

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

André Martins de Oliveira
Simone Monteiro Miranda

**Uma Ferramenta de Auxílio no Diagnóstico de
Hipertensão Arterial**

Niterói
2010

André Martins de Oliveira
Simone Monteiro Miranda

Uma Ferramenta de Auxílio no Diagnóstico de Hipertensão Arterial

**Monografia apresentada ao Departamento
de Ciência da Computação da Universidade
Federal Fluminense como parte dos
requisitos para obtenção do Grau de
Bacharel em Ciência da Computação.**

Orientadora: Viviane Torres da Silva

Niterói

2010

André Martins de Oliveira
Simone Monteiro Miranda

Uma Ferramenta de Auxílio no Diagnóstico de Hipertensão Arterial

**Monografia apresentada ao Departamento
de Ciência da Computação da Universidade
Federal Fluminense como parte dos
requisitos para obtenção do Grau de
Bacharel em Ciência da Computação.**

Aprovado em Dezembro de 2010

BANCA EXAMINADORA

Prof^a. Viviane Torres da Silva, D.Sc.
Orientadora
UFF

Prof. Leonardo Gresta Paulino Murta, D.Sc.
UFF

Prof^a. Teresa Cristina de Aguiar, D.Sc.
UFF

Niterói
2010

DEDICATÓRIA

Dedicamos este trabalho a todas as pessoas que poderão se beneficiar do sistema aqui apresentado, lhes proporcionando uma melhor qualidade de vida. E aos especialistas, que este sistema possa ser um aliado no exercício de sua profissão.

AGRADECIMENTO

Primeiramente, gostaria de agradecer a minha orientadora Viviane Torres pela sempre presente solicitude e atenção. Sua contribuição se fez presente desde as sugestões para novas idéias à serem integradas em nosso projeto até a revisão de cada capítulo, sendo fundamental para a realização deste projeto.

Não poderia deixar de agradecer também a diversos professores da Universidade Federal Fluminense, pela inspiração e apoio em diferentes momentos desses últimos cinco anos.

Também sinto que devo uma homenagem aos meus pais, Maria Antônia e Arivaldo, pelo suporte incondicional e constante confiança. Foi somente graças a sua dedicação exemplar que a realização do presente trabalho se tornou possível.

Por fim, gostaria de agradecer à minha namorada, Karine. O fato de termos caminhado juntos ao longo de toda a faculdade me garantiu um exemplo de dedicação aos estudos a ser seguido. Seu companheirismo e paciência foram imprescindíveis para que eu chegasse até aqui.

Obrigado

André Martins de Oliveira

AGRADECIMENTO

Agradeço, primeiramente, a Deus que me permitiu chegar até este momento. Sabia que a caminhada até aqui seria difícil, mas com a força e determinação que Ele derramou sobre mim, pude seguir adiante com o pensamento firme em meu objetivo.

Agradeço aos meus pais, Germano e Olga pelo apoio em todos os momentos. Agradeço ainda ao meu irmão Sidney, por tornar mais alegre e leve a caminhada até aqui. Obrigada por acreditarem em meu potencial. Ainda peço desculpas e agradeço a compreensão por entenderem a minha ausência em suas vidas, unicamente para me dedicar a este trabalho.

Agradeço a minha Orientadora Viviane pelos ensinamentos, pelas dicas e pelo longo tempo de trabalho e companheirismo.

Agradeço ao meu colega André por aceitar seguir comigo neste desafio, compartilhando tantas horas de esforço e privando-se de finais de semanas e feriados para concluir este trabalho.

Agradeço ainda ao amigo Carlos Heitor, por ter sido uma das muitas pessoas que me fizeram crer que eu seria capaz de chegar até aqui no momento em que pensei em desistir de tudo.

E a todos os amigos que conquistei ao longo deste curso, meu sincero obrigado e espero vê-los novamente muito em breve nos caminhos da vida.

A todos vocês e a toda a minha família, em especial minhas avós Raimunda e Matilde, que mesmo distante sei que torcem por mim, dedico este trabalho.

Obrigada por Tudo

Simone Monteiro Miranda

RESUMO

A doença de hipertensão arterial consiste na elevação abrupta da pressão arterial. Os pacientes portadores dessa doença devem ter constante monitoramento de suas pressões a fim de mantê-las em níveis aceitáveis e consequentemente não trazendo qualquer risco para o mesmo. Este trabalho apresenta uma ferramenta para auxiliar especialistas em cardiologia a diagnosticar a hipertensão arterial, considerando, além das medições de Pressão Arterial em Consultório (que aqui chamaremos, abreviadamente de PAC) e da Monitorização Ambulatorial de Pressão Arterial (MAPA) e outros fatores que influenciam na variação da pressão arterial, como diabetes e tabagismo, e atividades desenvolvidas pelo paciente durante a medição da MAPA. Com base nas informações recebidas, a ferramenta informa ao usuário a classificação da pressão arterial e o risco individual do paciente hipertenso. A ferramenta ainda poderá ser utilizada pelos médicos em seus consultórios funcionando como um gerenciador de cadastros dos pacientes.

Palavras Chaves:

Pressão Arterial, Hipertensão Arterial, MAPA, Diagnóstico, Monitoramento

ABSTRACT

The disease of hypertension is the abrupt rise in blood pressure. Patients with this disease must be constantly monitoring their blood pressure to maintain them at acceptable levels and therefore not bringing any harm to it. This paper presents a tool to assist specialists in cardiology to diagnose hypertension, whereas, than measurements of Blood Pressure at Clinic (which we call here abbreviated CAP) and Ambulatory Blood Pressure Monitoring (ABPM) and other factors that influence variation in blood pressure, such as diabetes and obesity, and activities developed by the patient during measurement of MAP. Based on information received, the tool informs the user the classification of blood pressure and individual risk of hypertensive patients. The tool can still be used by physicians in their offices working as manager registries of patients.

Keywords:

Blood Pressure, Hypertension, ABPM, Diagnosis, Monitor

SUMÁRIO

CAPÍTULO 1- INTRODUÇÃO	12
CAPÍTULO 2- HOME CARE.....	14
2.1 Serviço de assistência domiciliar à saúde	14
2.2 SADST: Sistema de Assistência Domiciliar à Saúde Telemonitorada.....	16
2.3 SCIADS: Sistema Computacional Inteligente de Assistência Domiciliar à Saúde.....	18
2.3.1 PressureTool	19
CAPÍTULO 3 – FERRAMENTA JESS	22
3.1 Regras de Produção e Sistemas Especialistas	22
3.2. Definição do Jess.....	25
3.3. Linguagem Jess e Comunicação com Java.....	28
3.3.1. Modelos	28
3.3.2. Fatos	29
3.3.3. Regras	30
3.3.4 Funções.....	32
3.3.5 Comunicação com Java	32
3.4. Exemplo em Jess	33
CAPÍTULO 4 - PRESSURETOOL: UMA VISÃO DETALHADA DO SISTEMA	39
4.1. Escopo do Sistema	39
4.2 Detalhes da Implementação	49
4.3 As regras de inferência do PressureTool.....	50
4.4 Detalhes do funcionamento.....	54

CAPÍTULO 5 - TRABALHOS RELACIONADOS.....	61
CAPÍTULO 6 - CONCLUSÃO	63
REFERÊNCIAS BIBLIOGRAFICAS	64
APÊNDICE I.....	66

LISTA DE FIGURAS

Figura 1. Anúncio de empresa de serviço de Home Care, em Niterói	15
Figura 2. Organização e atores do SADST	17
Figura 3. Ambiente Domiciliar.....	19
Figura 4. PressureTool acoplado ao SCIADS	20
Figura 5. Arquitetura de um sistema de produção.....	23
Figura 6. Jess acoplado à arquitetura do sistema de produção	27
Figura 7. Tela de confirmação da instalação do Jess.....	28
Figura 8. (a) Sintaxe que define um template (b) Exemplo de declaração de template	29
Figura 9. Exemplo de declaração de fatos.....	30
Figura 10. (a) Sintaxe de uma regra (b) Exemplo de declaração de regra	31
Figura 11. Exemplo de definição de função	32
Figura 12. (a) Biblioteca do Jess (b) Trecho de código de comunicação entre Jess e Java.....	33
Figura 13. (a) Trecho de inicialização do programa em Jess (b) Regras executadas no programa (c) Execução do programa em Jess	38
Figura 14. Diagrama de Casos de Uso	40
Figura 15. Diagrama de Implantação	46
Figura 16. (a) Diagrama de Classes das Camadas de Negócio (b) Diagrama de Classes da Camada de Banco de Dados	47
Figura 17. Diagrama de Sequência: Cadastrar novo paciente	48
Figura 18. Diagrama de Sequência: Cadastrar nova consulta	49
Figura 19. Trecho de Relatório de Avaliação de Pressão Arterial	53
Figura 20. Tela de Login	55
Figura 21. (a) Menu de funcionalidades para o administrador (b) Tela de importação do médico	56
Figura 22. (a) Menu de paciente e opções de cadastrar, visualizar, editar e excluir cadastro do paciente (b) Tela de novo cadastro de paciente.....	57
Figura 23. (a) Menu de consulta do sistema e opções de cadastrar e exibir consulta (b) Tela de cadastro de nova consulta.....	59
Figura 24. Menu de ajuda do sistema	60
Figura 25. Menu de Sistema: Logout e Encerramento do sistema	60

LISTA DE TABELAS

Tabela 1. Lista de fatores de risco contemplados no PressureTool.....	51
Tabela 2. Classificação de pressão arterial para adultos acima de 18 anos.....	51
Tabela 3. Risco cardiovascular adicional ao paciente, segundo seus fatores de risco	51
Tabela 4. Tabela de atividades do PressureTool	53
Tabela 5. Tabela de valores de referência para gestantes.....	54

CAPÍTULO 1- INTRODUÇÃO

No ramo da medicina muitas são as contribuições da tecnologia a fim melhorar o trabalho dos médicos e o tratamento de seus pacientes. Dentre as contribuições da tecnologia a serviço da medicina, aliaram-se telecomunicações à tecnologia da informação, criando a Telemedicina que se apresenta como sendo um *“conjunto de tecnologias e aplicações que permitem a realização de ações médicas à distância”* [6].

No mercado atual, tem-se o serviço denominado Home Care, cujo objetivo é a contratação de profissionais a fim de realizar um tratamento ou mesmo acompanhamento de reabilitação dos pacientes em suas residências. A Telemedicina pretende ser *“um método que consiste na aplicação do atendimento médico nos casos em que a distância é um fator crítico entre o profissional e o paciente, sendo um recurso possível com o advento das evoluções tecnológicas”* [6] não possuindo a pretensão de ser uma nova especialidade médica, mas sim atuar de forma conjunta com todos os ramos da medicina.

O objetivo deste projeto é o desenvolvimento de um sistema baseado em regras para auxiliar um médico no diagnóstico de hipertensão arterial, considerando, além das medições de Pressão Arterial em Consultório (PAC), as medições obtidas por um exame complementar de mapeamento de pressão arterial denominado Medição Ambulatorial de Pressão Arterial (MAPA), quando houver. Esse exame é realizado pelo paciente, prescrito pelo médico, para medir as pressões arteriais durante 24 horas, resultando em um conjunto de pressões onde cada medição está associada à atividade que o paciente estava executando no momento em que a pressão fora medida. . Outras doenças crônicas, como diabetes e mesmo condições temporárias, como gravidez, também são importantes para um diagnóstico correto de hipertensão arterial e, portanto serão considerados no sistema. Para cada um dos critérios, a pressão arterial oscila de maneira diferente, não significando necessariamente um caso de hipertensão arterial.

Alguns dados descritos anteriormente são obtidos durante a consulta do paciente e outros dados, como a MAPA, serão importados de uma central a qual o sistema aqui apresentado estará em comunicação direta. A central faz parte do contexto onde o sistema foi inserido, contexto este sendo um projeto denominado SCIADS, cujo objetivo é monitorar o estado de saúde do paciente em sua residência configurando o conceito da Telemedicina.

Pretendemos com este trabalho contribuir para uma avaliação médica com maior coleta de dados que possibilitem a elaboração de um diagnóstico mais preciso na identificação do problema de hipertensão arterial. As principais contribuições deste trabalho são:

- Definição e implementação de um conjunto de regras de inferência para auxiliar no diagnóstico do nível de risco de problemas cardiovasculares. As regras foram definidas com base na documentação provida pela OMS [4];
- Definição e implementação de um conjunto de regras para auxílio no diagnóstico de hipertensão arterial;
- Implementação de uma ferramenta para auxiliar o médico no cadastro de pacientes e na execução das regras citadas acima.

Este trabalho está organizado da forma que segue abaixo:

O Capítulo 2 apresenta o contexto em que este trabalho está inserido, definindo o serviço que está presente no mercado e mostrando alguns projetos em desenvolvimento na Universidade Federal Fluminense para esse tipo de serviço.

O Capítulo 3 apresenta a ferramenta utilizada para criar as regras de avaliação do sistema, mostrando desde a sua arquitetura até a sintaxe que permitiu gerar as regras. A fim de um melhor entendimento sobre as regras utilizadas na avaliação do paciente, ao final do trabalho é apresentado um apêndice contendo a implementação das regras na linguagem utilizada.

O Capítulo 4 mostra efetivamente o sistema objeto deste trabalho. Neste capítulo foram aplicadas as técnicas de Engenharia de Software para expor todo o escopo do sistema através da análise dos requisitos funcionais e diagramas.

O Capítulo 5 compara o sistema apresentado neste trabalho a outros que atuam na mesma área, elencando vantagens e desvantagens de ambos os lados.

Por fim, o Capítulo 6 apresenta a conclusão deste trabalho e ainda mostra as propostas de continuidade do sistema.

CAPÍTULO 2- HOME CARE

Nas seções seguintes será explicado o serviço de assistência domiciliar à saúde bem como projetos que estão sendo desenvolvidos nessa área.

2.1 SERVIÇO DE ASSISTÊNCIA DOMICILIAR À SAÚDE

“A assistência domiciliar é a provisão de serviços de saúde por prestadores formais e informais com o objetivo de promover, restaurar e manter o conforto, função e saúde das pessoas num nível máximo, incluindo cuidados para uma morte digna. Serviços de assistência domiciliar podem ser classificados nas categorias de preventivos, terapêuticos, reabilitadores, acompanhamento por longo tempo e cuidados paliativos”. [4]

Historicamente, o Serviço de Assistência Domiciliar à Saúde (SADS), se confunde com a história dos antigos médicos de família. Na verdade, o SADS, em sua essência, ainda possui os mesmos objetivos, que é o de tratar as patologias dos pacientes em suas residências. Com essa prática, o paciente pode ser privado dos riscos de infecções hospitalares, bem como do desconforto e do ambiente inóspito de um hospital. O paciente é acompanhado em seu lar e entre seus familiares, formando um ambiente propício para a evolução de seu quadro clínico.

O SADS, também conhecido pelo termo Home Care, é uma das áreas que mais cresce em termos de pesquisa dentro do contexto de assistência à saúde [5], devido o crescimento do público a quem se destina tal serviço. Esse crescimento é motivado pelo aumento da incidência das doenças crônicas, como a hipertensão que cresceu de 21,5%, em 2006, para 24,4%, em 2009¹, bem como o aumento da expectativa de vida (passando de 69,9 anos em 1998 para 72,8 em 2008)², formando uma grande população de idosos. O SADS visa o tratamento desse público, mas não deve limitar-se somente a ele. O serviço ainda pode ser visto como uma alternativa de diminuição de custos das internações hospitalares de pacientes em estado estável.

