.15.2 Atributos

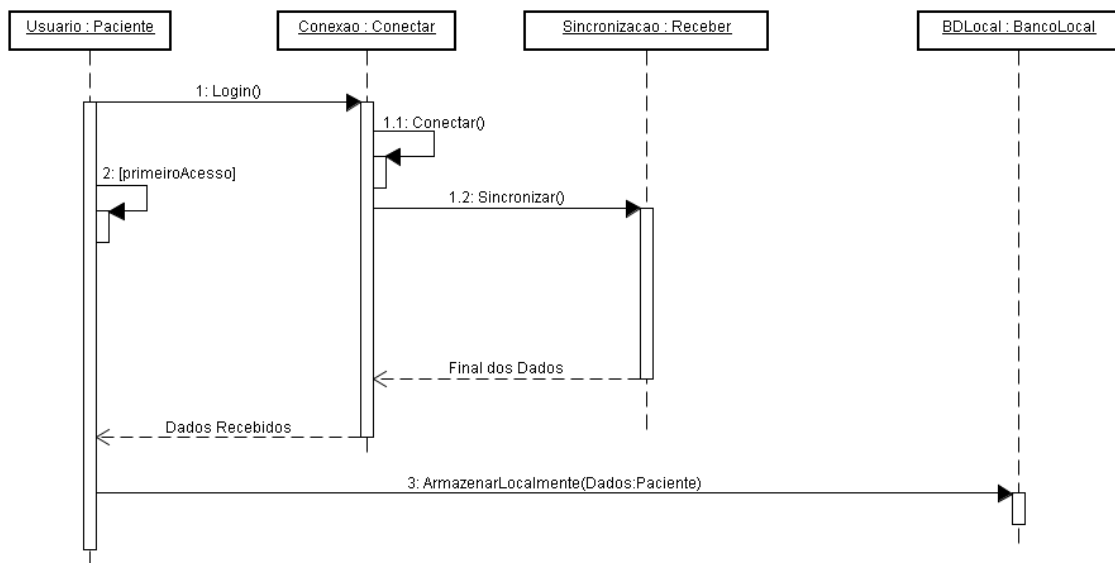
Número de ordem	Nome	Descrição (tipo, valor inicial, comentários)
1	IDPaciente	Identificador do paciente no sistema GOITACÁ.
2	Data	Data no formato dd/mm/aaaa do armazenamento do registro no banco.
3	Hora	Hora no formato h : m do armazenamento do registro no banco.
4	FlagSincronizado	Indica se o registro correspondente já foi sincronizado com o GSWEB.
5	FlagLembrarMedicacao	Indica se o lembrete de medicações ativas deve ser exibido de tempos em tempos.

4.1.2.15.3 Operações

Número de ordem	Nome	Descrição (assinatura, comentários)
1	Compactar	Compacta os dados antes de serem armazenados ou transferidos.
2	ArmazenarLocalmente	Armazena os dados do paciente localmente no dispositivo móvel.
3	Recuperar	Recuperar(boolean, boolean, boolean). Recupera a informação sinalizada pelo parâmetro correspondente, por exemplo, obj.Recuperar(null,true,null) irá recuperar somente os problemas do paciente, onde temos: (Diagnósticos, Problemas, Medicacoes).
4	RemoverSincronizados	Remove do Banco de Dados os registros já sincronizados.
5	Sincronizar	