¹Ministério da Saúde 2010 -

http://portal.saude.gov.br/portal/aplicacoes/noticias/default.cfm?pg=dspDetalheNoticia&id_area=124&CO_NOTICIA=11290

² IBGE 2009 - <http://www.estadao.com.br/noticias/geral,ibge-expectativa-de-vida-no-brasil-chega-aos-728-anos,474856,0.htm>

Não somente na área da pesquisa que o SADS vem crescendo, mas o mercado de empresas que prestam esse serviço também aumentou. Segundo censo publicado em 2004 pelo Núcleo Nacional de Empresas de Assistência Domiciliar (NEAD) eram 108 empresas cadastradas no Conselho Regional de Medicina (CRM)³, passando para 170 empresas cadastradas, no censo também realizado pelo mesmo NEAD e divulgado em 2006.⁴ Como mostra a Figura 1, já é possível, inclusive, encontrar anúncios de empresas oferecendo este serviço.

Com o objetivo de regulamentar este tipo de serviço, a Associação Brasileira de Health Home Care (ABRAHHCARE)⁵ foi criada. Além desse órgão, há também a Resolução CFM nº 1.668/2003⁶ criada pelo Conselho Federal de Medicina (CFM), que visa regulamentar as empresas que oferecem tal serviço e é tida como uma legislação para os serviços de Home Care.



Figura 1. Anúncio de empresa de serviço de Home Care, em Niterói

³ NEAD - <http://www.neadsaude.org.br/>

⁴ O Estadão 2006- <http://www.estadao.com.br/arquivo/vidae/2006/not20060214p69207.htm>

⁵ ABRAHHCARE – <http://www.abrahhcare.org.br>

⁶ Associação de Hospitais e Clínicas do Rio de Janeiro(AHCRJ) - <http://www.ahcrj.com.br/legislacao.htm>

O SADS é uma prática nova no Brasil e ainda é objeto de muitos estudos e projetos. A seguir, descrevemos dois sistemas desenvolvidos no Instituto de Computação da Universidade Federal Fluminense (UFF).

2.2 SADST: SISTEMA DE ASSISTÊNCIA DOMICILIAR À SAÚDE TELEMONITORADA

O SADST tem por objetivo o monitoramento de pacientes com doenças crônicas, em fase de reabilitação ou mesmo idosos, que necessitam de constante acompanhamento de suas prescrições médicas (chamado aqui de plano de cuidado) e de dados fisiológicos.

Esse monitoramento consiste no ambiente residencial sendo controlado por sensores interligados, por uma rede sem fio, a uma central instalada em uma instituição de saúde, responsável pelo paciente. Sendo assim, um canal de comunicação é formado entre o paciente em sua residência e a instituição de monitoramento. Outros mecanismos de comunicação com o especialista e com a família do paciente, também são utilizados para uma efetiva monitorização.

Um conjunto de sensores, anteriormente citados, faz a medição da pressão arterial e obtém informação sobre atividade realizada pelo paciente, armazenando-as em um dispositivo local. Esse dispositivo envia esses dados, bem como dados referentes a características físicas do ambiente (temperatura e umidade), que por sua vez foram obtidos por outros sensores, para a central de monitoramento. A partir desses dados é possível inferir a atual situação de saúde do paciente e tomar medidas pró-ativas ou emergenciais, caso se verifique uma alteração crucial no estado de saúde do mesmo. As medidas que podem ser tomadas variam entre alertar o paciente para realizar alguma ação prescrita pelo médico até informar ao médico, a família ou solicitar atendimento emergencial, dependendo da gravidade da alteração.

O nível de gravidade é definido por uma análise da pressão arterial, levando em consideração a atividade sendo realizada pelo paciente durante a medição e os demais dados que foram transmitidos. É muito importante ressaltar a importância de se ter claro qual a atividade que fora realizada pelo paciente, pois durante uma determinada atividade pode ser normal que a pressão se eleve, não significando necessariamente um caso de emergência. Sem essa informação, pode-se concluir uma situação de saúde equivocada. Para tal análise, são definidos três níveis indicativos da situação do paciente: normal, alerta ou emergência, que

ficará armazenado em um histórico do paciente, na central, podendo ser consultado pelo médico especialista e pela família do mesmo.

Como produto final desse projeto, criou-se um protótipo composto de um sensor de medição da pressão arterial que deve ser ajustado ao pulso do paciente. Esse sensor envia as medições para um dispositivo, uma espécie de PDA, que solicita que o paciente informe a atividade realizada durante a medição. Os dados são processados por um módulo denominado HHS (Home Health Station) que também os armazena localmente, com o intuito criar o histórico do paciente. Esses dados são posteriormente transferidos para a central de monitoramento, tornando esses dados acessíveis ao especialista responsável bem como à família do paciente. Desse modo podem-se identificar quatro elementos básicos do HHS: **obtenção, análise, armazenamento, distribuição** de dados.

Todo este processo tem como base o plano de cuidado, residente na central de monitoramento, prescrito pelo médico em função dos dados do paciente. Um remédio a ser tomado em um determinado horário ou uma pressão a ser medida, são exemplos que podem ter sido descritos pelo especialista e que o sistema deve solicitar que paciente o faça. Quando o paciente cumprir com a sua prescrição, o sistema deve ser atualizado indicando que já fora cumprido. O fato dos planos de cuidados estarem armazenados na central de monitoramento permite que o especialista faça alterações de forma dinâmica e quando necessário, dependendo dos dados que são coletados pelo sistema.

A figura 2 extraída de [5] ilustra como está organizado o SADST, quem são os atores envolvidos em todo processo, bem como o fluxo dos dados. É possível também identificar as quatro partes que compõe o HHS, que já foram anteriormente descritas.

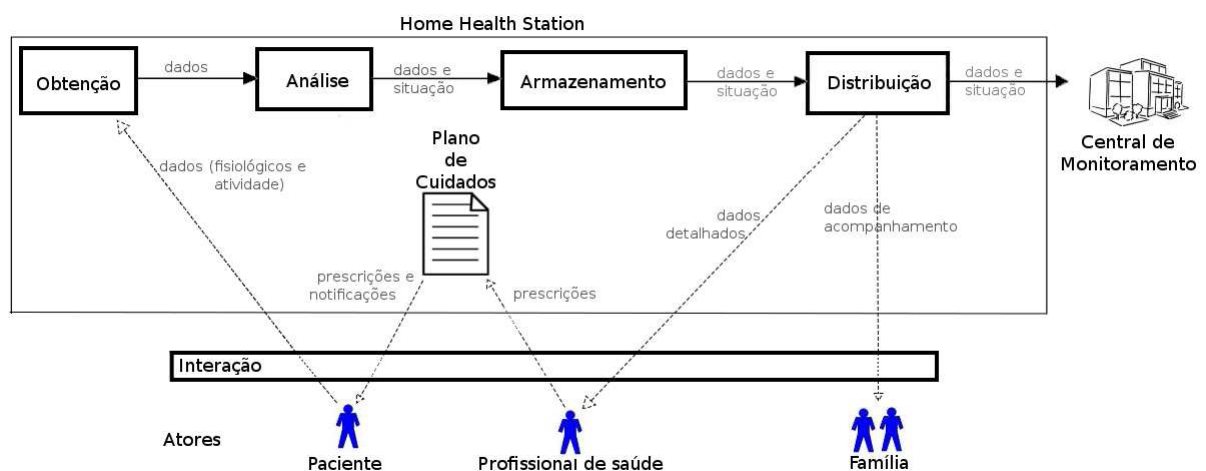


Figura 2. Organização e atores do SADST

2.3 SCIADS: SISTEMA COMPUTACIONAL INTELIGENTE DE ASSISTÊNCIA DOMICILIAR À SAÚDE

Na seção anterior, foi explicada a primeira fase do projeto na área de Home Care, o SADTS. Fase esta já concluída, e que agora entra numa segunda fase, denominada Sistema Computacional Inteligente de Assistência Domiciliar à Saúde (SCIADS).

A fase anterior tinha como foco a coleta dos dados, identificação do estado de saúde do paciente em questão, criação de histórico local dos dados coletados bem como sua distribuição para as entidades de monitoramento responsáveis.

Como objetivo para esta fase, além de melhorias na obtenção dos dados no SADTS, é o tratamento dos dados coletados do paciente. Isto é, o foco do SCIADS é o modo como esses dados serão distribuídos, visualizados e tratados pela entidade responsável. Já a partir desta fase, serão renomeados alguns elementos que compõem o sistema e que serão utilizados ao longo deste trabalho, sendo mostrados na figura 3 extraída de [7]. Será chamada de Central de Saúde Residencial (CSR) a parte do sistema que ficará na residência do paciente e, Central de Supervisão Médica (CSM) a outra parte do sistema responsável pelo monitoramento do paciente, estando de responsabilidade de um especialista.

Algumas características que serão introduzidas nesta fase, incluem dentre outras, a identificação e filtragem de estados relevantes a um determinado paciente a partir de regras adaptáveis ao mesmo. Outro aspecto será a reconfiguração dinâmica no plano de saúde. Por exemplo, se o sistema deve avisar ao paciente para tomar um remédio, mas ele está em repouso, o sistema deve ser capaz de identificar a impossibilidade imediata de cumprir com tal prescrição e alertá-lo assim que ele despertar, caso o horário do remédio possa ser adiado. Fica intrínseco que o sistema deverá também ser capaz de identificar qual a atividade está sendo realizada pelo paciente.

Um novo Módulo de Raciocínio, também chamado de Módulo de Decisão, será introduzido no CSR, novamente mostrado na figura 3 [7]. O intuito é formar esse módulo a partir de Agentes de Software, que permitem o auxílio ao especialista inferindo diagnósticos a partir da coleta dos dados. Agentes são entidades autônomas, interativas com outros agentes e orientadas a objetivos, isto é, possuem um plano de ações a serem desenvolvidos para realização dos objetivos ao qual foram designados. Além disso, os agentes possuem grande capacidade de aprendizagem e adaptabilidade ao ambiente em que estão inseridos,

convergindo com a necessidade de um monitoramento customizado a diferentes pacientes, buscada nesta fase do projeto.

De modo geral, em ambas as fases, o projeto possui grandes desafios, o maior deles é o tratamento de vidas humanas. “*Sistemas de assistência domiciliar não podem falhar. Uma falha sistêmica pode causar a perda de uma vida*” [7]. É contraditório criar um sistema que não possa ser confiável, principalmente em se tratando de vidas. Para isso, também há um foco muito grande na garantia da segurança e da validação da informação, o que está fora do escopo deste trabalho.

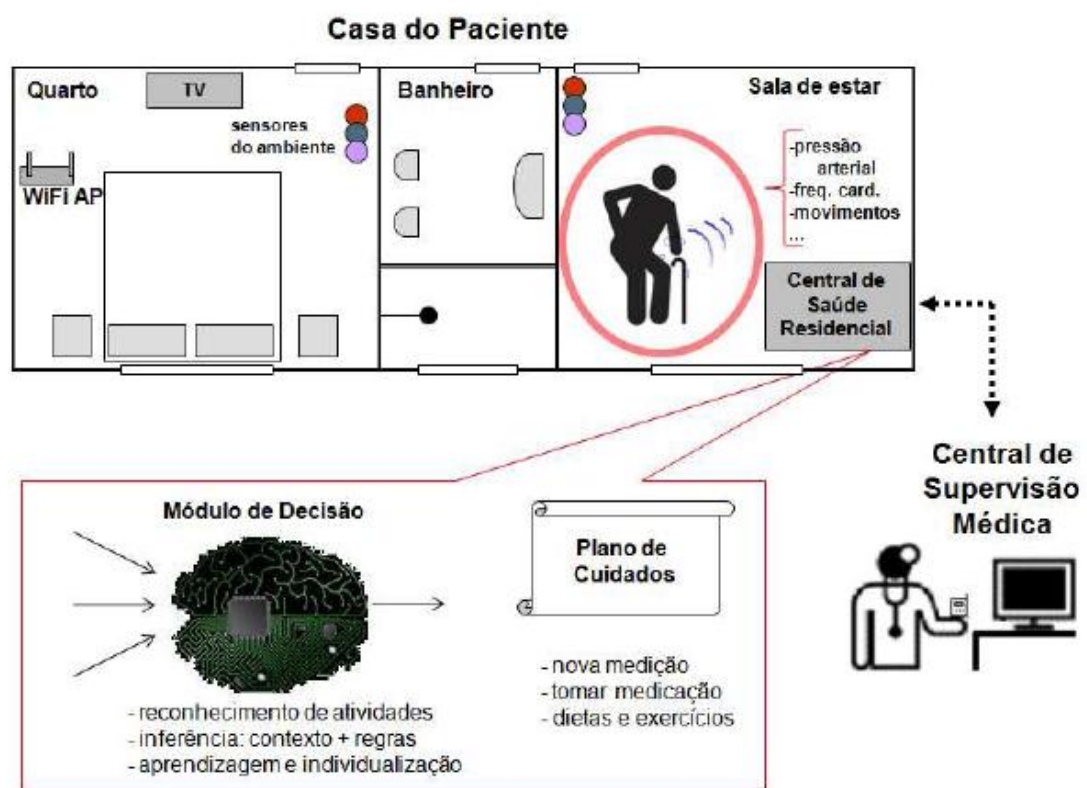


Figura 3. Ambiente Domiciliar

2.3.1 *PressureTool*

Este trabalho tem como objetivo apresentar o protótipo de uma ferramenta, aqui denominada *PressureTool*, cuja função é o auxílio ao especialista em cardiologia realizando inferência de diagnósticos de doenças cardiovasculares. Essa inferência incidirá sob pacientes com diagnóstico em fase de comprovação dessas doenças, em especial a hipertensão arterial, que é o foco deste trabalho. Desta forma, os pacientes envolvidos compõem o público alvo para o sistema em questão.

Como mostrado na figura 4, adaptada de [7], o *PressureTool* está inserido no ambiente descrito da seção anterior, estando acoplado à infraestrutura já existente no SCIADS, pois ele corresponde ao CSM (Central de Supervisão Médica), isto é, o sistema para quem os dados da central estarão disponíveis para futura análise clínica.

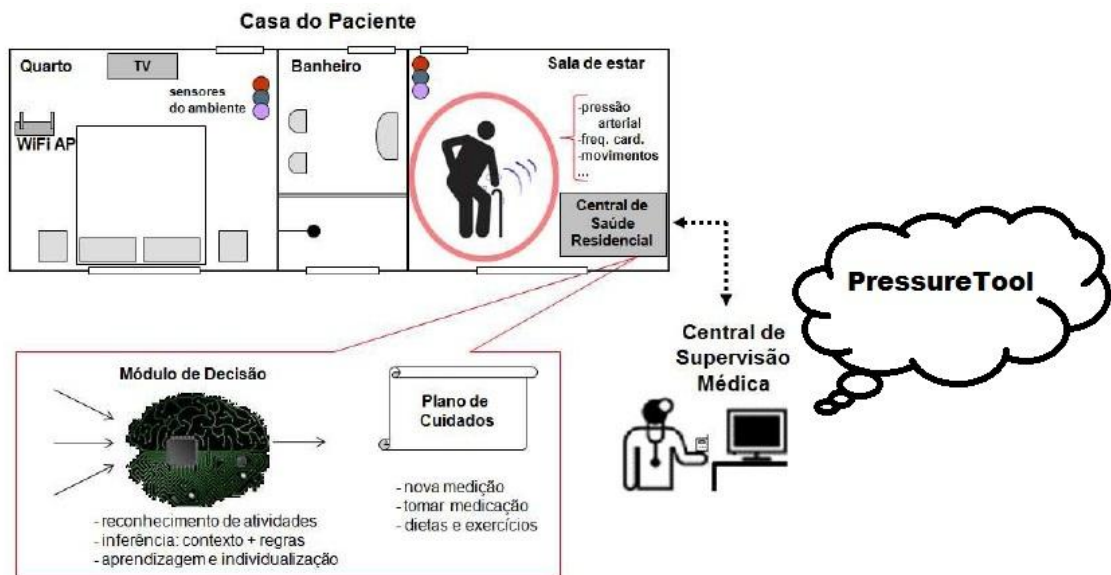


Figura 4. PressureTool acoplado ao SCIADS

Em linhas gerais, a dinâmica do sistema começa quando o paciente, ao se consultar pela primeira vez por um médico que utiliza o *PressureTool*, tem seus dados pessoais como nome, endereço, telefone, data de nascimento e idade, armazenados no sistema juntamente com os dados relativos a consulta, como o motivo da consulta, prescrições médicas e medições da pressão. Caso o paciente já esteja cadastrado no sistema, o médico poderá visualizar as consultas anteriores antes de prosseguir com o cadastro da nova consulta.⁷ Em particular, o médico poderá solicitar outros exames complementares de forma a comprovar as medições da pressão feitas no consultório. Geralmente, é prescrito o uso da Monitorização Ambulatorial de Pressão Arterial (MAPA), que é o acompanhamento durante 24hr da variação da pressão arterial como exame complementar e comprobatório da alteração anormal da pressão. Durante o uso da MAPA, são coletadas, periodicamente, as pressões bem como as atividades realizadas durante as medições. É importante ressaltar que a informação de qual atividade fora realizada durante a medição propicia a avaliação que será feita futuramente. Por exemplo, se um paciente teve uma medição de pressão considerada alta no consultório e

⁷ Outros detalhes de implementação do sistema serão apresentados no capítulo 4 deste trabalho.

devido a isso, foi indicado o uso da MAPA, e mesmo durante o exame as medições da pressões também se mostraram elevadas, é necessário, atentar para a atividade realizada durante a medição, pois a pressão pode se elevar de forma considera normal não necessariamente sendo um caso de hipertensão arterial. A atividade realizada passa a ser um critério de avaliação, para eliminar alarmes falsos.

Todas as medições são enviadas para a central, e através de um mecanismo de importação de dados do sistema, o médico poderá ver um relatório dos dados medidos. O *PressureTool* ainda apresenta uma funcionalidade de pré-avaliação sobre o estado de saúde do paciente durante a realização da MAPA, baseando-se nos dados coletados. É a partir dessa funcionalidade que se constitui o principal objetivo do sistema, descrito no início desta seção. Para execução da pré-avaliação são necessários alguns dados de entrada como atividade executada no momento da medição da MAPA, idade, algumas características físicas e outros fatores relevantes que possam de alguma forma influenciar na elevação da pressão.

O resultado dessa pré-avaliação será visualizado juntamente com as medições importadas e todos os dados armazenados no cadastro do paciente junto ao sistema, formando assim seu histórico clínico e servindo de base para a avaliação final que será dada pelo médico que atende ao paciente. O *PressureTool* não possui pretensão de substituir os conhecimentos do especialista, seu objetivo é ser mais um aliado do mesmo na geração de diagnósticos corretos de pacientes hipertensos.

CAPÍTULO 3 – FERRAMENTA JESS

Neste capítulo será apresentada a ferramenta *Jess (Java Expert System Shell)* [1] utilizada na construção das regras de inferência que fazem a pré-avaliação do estado de saúde do paciente, baseado nas medições da pressão provenientes da Monitorização Ambulatorial de Pressão arterial (MAPA), quando o paciente o tiver realizado, e baseada também na medida de pressão realizada pelo médico no consultório.

3.1 REGRAS DE PRODUÇÃO E SISTEMAS ESPECIALISTAS

A definição do Jess tem como base o conceito de Regras de Produção de Conhecimento, termo conhecido na Inteligência Artificial. Elas são formadas pelos pares seguintes:

Se (*condição ou premissa*),

Então (*ação ou conclusão*)

Onde *condição* é conhecida também por LHS⁸ (parte esquerda da regra) e a *ação* por RHS⁹ (parte direita regra). Por exemplo, dada uma condição, através das regras, pode-se inferir a ação a ser tomada:

Se (está calor) e (não está chovendo),

Então (vou à praia).

Dentro da Inteligência Artificial, essas regras modelam o encadeamento do pensamento humano. As regras, por sua vez, formam os Sistemas de Produção, nomenclatura genérica para os sistemas baseados em regras. Dentre esses sistemas, em particular neste trabalho, será abordado os Sistemas Especialistas, que segundo [8] os define como sendo:

“(...) programas que possibilitem a utilização dos conhecimentos dos especialistas através de uma máquina que permita o armazenamento e o seqüenciamento de informações e a auto-aprendizagem (...)”, possuindo, ainda segundo [8],

⁸ Do inglês: *Left Handle Side*

capacidade para “(...) *processar informações não numéricas* (...)” e “(...) *de apresentar conclusões sobre um determinado tema, desde que devidamente orientado e ‘alimentado’*”.

Os conceitos de “*orientado*” e “*alimentado*” apresentados na definição anterior, partem da arquitetura dos sistemas de produção, como mostrado na figura 5 extraída de [9] e que se aplica a todos outros sistemas baseados em regras, incluindo os especialistas. O primeiro deles referencia a capacidade de aprendizagem do sistema representada pela memória de trabalho e pela base regras. A memória de trabalho armazena *fatos* já conhecidos pelo sistema enquanto a base de regras possui as regras para inferência de novos fatos a partir dos fatos existentes para a memória de trabalho. A inferência de novos fatos é o significado para o conceito “*alimentado*”. O conjunto de memória de trabalho e base de regras forma a base de conhecimento do sistema.

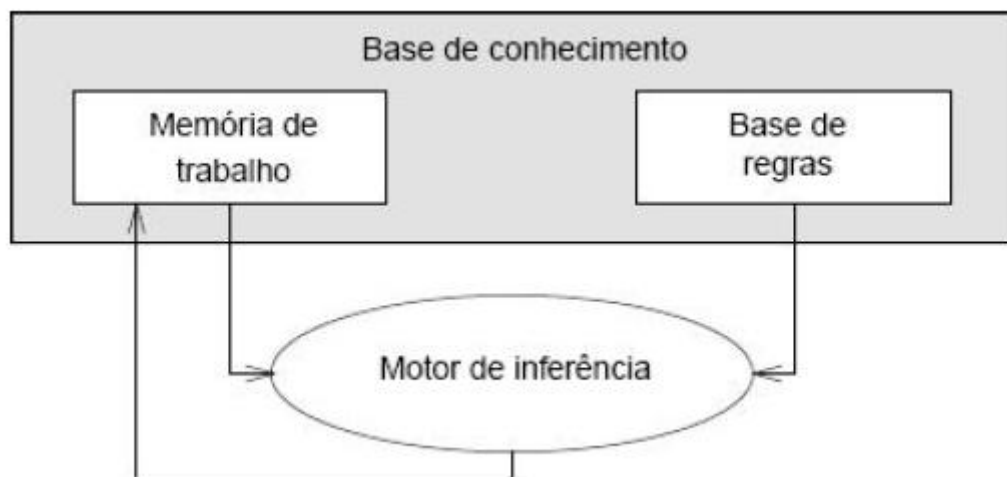


Figura 5. Arquitetura de um sistema de produção

O “casamento” de fatos e regras é feito pelo motor de inferência que emula o raciocínio humano. Existem dois tipos de encadeamento de raciocínio: encadeamento para frente (Forward Channing) e encadeamento para trás (Backward Channing). O primeiro deles parte da aplicação inicial das regras feita sobre a premissa. A saída de cada regra é entrada para outra regra, até não ser possível aplicar novas regras. Dessa forma, conclui-se que a premissa também é verdadeira. Caso contrário, ou seja, não houve convergência para um fato existente na memória, então a premissa passa a ser um novo fato para a memória. Porém, pode-se inferir para um absurdo, isto é, o resultado das aplicações das regras contradiz a

⁹ Do inglês: *Right Handle Side*

memória de trabalho. Como todos os fatos que estão contidos na memória são verdadeiros, então se conclui que a premissa é falsa. Os mesmos resultados ocorrem para o encadeamento para trás, contudo a diferença é a aplicação inicial das regras que é feita sobre a conclusão. Os fatos da memória devem coincidir com a premissa. Cada encadeamento possui um foco; o encadeamento para frente cria novos fatos enquanto o encadeamento para trás tenta validar a premissa.

Para elucidar a explicação a respeito dos tipos de encadeamentos, considere uma memória de trabalho que não contém nenhum fato e as seguintes formações de regras presentes em uma base de regras¹⁰:

Regra 1: Se glicose < 100 então

“Pessoa Saudável”

Regra 2: Se glicose ≥ 100 então

“Pessoa em Alerta”

Regra 3: Se “Pessoa em Alerta” e insulina < 80 então

“Pessoa Diabética”

Regra 4: Se “Pessoa em Alerta” e insulina > 80 então

“Pessoa Precisa Fazer Exercício”

Se os seguintes fatos glicose = 100 e insulina = 100 forem considerados como premissas e aplicando o encadeamento para frente, as regras serão executadas como a seguir:

Regra 1: Não será executada, devido as premissas não satisfazerem a condição da regra

Regra 2: Será executada e seu resultado (Pessoa em Alerta) será incluído na memória de trabalho.

Regra 3: Não será executada, pois apesar da Regra 2 produzir uma fato “Pessoa em Alerta”, o valor da insulina não satisfaz a condição.

Regra 4: Será executada devido existe na memória de trabalho uma fato “Pessoa em Alerta” e o valor da insulina satisfazer a condição da regra. Novamente, seu resultado (“Pessoa Precisa Fazer Exercício”) será incluído na memória de trabalho.

¹⁰ Os valores para formação dessas regras são aleatórios.

Como não existem outras regras a serem disparadas, a execução é encerrada. Note que nesta abordagem de encadeamento, foram produzidos novos fatos para a memória de trabalho. E uma vez incluídos, esses resultados são considerados verdadeiros.

Para o encadeamento para trás, deve-se admitir uma memória de trabalho contendo os fatos iniciais, pois o foco para esse tipo é a validação das premissas. A varredura das regras, neste caso, é feita pelo resultado das regras. Ou seja, enquanto no encadeamento para frente foram disparadas regras onde a condição é verdadeira, agora, as regras serão disparadas quando seu resultado for verdadeiro.

Usando as mesmas premissas anteriores e o fato inicial como sendo “Pessoa Precisa Fazer Exercício”, as regras serão executadas como a seguir:

Regra 4: Será executada devido existe na memória de trabalho uma fato “Pessoa Precisa Fazer Exercício” correspondente a conclusão desta regras. Sendo assim, as condições dessa regra são verdadeiras, comprovando a premissa inicial a respeito da insulina (a premissa indicava $\text{insulina} = 100$ e de fato pela regras $\text{insulina} > 80$). A condição “Pessoa em Alerta” também será incluída na memória de trabalho.

Regra 3: Não será executada, devido os fatos da memória de trabalho não satisfazerem a conclusão da regra “Pessoa Diabética”.

Regra 2: Será executada pois existe na memória de trabalho uma fato “Pessoa em Alerta” resultado da execução da Regra 4. Sendo assim, a conclusão dessa regra é verdadeira, e por consequência, a condição também é verdadeira. Assim $\text{glicose} \geq 100$ será incluída na memória de trabalho, comprovando a premissa $\text{glicose} = 100$.

Regra 1: Não será executada, devido os fatos da memória de trabalho não satisfazerem a conclusão da regra “Pessoa Saudável”.

3.2. DEFINIÇÃO DO JESS

Com base nos conceitos apresentados na seção anterior, o sistema Jess é definido em [9] como sendo:

“(...) um motor de inferência baseado em regras desenvolvido para a plataforma Java. A utilização do Jess é feita por declarações de regras e fatos que podem ser definidas em dois formatos: na linguagem de regras Jess ou em XML. Quando o motor de inferência é ativado, regras são disparadas conforme

o casamento de fatos, sendo que as implicações destas regras normalmente alteram o conjunto de fatos ou executam um código propiciado pela linguagem Java.”

Adaptando a Figura 5, é possível incluir o Jess na arquitetura dos sistemas de produção como mostrado na Figura 6, conforme a definição anterior. Com o Jess também é possível realizar, além das combinações de regras e fatos, a implementação dessas regras que serão usadas na inferência dos resultados do problema em questão, sendo utilizada uma linguagem própria¹¹ do Jess, embutida no sistema. O Jess tem por característica realizar a inferência do tipo encadeamento para trás, o que significa, conforme explicado na seção anterior, que o intuito Jess é validar as premissas que lhe são passadas.

O Jess possui compatibilidade com o Java, devido ser escrito nessa linguagem, e é independente do sistema operacional que está sendo usado, Windows, Linux ou Solaris. Esta compatibilidade permite que programas em Java executem programas descritos em Jess. Existem várias versões do Jess, e neste trabalho será usada a última versão do Jess, a versão 7.1. Para realizar sua instalação e execução deve-se garantir que exista uma máquina virtual Java (JVM) instalada, em qualquer versão a partir do JDK 1.4 até o mais recente, e que o ambiente de desenvolvimento Eclipse IDE nas versões superiores à 3.1 também esteja instalado.

Para instalar o Jess é necessário obter o arquivo de instalação compactado do site oficial <http://www.jessrules.com>. Sem executar o Eclipse, este arquivo deve ser descompactado em algum diretório de escolha do usuário e os arquivos contidos na pasta /eclipse desse mesmo arquivo devem ser descompactados também na pasta /plugin do diretório do Eclipse. Esse último processo de descompactação faz a instalação dos plugins do Jess no Eclipse. Após esse processo, para verificar se o Jess foi instalado com sucesso, o Eclipse deve ser executado, e abrindo o menu “Ajuda” no item “Sobre Eclipse SDK” deve atentar para a exibição do logo do Jess, como está destacado na figura 7. Os arquivos em Jess são criados com a extensão .clp, cujo ícone também é o logo e o código possui marcações com cores diferentes. É importante ressaltar que o download é gratuito para fins acadêmicos, cuja licença é renovável a cada 30 dias. Para uso não-acadêmico, existe uma versão *Trial*.

¹¹ Mais detalhes sobre a linguagem de regras Jess na seção 3.3 deste capítulo

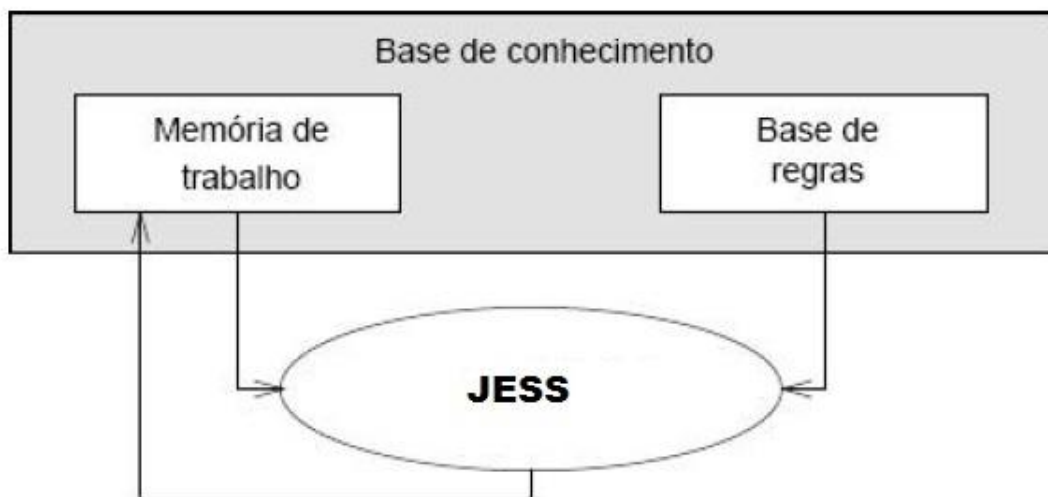


Figura 6. Jess acoplado à arquitetura do sistema de produção

Além do Jess, existem outros sistemas similares escritos em C/C++, porém segundo [9] Jess “(...) é mais eficiente mostrando melhores desempenhos (...) principalmente em problemas com grandes espaços de busca.”. Como sistemas baseado em regras possuem a memória de trabalho em constante alteração e, por outro lado as regras são fixas, deve ser empregado um mecanismo capaz de fazer as combinações necessárias em um tempo considerável e de forma eficiente. A eficiência do Jess é devida ao algoritmo que está por trás realizando o casamento dos fatos e regras. O algoritmo utilizado é o Algoritmo Rete (Charles L. Forgy- 1990) cuja ideia é testar somente fatos novos na memória mantendo um histórico dos testes realizados anteriormente. Dessa forma não seria necessário testar todos os fatos novamente quando houve alteração na memória de trabalho, implementação básica e ineficiente. A vantagem de usar o Rete é comprovada em termos computacionais passando da complexidade exponencial, da implementação básica, para a complexidade linear.