4.1.3 Realizações dos casos de uso

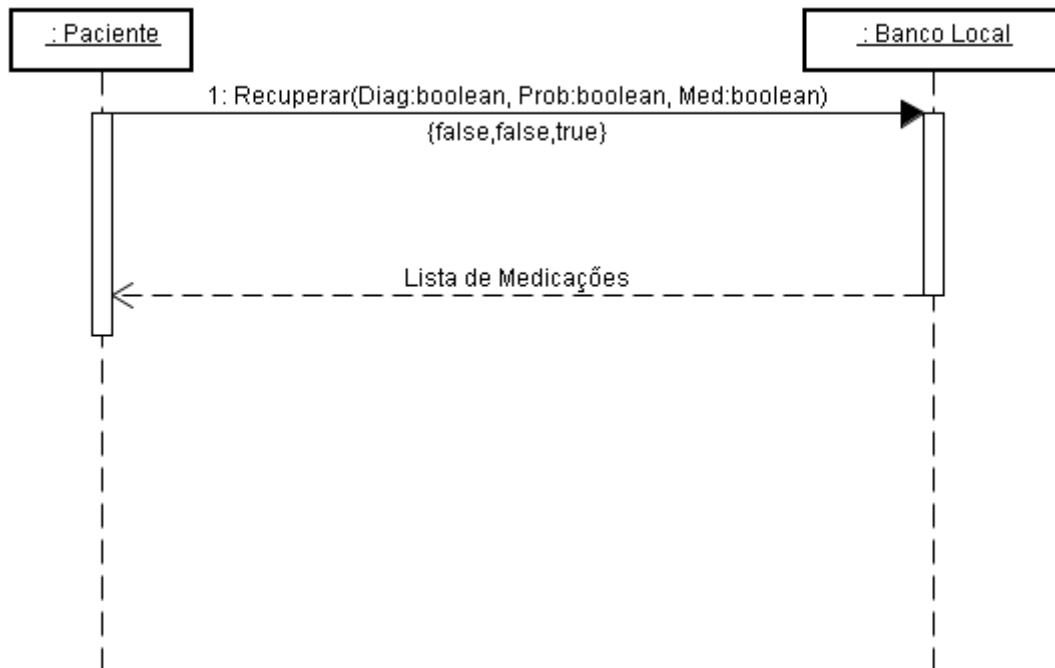
4.1.3.1 Realizações do caso de uso Primeiro Acesso



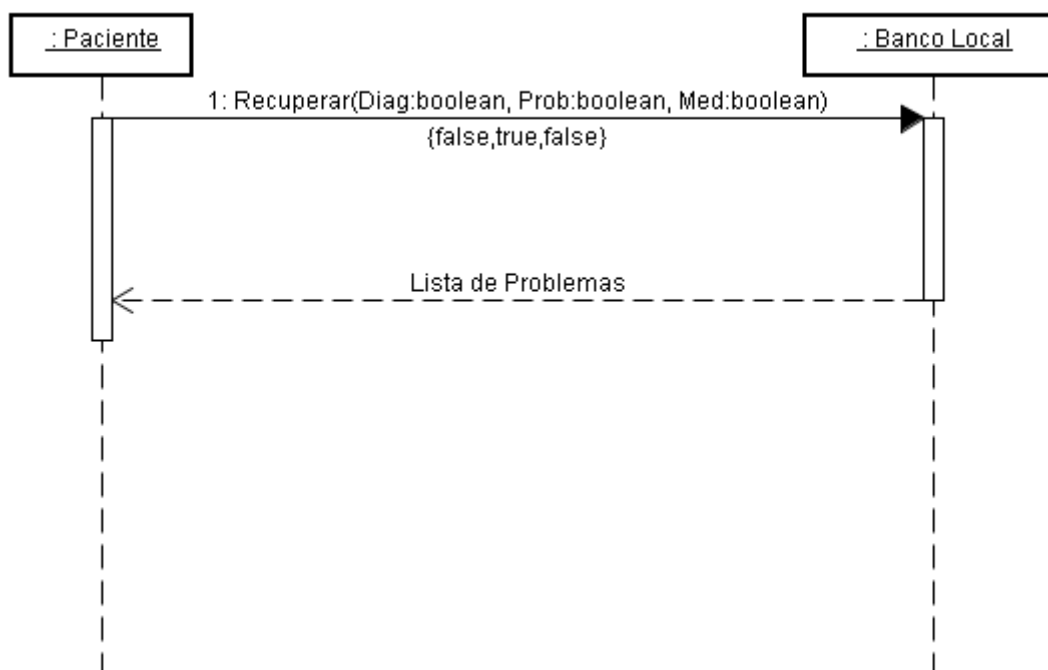
4.1.3.2 Realizações do caso de uso Executando Interrogatórios

Não aplicável.