Apresentamos nesta seção outro método de programação voltado para sistemas especialistas: o Jess. Mas se comparado com o Prolog, é possível destacar sua principal vantagem em utilizar o tipo de encadeamento para frente, enquanto o Prolog utiliza o encadeamento para trás. Este último foca em validar as entradas a partir dos fatos já conhecidos e o Jess consegue obter respostas para as entradas que lhe são passadas, que é o intuito deste trabalho.

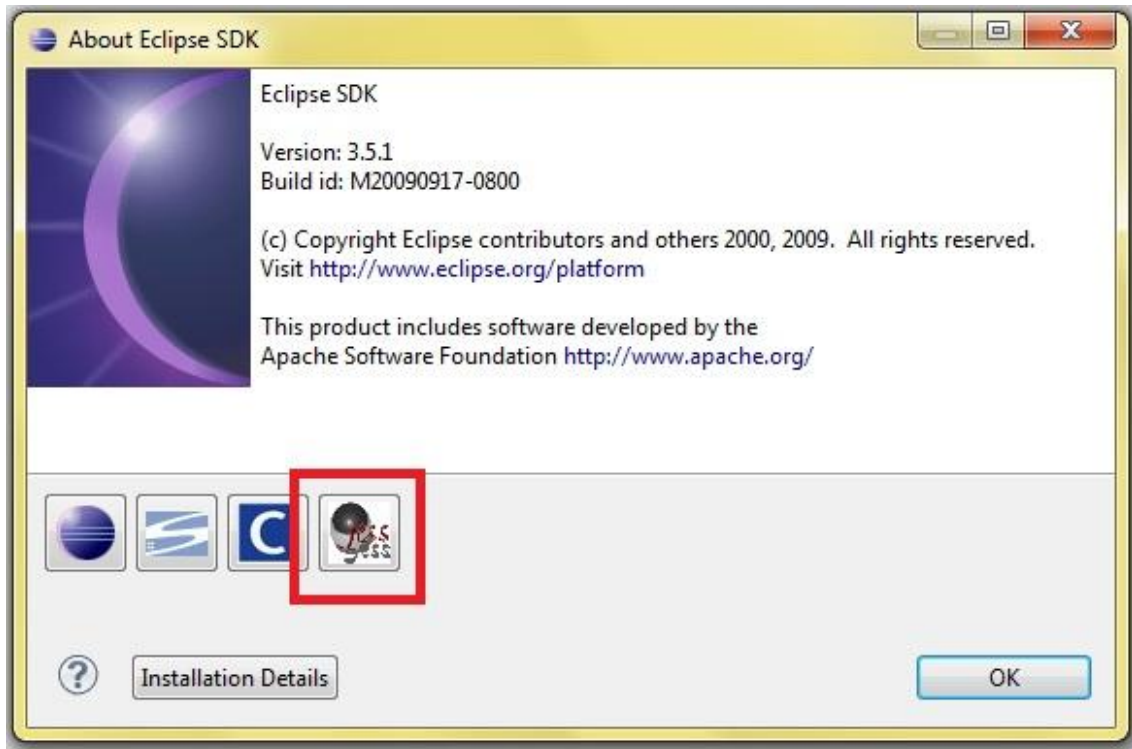


Figura 7. Tela de confirmação da instalação do Jess

3.3. LINGUAGEM JESS E COMUNICAÇÃO COM JAVA

Conforme citado na seção anterior, o Jess possui uma linguagem própria, o *Jess Rule Language*, baseada na linguagem LISP [McCarthy 1979], e segundo [9] o Jess possui “uma alta expressividade, com a possibilidade de escrita de relações lógicas complexas com pouco código.”. No entanto, o fato de possuir uma notação prefixada e uma quantidade excessiva de parênteses, a leitura e compreensão do código não são triviais.

3.3.1. Modelos

Como já definido na seção anterior, o Jess é um sistema baseado em regras e como todo o sistema desse tipo sua unidade de trabalho são os fatos presentes na memória de trabalho. Os fatos são formados pelos modelos (ou *templates*), que por sua vez possuem a sintaxe mostrada na figura 8(a) extraída de [9]. Cada *template* possui obrigatoriamente um nome, algumas declarações opcionais de extensões e comentários e uma lista de campos ou *slots* que contem as informações dos fatos. Os *slots* armazenam informação monovaloradas. Caso seja necessário armazenar uma informação multivalorada deve ser utilizada o *multislot*, que se comporta como uma lista. Em alguns casos há necessidade de associar um tipo à informação, sendo assim podem ser utilizados alguns dos tipos que o Jess dá suporte, como

mostrado na figura 8(a) extraída de [9]. A partir do *template* mostrado na figura 8(b) são construídos fatos contendo os dados sobre nome, taxa de glicose e de insulina. Os tipos que são definidos nos *slots* serão apresentados mais adiante.

```
(deftemplate template-name
  [extends template-name]
  ["Documentation comment"]
  [(declare (slot-specific TRUE | FALSE)
             (backchain-reactive TRUE | FALSE)
             (from-class class name)
             (include-variables TRUE | FALSE)
             (ordered TRUE | FALSE))]
  (slot | multislot slot-name
    [(type ANY | INTEGER | FLOAT |
          NUMBER | SYMBOL | STRING |
          LEXEME | OBJECT | LONG)]
    [(default default value)]
    [(default-dynamic expression)]
    [(allowed-values expression+)]*)
```

(a)

```
(deftemplate dados_entrada
  (slot nome)
  (slot valor_glicose (type integer))
  (slot valor_insulina (type integer))
)
```

(b)

Figura 8. (a) Sintaxe que define um template (b) Exemplo de declaração de template

3.3.2. Fatos

Na figura 9, é exemplificada a declaração de três fatos provenientes do template mostrado na figura 8(b), da seção anterior, e que são utilizados, neste caso, para inferir resultados sobre uma pessoa ser diabética ou não.

Todo fato possui um nome logo após a chamada do comando *deffacts*, usado para fazer a declaração dos fatos. Novamente, na figura 9, o nome dado ao fato é *dado_inicial*. Note que, a formação dos fatos é feita declarando o nome do *template*, bem como os *slots* contendo o dado que será armazenado¹². Esses fatos estão em constante movimentação de

¹² Os dados armazenados nos slots da figura são aleatórios e hipotéticos.

inclusão, exclusão e modificação na memória de trabalho. Para isso são utilizados os comandos **(assert)**, **(retract)**, **(modify)**, respectivamente, durante a execução das regras.

```
(deffacts dado_inicial
  (dados_entrada (nome "Simone") (valor_glicose 100) (valor_insulina 80))
  (dados_entrada (nome "André") (valor_glicose 120) (valor_insulina 80))
  (dados_entrada (nome "Viviane") (valor_glicose 100) (valor_insulina 90))
)
```

Figura 9. Exemplo de declaração de fatos

3.3.3. Regras

Sempre que ocorrer alguma movimentação na memória de trabalho, as regras são disparadas de forma a atualizar os fatos da memória (incluindo, excluindo ou modificando-os) mantendo consistente a memória de trabalho. Para criar uma regra, utiliza-se o comando *defrule*, com a sintaxe apresentada na figura 10(a). Note que, as regras são compostas de duas partes: a primeira anterior ao símbolo $\Rightarrow$, denominada LHS e a segunda posterior ao mesmo símbolo, a RHS (ambos já citados na seção 3.1). A LHS possui as condições que devem ser satisfeitas para que a regra possa ser executada. Aqui serão filtrados os fatos da memória sob os quais a regra poderá ser aplicada. Enquanto a RHS possui as ações que devem ser efetuadas sobre os fatos pertinentes a regra. Um exemplo de definição de uma regra é visualizado na figura 10(b). Importante ressaltar, nesse exemplo, que na LHS da regra foram filtrados fatos cujo *template* está no formato de *dado_inicial* (figura 9), porém as ações sobre esses fatos ocorrem na RHS, pois o que se quer da regra é verificar se o valor da glicose é maior que 150. Essa avaliação de intervalos (maior que, menor que, entre) não é possível na primeira parte da regra, devido à sua estrutura na linguagem Jess.

```
(defrule rule-name
  ["rule-description"]
  <match-definition>
  =>
  <action-definition>
)
```

(a)

```

(defrule regra2
  ?dado <- (dados_entrada)
  =>
  (if (and (> (fact-slot-value ?dado valor_glicose) 150)
        (<= (fact-slot-value ?dado valor_glicose) 220)) then
    (assert (resultado (mensagem "Pessoa em Alerta")
                        (nome (fact-slot-value ?dado nome))
                        (valor_glicose (fact-slot-value ?dado valor_glicose))
                        (valor_insulina (fact-slot-value ?dado valor_insulina))))
    )
)

```

(b)

Figura 10. (a) Sintaxe de uma regra (b) Exemplo de declaração de regra

Variáveis, Strings e Números

A RHS é composta de código de software, obviamente na linguagem Jess. Para escrever os códigos são necessárias as declarações de variáveis, que no Jess são feitas colocando um ‘?’ anterior ao símbolo da variável. Desse jeito, tanto ?x quanto ?casa são variáveis. Uma variável pode ser declarada em qualquer parte da RHS, sempre que necessário. As atribuições são possíveis através do comando (**bind** <variável> <valor>). Lembrando, que assim como em LISP, o Jess utiliza a notação prefixada e que a palavra subsequente ao ‘(’ é uma chamada de função.

O valor armazenado em uma variável pode ser uma string, números ou booleanos, já que não existe declaração de tipo no Jess. As strings são declarada entre “ ” e os números seguem o mesmo padrão da linguagem Java [9]. É importante ressaltar, que o Jess é *case sensitive*, isto é, diferencia maiúsculas de minúsculas, tanto para as strings quanto para as variáveis. Assim “Casa” é diferente de “casa”, o mesmo se aplica para ?x e ?X.

Controle de Fluxo

O Jess também permite o controle de fluxo utilizando os comandos: **while**, **if**, **for**, **foreach**, **break**, **try**. O controle do fluxo das regras se dá pelo LHS, que combina os fatos da memória a condição da regra. Sempre que um fato satisfizer a condição, a regra será disparada sobre ele, até que todos os fatos tenham sido testados. Quando um fato não valida a condição, a regra não será disparada para ele, mas seguirá sendo testado em outras regras.

Manipulação da Memória de Trabalho

Assim como já apresentado na seção 3.3.2, os três comandos que manipulam os fatos na memória são **(assert)**, **(retract)** e **(modify)**. Eles podem ser declarados na RHS da regra, seguinte a sintaxe **(assert/retract/modify (nome_template (nome_slot1 valor) (nome_slot2 valor) (...) (nome_slot_n valor)))**. Um *slot* pode ter um valor nulo, para isso utiliza-se o valor *nil*.

3.3.4 Funções

Assim como em outras linguagens, o usuário pode declarar suas próprias funções, passando um valor e retornando uma resposta. A figura 11¹³ exemplifica uma função que recebe uma lista de notas de um aluno, e retorna sua média. Para isso, além do parâmetro de entrada e a variável de retorno, foram definidas outras três variáveis associado-as a um valor. Para isso foi utilizado o comando **(bind)**, explicado na seção anterior.

A função, ao entrar no loop, representada pelo comando **(foreach)**, soma cada uma das notas bem como a quantidade de notas. O loop de iteração é encerrado quando não houver mais notas a serem processadas. Tal condição não é explícita no código, mas seu controle é feito internamente, o que é uma questão fora do escopo deste trabalho.

```
(deffunction calculaMedia ($?notas)
  (bind ?sumNotas 0.0)
  (bind ?totalNotas 0)
  (foreach ?nota $?notas
    (bind ?sumNotas (+ ?sumNotas ?nota) )
    (bind ?totalNotas (+ ?totalNotas 1))
  )
  (bind ?media (/ ?sumNotas ?totalNotas))
  (return ?media)
)
```

Figura 11. Exemplo de definição de função

3.3.5 Comunicação com Java

Para realizar a comunicação do Jess com Java, deve ser importado um pacote da biblioteca Jess que contém a classe Rete (Algoritmo Rete). O programa Java será escrito na classe convencional .java. Nessa classe deve existir um objeto da classe Rete que fará a

¹³ Para o exemplo já mostrado nas seções anteriores, as funções não são aplicáveis. Devido a isso, a figura 11 é proveniente de um exemplo onde é necessário realizar a média de notas.

execução dos arquivos .clp do Jess. Dessa forma, é possível coletar os dados no programa Java e enviá-los para serem submetidos às regras do Jess. Existem métodos próprios para realizar a passagem dos dados, que serão explicados na seção das regras que foram utilizadas neste trabalho. Um trecho do código de comunicação é visualizado na figura 12(b).

```
import jess.Context;
import jess.Deffacts;
import jess.Deftemplate;
import jess.Fact;
import jess.Funcall;
import jess.Jesp;
import jess.JessException;
import jess.RU;
import jess.Rete;
import jess.Value;
```

(a)

```
public static void main(String args[]) throws JessException, IOException{

    Rete engine = new Rete();

    FileReader file = new FileReader("C:\\Regras01.clp");
    try {
        Jesp parser = new Jesp(file, engine);
        parser.parse(false);
    } finally {
        file.close();
    }
}
```

(b)

Figura 12. (a) Biblioteca do Jess (b) Trecho de código de comunicação entre Jess e Java

3.4. EXEMPLO EM JESS

Após a breve apresentação da linguagem Jess na seção anterior, esta seção apresentará um programa escrito nessa linguagem. O objetivo do programa é inferir a partir da taxa de glicose e de insulina¹⁴, o estado de saúde de um paciente através das mensagens que serão armazenadas que podem ser “Pessoa Saudável” e “Pessoa em Alerta”. Sendo que esta última, ainda será avaliada como “Pessoa Diabética” ou “Pessoa precisa de atividade física”. Com o intuito de executar somente o arquivo.clp a obtenção dos dados é feita por uma regra

¹⁴ Os valores de referências mostrados nas regras são aleatórios e hipotéticos.

específica para tal objetivo, o que seria realizado pelo Java, caso houvesse uma comunicação ativa entre os arquivos Jess e Java.

No primeiro trecho do programa, mostrado na figura 13(a), são exibidas as declarações dos *templates* e a inclusão do fato inicial para dar a primeira movimentação na memória de trabalho, de forma a iniciar a execução das regras. A condição da regra (a LHS) *obter_dados* é verificar a existência de algum fato do tipo (*phase obter_dados*). Caso exista, o fato será armazenado na variável *?phase*, e o programa segue com a execução da RHS da regra, solicitando os dados para execução do programa. Os dados solicitados são o nome do paciente, e as taxas de glicose e de insulina do mesmo. Esses dados são armazenados no respectivo *template*, e incluído na memória de trabalho, através do comando (*assert*). Com mais essa inclusão na memória, as regras são disparadas sobre esse fato.