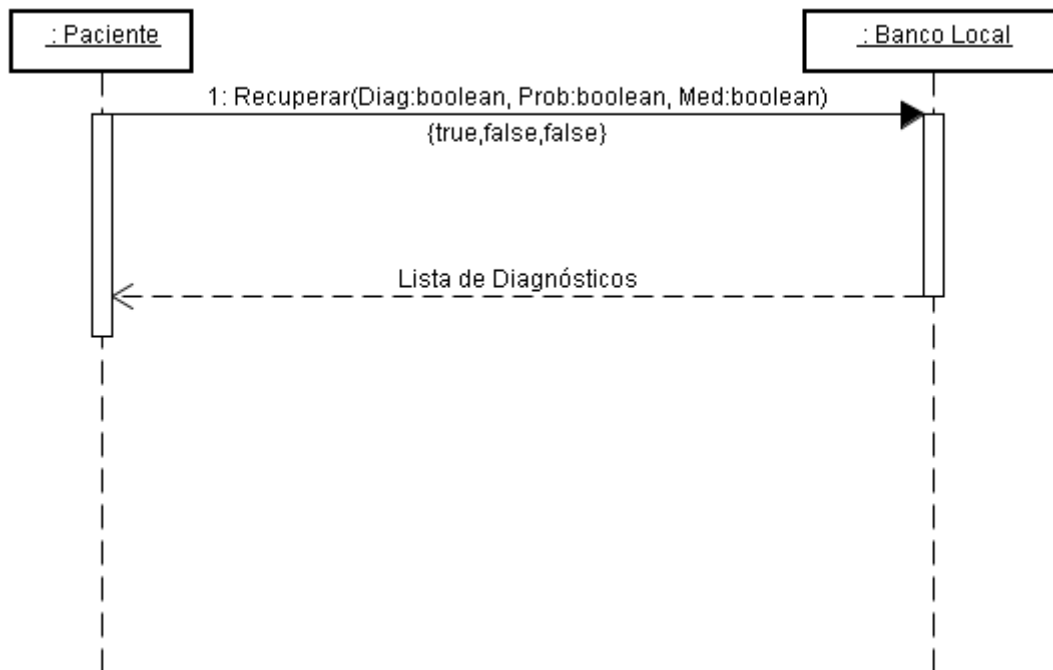
4.1.3.3 Realizações do caso de uso Verificando Medicações



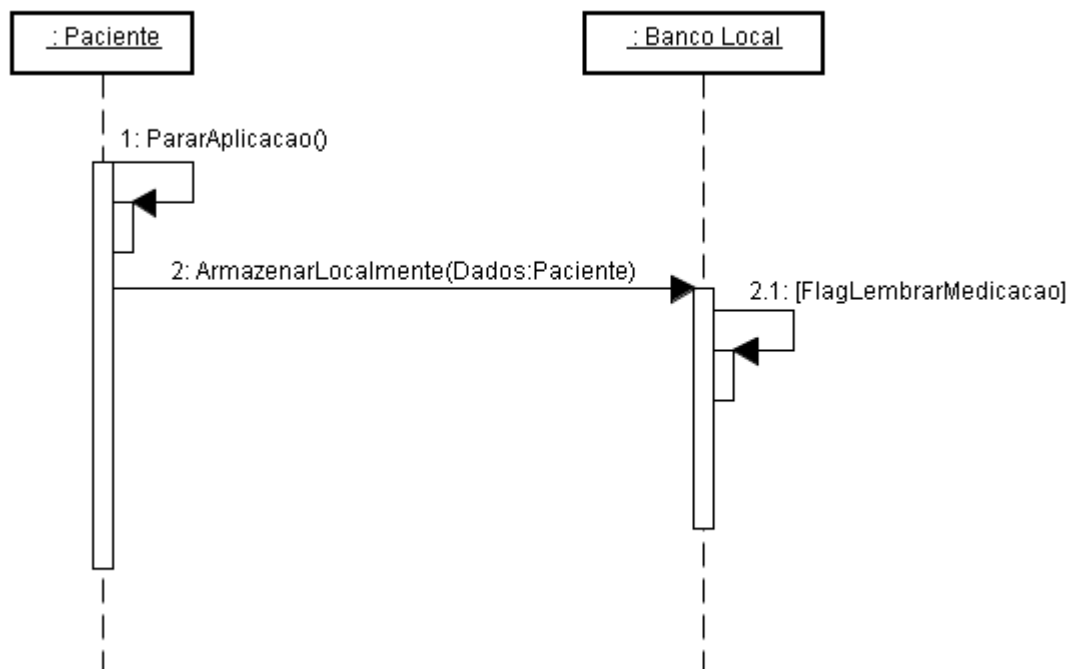
4.1.3.4 Realizações do caso de uso Verificando Problemas