Nas regras 1 e 2 visualizadas na figura 13(b), são criados novos fatos (cujo *template* é definido como *alarme* na figura 13(a)) a partir da avaliação feita sobre as taxas de glicose e insulina. Importante ressaltar que a condição verificada na LHS é a mesma para ambas as regras, porém somente uma delas será verdadeira devido a condição do *if* na RHS. Uma a uma as informações são armazenadas nos *slot* do *template*, e é realizada a inclusão do fato na base. Fica implícito que ocorreram duas verificações de condições. A primeira delas durante a LHS das regras, pois filtra somente os fatos em que é pertinente a aplicação daquela regra. A segunda validação, em geral, é feita na RHS¹⁵ da regra, sendo específica para o fato filtrado pela LHS, que dependerá da implementação e do objetivo da regra.

As regras 3 e 4 poderão ser disparadas assim que existir um fato do tipo *resultado* na base com a mensagem “Pessoa em Alerta”. Caso essa condição se verifique, será executada a RHS, que por sua vez também contém uma validação, que resultará na criação ou não de um novo fato.

Note que a conclusão de uma regra pode ser a condição para a execução de outra regra, sendo essa a dinâmica da programação baseada em regras. Seu término ocorre no momento em que não houver nenhuma modificação (seja ela inclusão, modificação ou exclusão) nos fatos da base. A execução do programa descrito é mostrada na figura 13(c), onde se verifica a inclusão de dois pacientes com suas respectivas taxas de glicose e insulina. Ao primeiro paciente, foi aplicada a regra 1, da figura 13(b), devido a existir um fato do tipo *dado_inicial* (que é o fato formado pelos dados de entrada) e também devido a taxa de glicose ser menor que o indicado, satisfazendo também a condição da RHS da regras. Como

¹⁵ Devido o formato das regras, que conforme mencionado no início do capítulo é do tipo se-então

resultado, foi incluído o fato alarme, contendo, além dos dados do paciente a mensagem de “Pessoa Saudável”. Como não existe outra regra a ser aplicado a esse fato, o programa pergunta ao usuário se ele deseja continuar com a execução ou deseja encerrar o programa. Como o usuário solicitou a continuidade da execução, então foram solicitados novamente outros dados de entrada.

Ao segundo paciente incluído na memória (cujo processo de inclusão é o mesmo executado para o primeiro paciente), foi aplicada a regra 2, da figura 13(b), devido ser verdadeira as condições tanto para a LHS (existe um fato do tipo dado_inicial) quanto para a RHS (é verdadeira a condição para a taxa de glicose). Neste caso, como um novo fato do tipo alarme contendo a mensagem “Pessoa em Alerta” foi incluído, configura-se uma condição verdadeira para a LHS tanto para regra 3 quanto para a regra 4, também mostradas na figura 13(b), onde será verificada a taxa de insulina do paciente. Para a taxa de insulina informada, foi executada a regra 3, resultando em mais um fato do tipo alarme que contém a mensagem "Pessoa Diabética", além dos dados do paciente. Como não existem mais regras para o fato incluído, novamente é perguntado ao usuário se ele deseja continuar a execução, cuja resposta foi negativa, o que encerra o programa.

Antes de o usuário ser questionado sobre a continuidade da execução do programa, é exibida a pilha de fatos presentes na memória de trabalho. Em sua última exibição, é observado que, diferentemente do primeiro paciente, o segundo possui dois fatos associados a ele resultantes da execução da regra 2 e 3, respectivamente.

```

(deftemplate dados_entrada
  (slot nome)
  (slot valor_glicose (type integer))
  (slot valor_insulina (type integer))
)

(deftemplate resultado
  (slot mensagem)
  (slot nome)
  (slot valor_glicose)
  (slot valor_insulina)
)

(deffacts dado_inicial
  (phase obter_dados))

(defrule obter_dados
  ?phase <- (phase obter_dados)
  =>
  (printout t "Entre com o nome do paciente:")
  (bind ?nome (read))

  (printout t "Entre com o valor da glicose:")
  (bind ?valor_glicose (read))

  (printout t "Entre com o valor da insulina:")
  (bind ?valor_insulina (read))

  (assert (dados_entrada (nome ?nome) (valor_glicose ?valor_glicose)
    (valor_insulina ?valor_insulina)))

  (retract ?phase)
  (assert (phase re-start))
)

```

(a)

```

(defrule regra1
  ?dado <- (dados_entrada)
  =>
  (if (<= (fact-slot-value ?dado valor_glicose) 150) then
    (assert (resultado (mensagem "Pessoa Saudavel")
      (nome (fact-slot-value ?dado nome))
      (valor_glicose (fact-slot-value ?dado valor_glicose))
      (valor_insulina (fact-slot-value ?dado valor_insulina))))
    )
  )

(defrule regra2
  ?dado <- (dados_entrada)
  =>
  (if (and (> (fact-slot-value ?dado valor_glicose) 150)
    (<= (fact-slot-value ?dado valor_glicose) 220)) then
    (assert (resultado (mensagem "Pessoa em Alerta")
      (nome (fact-slot-value ?dado nome))
      (valor_glicose (fact-slot-value ?dado valor_glicose))
      (valor_insulina (fact-slot-value ?dado valor_insulina))))
    )
  )

(defrule regra3
  ?dado <- (resultado (mensagem "Pessoa em Alerta"))
  =>
  (if (<= (fact-slot-value ?dado valor_insulina) 80) then
    (assert (resultado (mensagem "Pessoa Diabetica")
      (nome (fact-slot-value ?dado nome))
      (valor_glicose (fact-slot-value ?dado valor_glicose))
      (valor_insulina (fact-slot-value ?dado valor_insulina))))
    )
  )

(defrule regra4
  ?dado <- (resultado (mensagem "Pessoa em Alerta"))
  =>
  (if (> (fact-slot-value ?dado valor_insulina) 80) then
    (assert (resultado (mensagem "Pessoa precisa de atividade física")
      (nome (fact-slot-value ?dado nome))
      (valor_glicose (fact-slot-value ?dado valor_glicose))
      (valor_insulina (fact-slot-value ?dado valor_insulina))))
    )
  )

```

(b)

```

Console X
<terminated> exemplo3.clp [Jess Application] C:\Arquivos de programas\Java\jre6\bin\javaw.exe (20/10/2010 21:48:08)

Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 104 day(s).
Entre com o nome do paciente: Simone
Entre com o valor da glicose: 100
Entre com o valor da insulina: 80
f-0 (MAIN::initial-fact)
f-2 (MAIN::dados_entrada (nome Simone) (valor_glicose 100) (valor_insulina 80))
f-4 (MAIN::alarme (mensagem "Pessoa Saudavel") (nome Simone) (valor_glicose 100) (valor_insulina 80))
For a total of 3 facts in all modules.
Quer continuar? (s/n) s
Entre com o nome do paciente: André
Entre com o valor da glicose: 160
Entre com o valor da insulina: 70
f-0 (MAIN::initial-fact)
f-2 (MAIN::dados_entrada (nome Simone) (valor_glicose 100) (valor_insulina 80))
f-4 (MAIN::alarme (mensagem "Pessoa Saudavel") (nome Simone) (valor_glicose 100) (valor_insulina 80))
f-6 (MAIN::dados_entrada (nome André) (valor_glicose 160) (valor_insulina 70))
f-8 (MAIN::alarme (mensagem "Pessoa em Alerta") (nome André) (valor_glicose 160) (valor_insulina 70))
f-9 (MAIN::alarme (mensagem "Pessoa Diabetica") (nome André) (valor_glicose 160) (valor_insulina 70))
For a total of 6 facts in all modules.
Quer continuar? (s/n) n

```

(c)

Figura 13. (a) Trecho de inicialização do programa em Jess (b) Regras executadas no programa (c) Execução do programa em Jess

CAPÍTULO 4 - PRESSURETOOL: UMA VISÃO DETALHADA DO SISTEMA

O objetivo deste capítulo é mostrar de forma detalhada o funcionamento do *PressureTool* tanto em uma visão técnica quanto na visão de um usuário que estaria utilizando o sistema.

4.1. ESCOPO DO SISTEMA

Nesta seção será apresentada toda a análise de requisitos do sistema *PressureTool*. A análise inclui os requisitos funcionais e os não-funcionais levantados pelos autores deste trabalho considerando as necessidades de um médico para realizar consultas com seus pacientes. Ainda serão mostradas as descrições de casos de uso e diagramas UML referente ao sistema.

Requisitos Funcionais

- ✓ Cadastrar dados pessoais do paciente
- ✓ Importar dados da MAPA disponíveis na central
- ✓ Avaliar a pressão do paciente
- ✓ Avaliar risco do paciente
- ✓ Editar dados do paciente
- ✓ Visualizar dados do paciente
- ✓ Excluir dados do paciente
- ✓ Cadastrar nova consulta do paciente
- ✓ Visualizar dados da consulta do paciente
- ✓ Fazer login no sistema
- ✓ Importar dados do médico

Requisitos Não-Funcionais

Sendo o objetivo deste trabalho apresentar o que o sistema é capaz de fazer para auxiliar o médico na geração de um diagnóstico de hipertensão os requisitos funcionais foram priorizados durante a implementação, por representarem as reais funções do sistema. Por isso, os requisitos não-funcionais não foram levados em consideração. No entanto, caso tivessem sido implementados estariam em conformidade com a lista abaixo.

- ✓ Disponibilidade
 - O sistema deve ficar disponível para coletar as medições de pressão do paciente
 - O sistema deve ficar disponível para emitir alertas
- ✓ Eficiência

- O sistema deve ser capaz de coletar 30 medições de pressão e calcular a média em 3 segundos, quando necessário.
- ✓ Integridade
 - O sistema deve ser capaz de manter inalterados os dados cadastrados
 - O sistema não deve permitir que os dados sejam alterados sem confirmação
- ✓ Interoperabilidade
 - O sistema deve ser capaz de importar dados em outros formatos
- ✓ Usabilidade
 - Um usuário novo deve conseguir realizar um cadastro em 5 minutos
 - O sistema de ter um módulo de ajuda ao usuário
- ✓ Manutenibilidade
 - O sistema deve possuir uma documentação compreensível a um usuário técnico

Diagrama de Casos de Uso

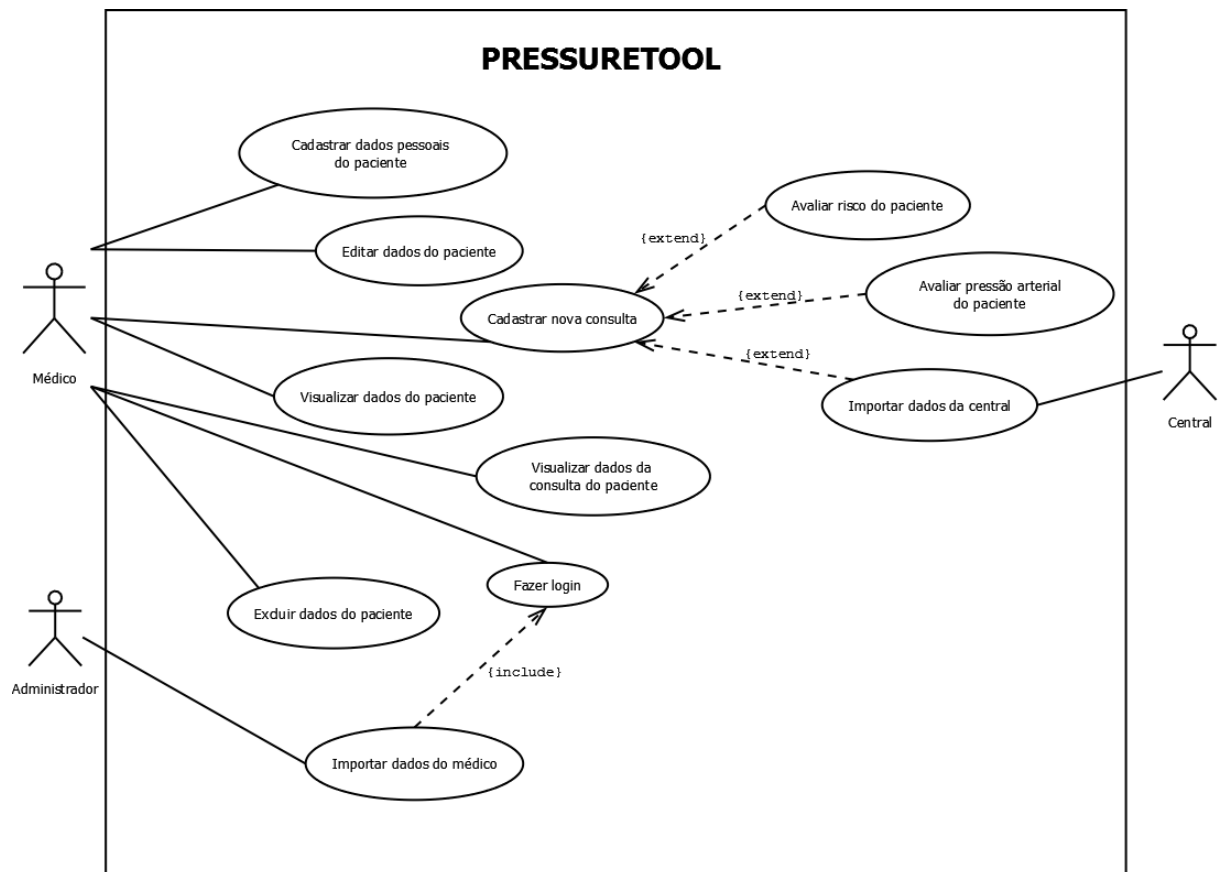


Figura 14. Diagrama de Casos de Uso

Casos de Uso

- UC1: O médico cadastra dados pessoais do paciente
- Cenário Típico
 - 1. O médico seleciona opção de cadastro dos dados do paciente.
 - 2. O sistema exibe uma tela de cadastro
 - 3. O médico preenche os campos com os dados do paciente.
 - 4. O médico solicita cálculo da idade do paciente
 - 5. O sistema calcula a idade do paciente
 - 6. O sistema exibe a idade do paciente na tela
 - 7. O médico seleciona os fatores de risco do paciente
 - 8. O médico solicita inclusão dos dados do paciente na base de dados
 - 9. O sistema valida os dados
 - 10. O sistema exibe os dados do paciente
 - 11. O médico confirma os dados do paciente
 - 12. O sistema grava os dados do paciente
 - 13. O sistema exibe confirmação de gravação de dados
- Cenário Alternativo
 - 5.a: O sistema não calcula idade do paciente
 - 1. O sistema exibe mensagem de obrigatoriedade de preenchimento da data de nascimento
 - 2. Retorna ao passo 3
 - 9.a: Os dados informados são inválidos
 - 1. O sistema exibe mensagem de erro
 - 2. Retorna ao passo 3
 - 11.a: O médico não confirma os dados
 - 1. Retorna ao passo 3.
- UC2: O médico importa dados da MAPA disponíveis na central
- Cenário Típico
 - 1. O médico confirma que o paciente possui dados da MAPA
 - 2. O sistema habilita opção de importar dados da MAPA
 - 3. O médico seleciona opção de importar dados da MAPA
 - 4. O sistema busca os dados do paciente informado
 - 5. O sistema transfere os dados da MAPA do paciente na central para o médico
 - 6. O sistema exibe os dados importados
- Cenário Alternativo
 - 4.a: O sistema não localiza o paciente
 - 1. O sistema exibe uma mensagem de erro
 - 2. Retorna ao passo 3
 - 5.a: Não há dados do paciente a serem importados
 - 1. O sistema exibe uma mensagem de inexistência de dados a serem importados
 - 2. Retorna ao passo 3
- UC3: O médico avalia a pressão arterial do paciente
- Cenário Típico
 - 1. O médico seleciona opção de execução da regra considerando pressão, idade e atividade
 - 2. O sistema verifica os dados necessários para execução

- 3. O sistema executa as regras de inferência que avaliam a pressão arterial
- 4. O sistema exibe o resultado na tela
- Cenário Alternativo
 - 1.a: A regra é executada para paciente gestante considerando, também a semana da gestação
 - 1. Retorna ao passo 2
 - 2.a: Não há dados necessários para avaliação
 - 1. Uma mensagem é exibida informando a falta de dados
 - 2. Retorna ao passo 1
 - 3.a: O sistema falha ao executar as regras de inferência que avaliam a pressão
 - 1. O sistema exibe uma mensagem de erro
 - 2. Retorna ao passo 1
- UC4: O médico avalia riscos do paciente
- Cenário Típico
 1. O médico seleciona opção de execução da regra considerando pressão, idade e fatores de risco
 2. O sistema verifica os dados necessários para execução
 3. O sistema executa as regras de inferência que avaliam os riscos do paciente
 4. O sistema exibe o resultado na tela
- Cenário Alternativo
 - 2.a: Não há dados necessários para avaliação
 - 1. Uma mensagem é exibida informando a falta de dados
 - 2. Retorna ao passo 1
 - 3.a: O sistema falha ao executar as regras de inferência que avaliam a pressão
 - 1. O sistema exibe uma mensagem de erro
 - 2. Retorna ao passo 1
- UC5: O médico edita dados do paciente
- Cenário Típico
 1. O médico seleciona opção de edição dos dados do paciente
 2. O médico informa o nome do paciente que será alterado
 3. O sistema busca o cadastro do paciente
 4. O sistema retorna uma lista de cadastros de paciente com o nome informado
 5. O médico seleciona o paciente na lista
 6. O sistema exibe uma tela contendo os dados do paciente
 7. O médico altera os dados do paciente
 8. O médico solicita alteração dos dados na base de dados
 9. O sistema valida os dados informados
 10. O sistema exibe as informações atuais do paciente
 11. O médico confirma as alterações
 12. O sistema salva as alterações na base de dados
 13. O sistema exibe mensagem de confirmação da alteração
- Cenário Alternativo
 - 4.a: O sistema não localiza o paciente
 - 1. O sistema exibe uma mensagem de inexistência de pacientes com o nome informado
 - 2. Retorna ao passo 13
 - 5.a: O médico não seleciona nenhum paciente da lista
 - 1. O médico cancela a operação

- 2. Retorna ao passo 13
- 9.a: O sistema não valida os dados
 - 1. O sistema exibe mensagem de erro
 - 2. Retorna ao passo 7
- 11.a: O usuário não confirma as alterações
 - 1. Retorna ao passo 3
- UC6: O médico visualiza dados do paciente
- Cenário Típico
 - 1. O médico seleciona a opção de consultar o cadastro do paciente
 - 2. O médico informa ao sistema o nome do paciente a ser consultado
 - 3. O sistema busca o cadastro do paciente
 - 4. O sistema retorna uma lista de cadastros de paciente com o nome informado
 - 5. O médico seleciona da lista o cadastro do paciente desejado
 - 6. O sistema exibe uma tela contendo os dados do paciente
 - 7. O médico finaliza a tela
- Cenário Alternativo
 - 4.a: O sistema não localiza o paciente
 - 1. O sistema exibe mensagem de inexistência de pacientes com o nome informado
 - 2. Retorna ao passo 8
- UC7: O médico exclui paciente
- Cenário Típico
 - 1. O médico seleciona a opção de exclusão do cadastro do paciente
 - 2. O médico informa o nome do paciente a ser removido
 - 3. O sistema retorna uma lista de cadastros de paciente com o nome informado
 - 4. O médico seleciona da lista o cadastro do paciente desejado
 - 5. O sistema pede confirmação da exclusão do paciente
 - 6. O médico confirma a exclusão
 - 7. O sistema remove todos os dados do paciente (cadastro, consulta e medições da MAPA)
- Cenário Alternativo
 - 3.a: O sistema não localiza o paciente
 - 1. O sistema exibe mensagem de inexistência do paciente
 - 2. Retorna ao passo 2
 - 4.a: O médico não seleciona nenhum paciente
 - 1. Retorna ao passo 8
 - 6.a: O médico não confirma a exclusão
 - 1. Retorna ao passo 8
- UC8: O médico cadastra nova consulta do paciente
- Cenário Típico
 - 1. O médico seleciona opção de cadastrar nova consulta
 - 2. O sistema solicita nome do paciente a ser buscado
 - 3. O médico informa o nome do paciente a ser buscado
 - 4. O sistema busca o paciente indicado
 - 5. O sistema exibe uma tela de cadastro de dados da consulta
 - 6. O sistema preenche automaticamente os campos de nome do paciente e data da consulta

- 7. O médico preenche a pressão sistólica medida no consultório
- 8. O médico preenche a pressão diastólica medida no consultório
- 9. O médico preenche a prescrição médica da consulta
- 10. O médico solicita inclusão da consulta na base de dados
- 11. O sistema valida os dados informados
- 12. O sistema exibe os dados da consulta do paciente
- 13. O médico confirma os dados da consulta do paciente
- 14. O sistema grava os dados da consulta do paciente
- Cenário Alternativo
 - 4.a: O cadastro do paciente não foi encontrado
 - 1. O sistema exibe mensagem de inexistência do cadastro do paciente
 - 2. Retorna ao passo 1
 - 10.a: A paciente está grávida
 - 1. O médico seleciona a indicação de que a paciente está grávida
 - 2. O sistema habilita a opção de cadastro da semana atual da grávida
 - 3. O médico preenche a semana atual da gravidez
 - 4. Retorna ao passo 10
 - 10.b: O médico deseja importar dados da MAPA da central
 - 1. O caso de uso UC2- O médico importa dados da MAPA disponíveis na central é chamado
 - 2. Retorna ao passo 10
 - 10.c: O médico deseja avaliar pressão arterial
 - 1. O caso uso UC3- O médico avalia a pressão arterial do paciente é chamado
 - 2. Retorna ao passo 10
 - 10.d: O médico deseja avaliar riscos do paciente
 - 1. O caso uso UC4- O médico avalia riscos do paciente é chamado
 - 2. Retorna ao passo 10
 - 11.a: Os dados informados são inválidos
 - 1. O sistema exibe uma mensagem de erro
 - 2. Retorna ao passo 5
 - 12.a: O médico não confirma os dados
 - 1. Retorna ao passo 5
- UC9: O médico visualiza os dados da consulta do paciente
- Cenário Típico
 - 1. O médico informa ao sistema o paciente a ser consultado
 - 2. O sistema busca as consulta do paciente
 - 3. O sistema exibe uma lista contendo as consultas do paciente
 - 4. O médico seleciona uma consulta da lista
 - 5. O sistema exibe a consulta selecionada
- Cenário Alternativo
 - 2.a: O sistema não localiza o paciente
 - 1. Uma mensagem de inexistência do paciente é exibida
 - 2. Retorna ao passo 1
 - 4.a: O médico não seleciona nenhuma consulta
 - 1. Retorna ao passo 6
- UC10: Fazer login no sistema
- Cenário Típico

- 1. O usuário seleciona o perfil do usuário
- 2. O usuário informa o login e a senha
- 3. O usuário envia o login e a senha
- 4. O sistema verifica o login e a senha
- 5. O sistema autoriza o uso
- Cenário Alternativo
 - 4.a: O sistema não localiza login e senha
 - 1. O sistema exibe mensagem
 - 2. Retorna ao passo 1
- UC11: O administrador importa dados do médico
- Cenário Típico
 - 1. O caso de uso UC10- Fazer login no sistema é chamado
 - 2. O administrador seleciona opção de importar dado do médico
 - 3. O sistema solicita o nome do médico
 - 4. O administrador informa o nome do médico
 - 5. O sistema requisita à central o médico solicitado
 - 6. A central retorna os dados do médico solicitado
 - 7. O sistema armazena localmente os dados recebidos da central
 - 12. O sistema exibe mensagem de confirmação da importação
 - 13. O caso de uso é encerrado
- Cenário Alternativo
 - 8.a: O administrador não informa o nome do médico
 - 1. O sistema exibe mensagem
 - 2. Retorna ao passo 7
 - 9.a: A central está indisponível
 - 1. O sistema exibe mensagem
 - 2. Retorna ao passo 13

Diagrama de Implantação

Para estruturar a comunicação entre os componentes do sistema em que está inserido o *PressureTool*, foi criado o diagrama de deployment apresentado na figura 15. Nele é visualizado o software do paciente, que é a parte do sistema instalado na residência do mesmo com o intuito de monitorar sua rotina médica, utilizando um mecanismo de execução do plano de cuidados¹⁶. Além dos sensores que captam as informações relevantes para o monitoramento (mecanismo de obtenção de dados), o software do paciente é composto dos mecanismos de análise dos dados obtidos e o mecanismo de envio para a central dos dados e análise. O software do paciente ainda possui um banco de dados local de forma a criar um histórico individual do mesmo.

A central de dados faz a ligação entre os dados obtidos pelo software do paciente e o software do especialista que fará a avaliação. Basicamente, a central é composta de banco de dados dos

¹⁶ Conforme convencionado no capítulo 2 deste trabalho, entende-se por plano de cuidado as prescrições médicas designadas ao paciente.

planos de cuidados dos diversos pacientes assistidos pelo sistema bem como seus históricos e suas medições da MAPA. Outro banco utilizado é de armazenamento dos dados dos médicos cadastrados no sistema.

Por fim, o software do especialista ou PressureTool se comunica diretamente com a central importando os dados da MAPA através do mecanismo de importação de dados. Com esses dados é possível executar as regras de inferência (Regras.clp) do mecanismo de execução de regras. Tais resultados e outras informações relativas ao paciente são armazenados na base de dados dos pacientes e, eventualmente, podem ser submetidos ao mecanismo de cálculos para realizar cálculos dinâmicos para complementar determinadas informações. O mecanismo de edição de plano de cuidados não foi contemplado neste trabalho, por não possuir relação direta com as prescrições médicas do paciente.

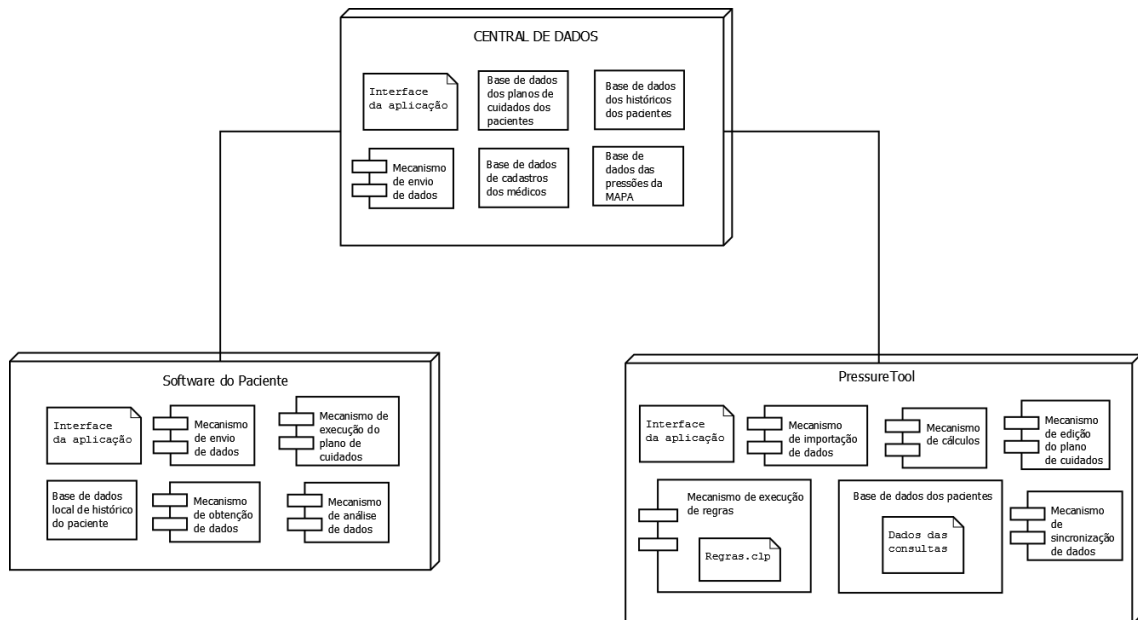
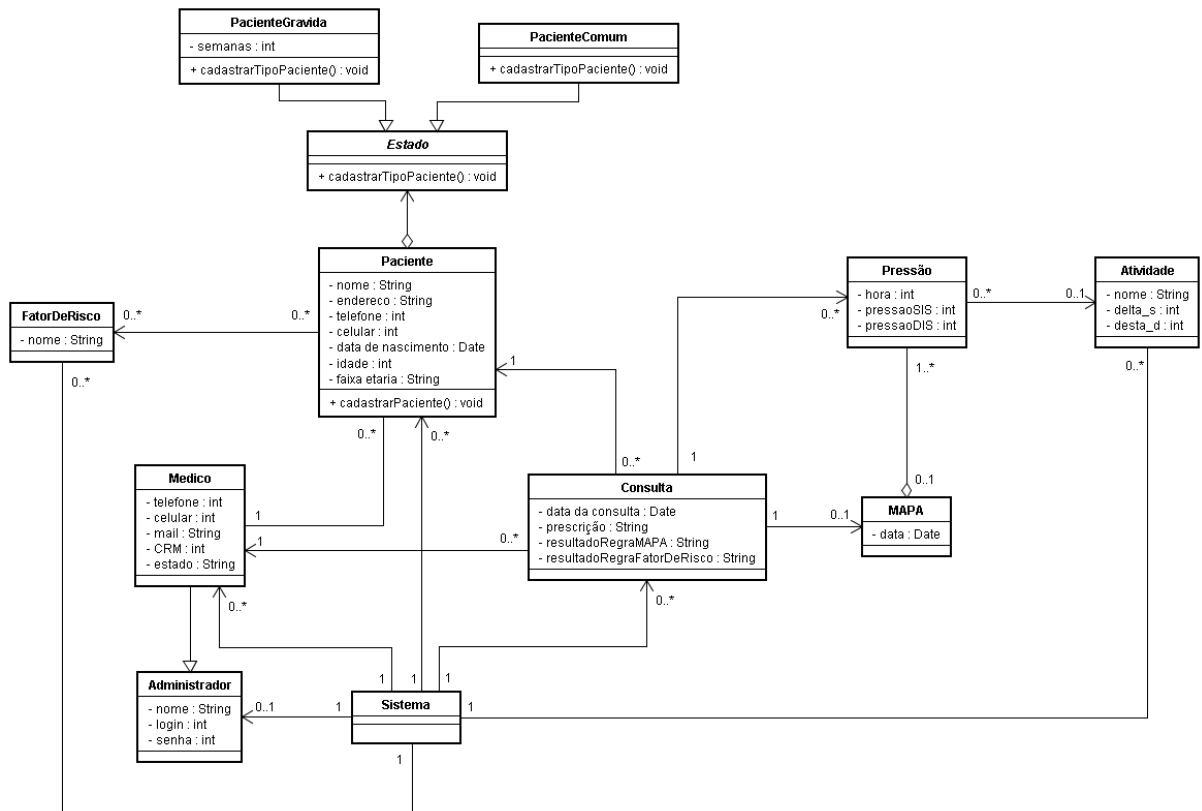