4.1.3.5 Realizações do caso de uso Verificando Diagnósticos



4.1.3.6 Realizações do caso de uso Parar Aplicação



4.2 Cadastro dos requisitos

Número	Nome do requisito	Tipo	Importância	Complexidade	Estabilidade
1	Tela de Primeiro Acesso	Interface	Essencial	Baixa	Média
2	Tela de Confirmação dos Dados do Usuário.	Interface	Essencial	Baixa	Média
3	Tela do Menu Principal	Interface	Essencial	Baixa	Média
4	Tela do Interrogatório Sintomas e Sinais	Interface	Essencial	Média	Média
5	Tela do Interrogatório Sintomas por Doenças	Interface	Essencial	Média	Média
6	Tela Informativa de Medicamentos	Interface	Essencial	Baixa	Média
7	Tela de Diagnósticos	Interface	Essencial	Baixa	Alta
8	Tela de Problemas	Interface	Essencial	Baixa	Baixa
9	Tela de Sincronização	Interface	Essencial	Média	Média
10	Tela de Parar Aplicação	Interface	Essencial	Média	Média
11	Checar Dados do Usuário	Caso de uso	Essencial	Média	Média
12	Interrogatório Sintomas e Sinais	Caso de uso	Essencial	Média	Média
13	Interrogatório por Doenças	Caso de uso	Essencial	Média	Média
14	Lembrar Medicamentos	Caso de uso	Essencial	Média	Média
15	Mostrar Últimos Diagnósticos	Caso de uso	Essencial	Média	Média
16	Mostra Últimos Problemas	Caso de uso	Essencial	Média	Média
17	Sincronizar Dados	Caso de uso	Essencial	Média	Média
18	Parar Aplicação	Caso de uso	Essencial	Média	Média
19	Acessar Banco Local	Caso de uso	Desejável	Média	Média
20	Acessar Item do Menu Principal Inicializar Aplicação	Caso de uso	Essencial	Baixa	Média
21	Acessar Tela de Problemas	Caso de uso	Desejável	Média	Média
22	Acessar Tela Parar Aplicação	Caso de uso	Essencial	Baixa	Baixa

23	Armazenar Localmente	Caso de uso	Essencial	Baixa	Média
24	Avaliar Sintomas	Caso de uso	Essencial	Média	Baixa
25	Cadastrar Medico	Caso de uso	Essencial	Média	Média
26	Cadastrar Paciente	Caso de uso	Opcional	Baixa	Baixa
27	Cadastrar Usuario	Caso de uso	Essencial	Baixa	Média
28	Coletar Sintomas	Caso de uso	Desejável	Média	Baixa
29	Confirmar Dados	Caso de uso	Essencial	Alta	Média
30	Criar Tabela de Hash	Caso de uso	Desejável	Alta	Média
31	Efetuar Login	Caso de uso	Essencial	Média	Média
32	Enviar Dados	Caso de uso	Essencial	Média	Média
33	Interrogatório Sintomas e Sinais	Caso de uso	Essencial	Média	Média
34	Interrogatório Sintomas por Doenças	Caso de uso	Essencial	Média	Média
35	Realizar Interrogatório	Caso de uso	Essencial	Média	Média
36	Receber Dados	Caso de uso	Essencial	Média	Média
37	Sincronizar com GSWEB	Caso de uso	Essencial	Média	Média
38	Validar Usuario	Caso de uso	Essencial	Média	Média
39	Visualizar Dados	Caso de uso	Essencial	Média	Média
40	AuxEnfermagemTecnico	Classe	Essencial	Baixa	Média
41	BancoLocal	Classe	Essencial	Alta	Alta
42	Conectar	Classe	Essencial	Média	Alta
43	Enfermeira	Classe	Essencial	Baixa	Alta
44	Enviar	Classe	Essencial	Média	Alta
45	Fisioterapeuta	Classe	Essencial	Baixa	Alta
46	Fonoaudiologo	Classe	Essencial	Baixa	Alta
47	Medico	Classe	Essencial	Baixa	Alta
48	Nutricionista	Classe	Essencial	Baixa	Alta
49	Paciente	Classe	Essencial	Baixa	Alta
50	ProfissionalSaude	Classe	Essencial	Baixa	Alta
51	Psicologo	Classe	Essencial	Baixa	Alta

52	Receber	Classe	Essencial	Média	Alta
53	Sincronizar	Classe	Essencial	Média	Alta
54	Usuario	Classe	Essencial	Baixa	Alta

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)