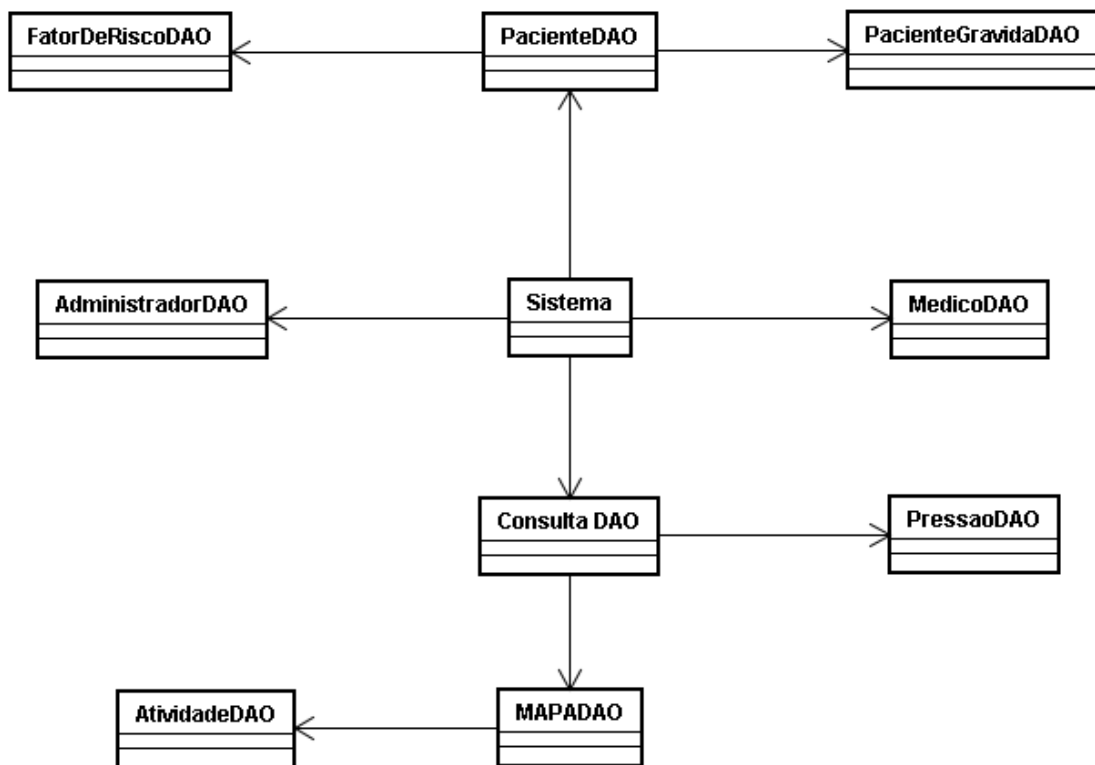
Figura 15. Diagrama de Implantação

Diagrama de Classes

O modelo do *PressureTool* foi elaborado como apresentado na figura 16(a). Nela é apresentado o diagrama de classe de projeto da camada de negócio. Como o sistema trabalha com manipulação de uma base dados, ainda há necessidade de criação de um diagrama de classe em nível de design, como mostrado na figura 16(b).



(a)



(b)

Figura 16(a). Diagrama de Classes das Camadas de Negócio (b) Diagrama de Classes da Camada de Banco de Dados

Diagrama de Sequência

Os diagramas de sequência apresentados a seguir retratam os dois casos mais importantes na obtenção dos dados necessários para a avaliação da pressão arterial do paciente. O primeiro diagrama mostra o Cadastro do Paciente desde a coleta das informações na interface, passando pela associação com os fatores de risco, que serão utilizados na avaliação de risco do paciente, até a inclusão desses dados na base de dados. Todas as informações dos pacientes estão em memória e nela são preparadas para só então essas informações serem incluídas na base de dados. O segundo diagrama apresenta o Cadastro da Nova Consulta, em que são informadas as pressões medidas no consultório, a importação dos dados da MAPA da central é feita e as regras de avaliação da pressão são executadas. Após todos os dados estarem completos as regras poderão se executadas, e posteriormente, a nova consulta incluída na base de dados.

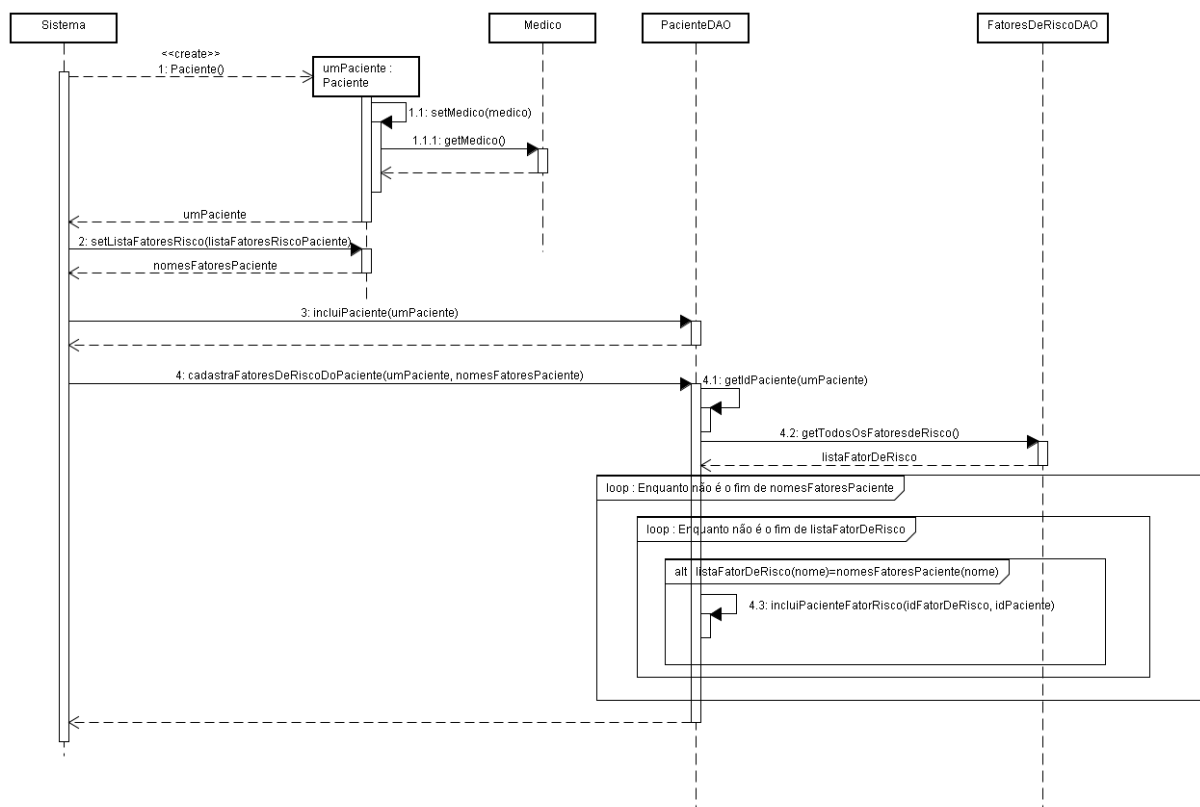


Figura 17. Diagrama de Sequência: Cadastrar novo paciente

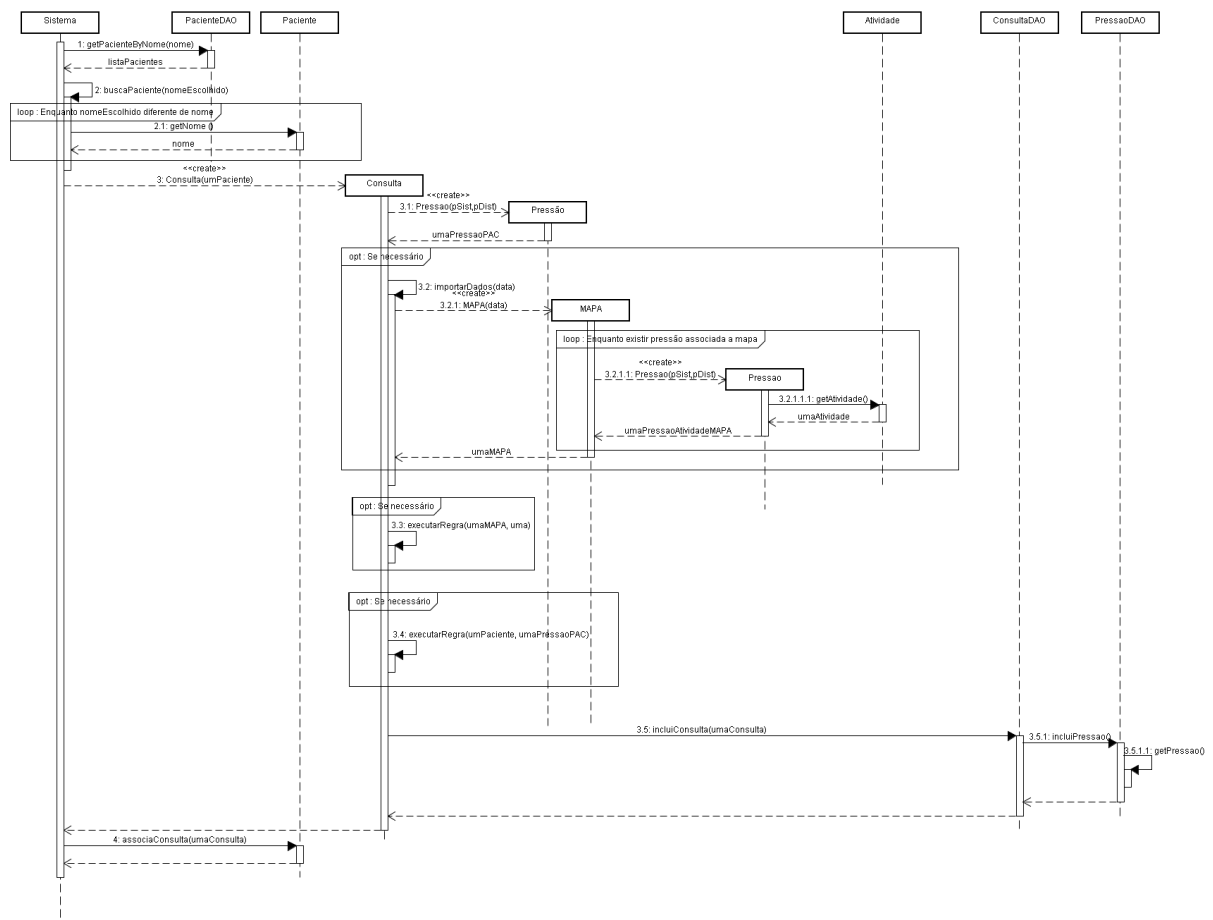


Figura 18. Diagrama de Sequência: Cadastrar nova consulta

4.2 DETALHES DA IMPLEMENTAÇÃO

O sistema foi criado na linguagem Java e acoplado com H2 [10], um framework de comunicação com banco de dados para Java. Ainda para comunicação com o banco de dados utilizamos a abordagem de mapeamento objeto-relacional, criando classes de banco específicas para o modelo implementado. Tais classes foram classificadas como Classes DAO.

Foram utilizados três padrões de projeto na implementação: o Singleton [11], Facade [12] e o State[16]. O primeiro deles permite criar uma e somente uma instância de uma determinada classe. No caso, como a classe Sistema (já apresentada no diagrama de classe) é classe onde tudo se inicia somente uma instância dessa classe será criada. E por consequência imediata, somente uma conexão com o banco de dados é instanciada, que devido ao Singleton, garante que todas as classes que realizem esse acesso só sejam criadas uma única vez. O Facade foi utilizado com o intuito de deixar independentes as classes da interface das

classes lógicas. Por fim, o State foi utilizado para definir o comportamento de uma paciente ao se encontrar gestante.

4.3 AS REGRAS DE INFERÊNCIA DO PRESSURETOOL

O *PressureTool* possui a funcionalidade de realizar dois tipo de inferência. Para isso, está embutida no sistema uma regra¹⁷ para cada um desses tipos. O usuário do sistema poderá optar pela execução de qualquer uma das regras, uma vez que elas são independentes uma da outra, e desde que as informações para execução já tenham sido fornecidas. A implementação das regras se encontra no Apêndice I.

A primeira regra executada pelo *PressureTool*, denominada de **Avaliar Risco do Paciente**, referencia a idade, a pressão do consultório utilizadas da mesma forma descrita na regra anterior, e uma lista de fatores de risco extraída de [3] apresentada na Tabela 3. A regra só poderá ser executada ao serem fornecidas todas essas informações, caso contrário, a opção de execução se mantém desabilitada.

A regra classifica o paciente segundo sua pressão arterial, como visualizado na Tabela 3. Após realizar essa classificação, é então analisado o risco cardiovascular adicional do paciente considerando a classificação dada e os fatores de risco, obtidos do cadastro do paciente. Dentre os fatores de risco associados ao paciente as doenças cardiovasculares, lesão do órgão alvo e diabetes melito são predominantes sobre os demais fatores. O resultado apresentado pode ser: **Sem risco adicional, Risco baixo, Risco médio, Risco alto e Risco muito alto** (Tabela 4). Esses resultados podem indicar se é necessário que o paciente utilize a MAPA: quanto mais alto o risco, maior a necessidade.

Os resultados provenientes das regras são armazenados no formulário de consulta do paciente.

¹⁷ Tais regras foram implementadas na linguagem própria do Jess, um motor de inferência, já apresentado no capítulo 3 deste trabalho.

Fatores de Risco
Tabagismo
Dislipidemias
Diabetes Melito
Doenças Cardiovasculares
Lesão do Orgão Alvo
Idade acima de 60 anos
Nefropatia
Histórico familiar de doença cardiovascular em mulheres com menos de 65 anos
Histórico familiar de doença cardiovascular em homens com menos de 55 anos

Tabela 1. Lista de fatores de risco contemplados no PressureTool

Classificação	Pressão sistólica (mmHg)	Pressão diastólica (mmHg)
Ótima	< 120	< 80
Normal	< 130	< 85
Limítrofe	130-139	85-89
Hipertensão estágio 1	140-159	90-99
Hipertensão estágio 2	160-179	100-109
Hipertensão estágio 3	≥ 180	≥ 110
Hipertensão sistólica isolada	≥ 140	< 90

Tabela 2. Classificação de pressão arterial para adultos acima de 18 anos

Fatores de risco	Pressão arterial				
	Normal	Limítrofe	Hipertensão estágio 1	Hipertensão estágio 2	Hipertensão estágio 3
Sem fator de risco	Sem risco adicional		Risco baixo	Risco médio	Risco alto
1 a 2 fatores de risco	Risco baixo	Risco baixo	Risco médio	Risco médio	Risco muito alto
3 ou mais fatores de risco ou lesão de órgãos-alvo ou diabetes melito	Risco médio	Risco alto	Risco alto	Risco alto	Risco muito alto
Doença cardiovascular	Risco alto	Risco muito alto	Risco muito alto	Risco muito alto	Risco muito alto

Tabela 3. Risco cardiovascular adicional ao paciente, segundo seus fatores de risco

A primeira regra, denominada **Avaliar Pressão Arterial**, utiliza a informação de idade, classificada dentro das faixas etárias de criança (0-11 anos), jovem (12-17 anos), adulto (18-59 anos) e idoso (acima de 60 anos)¹⁸. Além da informação de idade, obtida do cadastro do paciente, também são consideradas a pressão medidas no consultório e as pressões da MAPA associada a atividade que estava sendo realizada no momento da medição. Importante

¹⁸ Dados oficiais classificados pelo Organização Mundial de Saúde (OMS)

ressaltar que essa regra só poderá ser disparada se houver dados provenientes do exame da MAPA. Caso não existam tais informações, a opção de execução da regra fica desabilitada na interface.

Com todos esses dados fornecidos e validados, a regra então poderá ser executada e o seu retorno é um relatório contendo a pressão do consultório, as várias pressões da MAPA, a atividade de cada uma das pressões da MAPA e o resultado inferido pela regra para cada uma das pressões (Figura 18). Os resultados inferidos podem ser **Normotenso**, **Hipertensão**, **Hipertensão do Avental Branco** ou **Hipertensão Mascarada**.

“A normotensão se caracteriza por valores normais de pressão arterial no consultório (abaixo de 140/90 mm Hg) e na MAPA de 24 horas (igual ou abaixo de 130/80 mm Hg) (...) enquanto a hipertensão se caracteriza por valores anormais de pressão arterial no consultório (igual ou acima de 140/90 mm Hg) e na MAPA de 24 horas (acima de 130/80 mm Hg) (...).

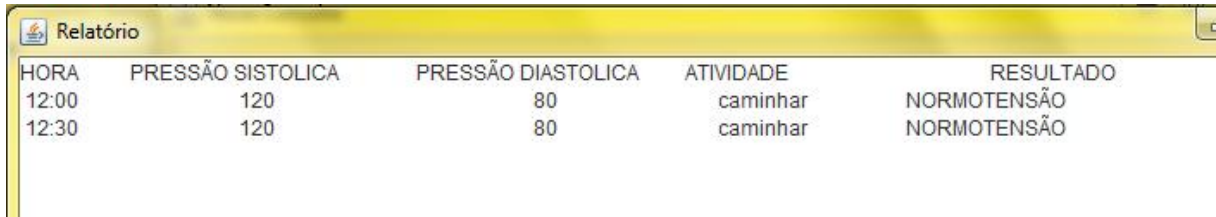
(...) A hipertensão avental branco ocorre quando há valores anormais da medida da pressão arterial no consultório (igual ou acima de 140/90 mm Hg) e valores normais de pressão arterial pela MAPA durante o período de vigília (igual ou abaixo de 135/85 mm Hg) (...).

A hipertensão mascarada ocorre quando há valores normais na medida da pressão arterial no consultório (abaixo de 140/90 mm Hg) e valores anormais de pressão arterial pela MAPA durante o período de vigília (acima de 135/85 mm Hg) (...).”

[2]

Há valores de referência específicos para cada faixa etária. Os valores citados acima são referência para os adultos e idosos. No caso dos jovens e crianças, [2] mostra que há dificuldades quanto a utilização da MAPA devido a falta de valores de normalidade. Atualmente, para essa faixa etária, os valores se reduzem de 5% do valor dos adultos na

vigília e 10% do valor dos adultos no sono. É devido à variação dos valores de referência que a idade deve ser um fator a ser considerado durante a inferência.



HORA	PRESSÃO SISTOLICA	PRESSÃO DIASTOLICA	ATIVIDADE	RESULTADO
12:00	120	80	caminhar	NORMOTENSÃO
12:30	120	80	caminhar	NORMOTENSÃO

Figura 19. Trecho de Relatório de Avaliação de Pressão Arterial

Além da idade, essa regra considera a atividade que fora exercida pelo paciente durante a medição da MAPA. Esse registro é importante devido à pressão oscilar conforme a atividade, podendo até ultrapassar os valores de referência considerados normais, contudo não configurando um caso de anormalidade de pressão arterial, devido a atividade apresentar um intervalo que deve ser adicionado (ou reduzido da) a pressão medida. Neste caso, haverá um novo valor de referência para avaliação da pressão. A partir da informação da atividade e sua variação, é possível fazer uma inferência mais completa, que considere todo o contexto em que se encontra o paciente, separando as situações normais daquelas que precisam de uma atenção especial. As atividades utilizadas no *PressureTool* estão apresentadas na Tabela 1 extraída de [2]

Variações da pressão arterial (mm Hg), de acordo com várias situações e atividades		
Atividades	Pressão Sistólica	Pressão Diastólica
Reuniões	+ 20,2	+ 15,0
Trabalho	+ 16,0	+ 13,0
Caminhar	+ 12,0	+ 5,5
Vestir	+ 11,5	+ 9,7
Tarefas domésticas	+ 10,7	+ 6,7
Telefonar	+ 9,5	+ 7,2
Conversar	+ 6,7	+ 6,7
Assistir TV	+ 0,3	+ 1,1
Repouso	0	0
Dormir	- 10,0	- 7,6

Tabela 4. Tabela de atividades do PressureTool

A regra avalia também as pacientes gestantes. Para isso a mesma deve ter um diagnóstico anterior indicado para hipertensão. O resultado da avaliação deve indicar a presença de hipertensão do avental branco ou hipertensão mascarada, ou ainda a própria

hipertensão. Serão consideradas a semana de gestação, as informações da MAPA, as atividades e as respectivas variações de pressões, prolongando o intervalo de referência considerado normal.

Essa identificação na gravidez torna-se importante, pois se evita “*o tratamento desnecessário e potencialmente lesivo ao feto*” [2]. Os valores de referências e suas variações para as gestantes são mostrados na Tabela 2 extraída de [2]. Note que, não há valores de referência para desde a primeira semana de gravidez até a oitava semana, pois não há comprovação desses valores devido a gestação ainda estar em seu início.

Valores da MAPA (mm Hg) durante os trimestres da gravidez				
	Semanas de Gestação			
	9-16	18-24	26-32	33-40
Vigília				
Pressão Sistólica	115±8	115±8	116±9	119±9
Pressão Diastólica	70±7	69±6	70±7	74±7
Sono				
Pressão Sistólica	100±7	99±8	101±8	108±8
Pressão Diastólica	55±5	54±6	55±6	58±7

Tabela 5. Tabela de valores de referência para gestantes

4.4 DETALHES DO FUNCIONAMENTO

Iniciando o PressusreTool

Ao iniciar o *PressureTool*, uma tela de login é exibida (Figura 20), solicitando o usuário, senha e o tipo do usuário que está iniciando a sessão do sistema, podendo ser Administrador ou Médico. Após a validação do login, as ações permitidas ao tipo de usuário selecionado são habilitadas e estão organizadas conforme o tipo de dado a ser manipulado. Para o Administrador, serão habilitadas as funcionalidades de manipulação dos dados do médico, tais como cadastro, edição e exclusão e inclusão de atividades. Neste caso, existem dois menus, um dedicado ao médico e outro dedicado às atividades. Já para o Médico, serão habilitadas as funcionalidades de manipulação dos dados do paciente, da consulta e ainda um menu de ajuda. Novamente, para cada funcionalidade existe um menu dedicado.

Realize o login para iniciar o PressureTool

Perfil de Login: Médico ▼

Usuário:

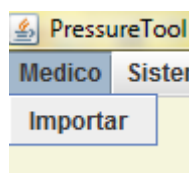
Senha:

Detailed description: This is a login window titled 'Realize o login para iniciar o PressureTool'. It features a dropdown menu for 'Perfil de Login' with 'Médico' selected and 'Administrador' as an alternative option. Below this are text input fields for 'Usuário' and 'Senha'. An 'Entrar' button is positioned to the right of the password field.

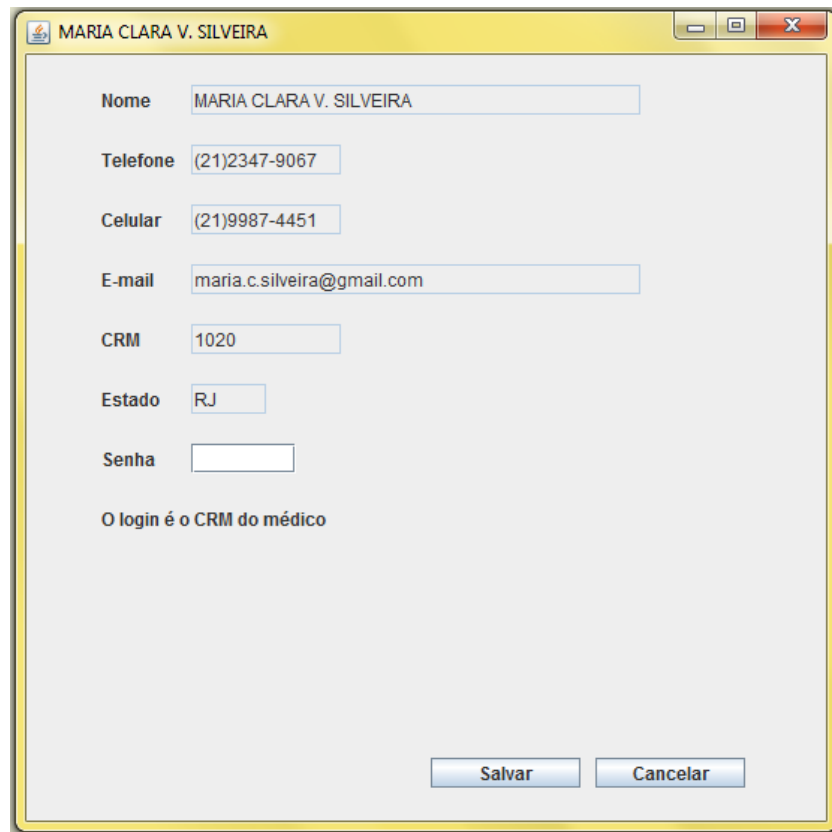
Figura 20. Tela de Login

Importando os dados do médico da central

O Administrador está permitido a realizar a importação dos dados do médico da central para a base local onde o *PressureTool* foi instalado. Após logar-se, será habilitado o menu visualizado na Figura 21(a) e para importar o dado do médico o Administrador deve selecionar a opção Importar e informar o nome do médico. O sistema realiza uma requisição ao servidor, passando o nome como parâmetro e como resultado a central deverá retornar as informações do médico informado na tela exibida na Figura 21(b). Note que todos os dados do médico não são editáveis a menos da senha que deve ser preenchida pelo médico. Como já explicado, esses dados são armazenados localmente somente por uma questão de desempenho para a funcionalidade de login do médico, que não necessitará realizar uma conexão com a central, bastará consultar a base local. Como premissa para a importação dos dados do médico, admite-se que o médico já teve seus dados cadastrados na central pelo próprio Administrador.



(a)



A screenshot of a web application window titled "MARIA CLARA V. SILVEIRA". The window contains a form with the following fields: "Nome" (filled with "MARIA CLARA V. SILVEIRA"), "Telefone" (filled with "(21)2347-9067"), "Celular" (filled with "(21)9987-4451"), "E-mail" (filled with "maria.c.silveira@gmail.com"), "CRM" (filled with "1020"), "Estado" (filled with "RJ"), and "Senha" (empty). Below the fields, a note states "O login é o CRM do médico". At the bottom right, there are two buttons: "Salvar" and "Cancelar".

(b)

Figura 21. (a) Menu de funcionalidades para o administrador (b) Tela de importação do médico

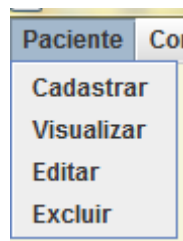
Manipulando os dados do paciente

As ações sob os dados do paciente são permitidas para o Médico e são feitas a partir do menu dedicado para essa finalidade (Figura 22(a)). É possível cadastrar, editar, visualizar e excluir um paciente.

Para o cadastro do paciente, uma tela é exibida (Figura 22(b)) contendo os campos de nome, endereço, telefone residencial, telefone celular, data de nascimento, idade e fatores de risco. A idade não será preenchida pelo médico, ela é calculada pelo sistema a partir da data de nascimento informada. Os fatores de risco cadastrados no sistema estão listados na tela de cadastro, permitindo a seleção de nenhum ou mais de um fator de risco associado ao paciente. Ao final o médico deve salvar o cadastro do paciente e só será incluído no banco se todos os campos obrigatórios estiverem corretamente preenchidos. A qualquer momento o cadastro pode ser cancelado ou mesmo apagar todos os campos preenchidos. No cadastro do paciente se encontram as informações que podem ser alteradas poucas vezes ou não mais serem alteradas.

Na edição do cadastro do paciente, o mesmo será buscado pelo nome e ao ser encontrado, a mesma tela de cadastro será exibida, porém com os campos preenchidos com os dados já cadastrados. Somente alguns campos poderão ser editado, como endereço, telefone fixo e celular e os fatores de risco, e ao serem submetidos a atualização no banco, serão novamente validados.

A visualização, bem como a exclusão do cadastro é buscada também pelo nome. A primeira delas possui a função somente de consultar e nada poderá ser alterado enquanto a exclusão só será efetuada, após a confirmação da mensagem que será exibida. Assim, todos os dados do paciente, das pressões da MAPA, das consultas também serão excluídos.



(a)

A imagem mostra uma janela de software intitulada 'Cadastro de Paciente'. Ela contém os seguintes campos e elementos:

- Campos de texto para: Nome, Endereço, Telefone (com um campo para o dígito entre parênteses e um hífen), Celular (com um campo para o dígito entre parênteses e um hífen), e Data Nascimento (com dois campos para o dia e o mês).
- Um campo rotulado 'Idade' com um botão 'Calcular Idade' ao lado.
- Uma lista de fatores de risco, cada um com uma caixa de seleção desmarcada:
 - Tabagismo
 - Diabetes Melito
 - Dislipidemias
 - Lesão em Órgãos Alvo
 - Idade acima de 60 anos
 - Doenças Cardiovasculares
 - Nefropatia
 - Histórico doença cardiovascular em mulheres abaixo de 65 anos na família
 - Histórico doença cardiovascular em homens abaixo de 55 anos na família
- Dois botões na base: 'Incluir' e 'Limpar'.

(b)

Figura 22. (a) Menu de paciente e opções de cadastrar, visualizar, editar e excluir cadastro do paciente (b) Tela de novo cadastro de paciente

Manipulando os dados da consulta

A funcionalidade mais importante do *PressureTool* é o cadastro de nova consulta, pois nela estão englobadas uma série de outras funcionalidades, como mostrado na Figura 23(a). Para realizar uma nova consulta, presente nas ações permitidas ao médico, o paciente deve estar cadastrado no sistema. Caso essa condição não seja satisfeita, uma mensagem será exibida informando ao médico que o paciente deve ser cadastrado. Caso o paciente seja encontrado, o sistema exibe a tela de cadastro da consulta (Figura 23(b)) com alguns campos já preenchidos automaticamente, como o nome do paciente, data em que está sendo realizada a consulta e os fatores de risco extraídos do cadastro do paciente. Outros campos devem ser preenchidos pelo médico como a prescrição médica daquela consulta, as pressões medidas no consultório e se for o caso, incluir a semana de gravidez da paciente. Essa informação é obtida durante a consulta e atualizada no cadastro da paciente e por ser uma condição temporária é registrada numa tabela a parte do cadastro do paciente.

Tendo as informações da pressão no consultório, é possível executar a regra que considera a idade, pressão no consultório e os fatores de risco do paciente, presentes em seu cadastro. O resultado será exibido na tela da nova consulta.

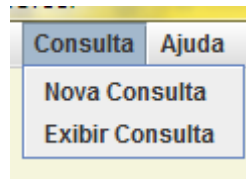
Para executar a regra que considera a idade, pressão no consultório, média da pressão arterial da MAPA e atividade, é necessário importar essas medições da MAPA da central. Para isso, deve-se primeiro indicar que o paciente possui um exame a ser importado. Dessa forma, serão habilitadas as funções de importar MAPA¹⁹ e execução da regra, porém esta última ainda verifica se os dados já foram importados. Em caso positivo, a execução é realizada e o resultado é exibido na tela. As regras são independentes umas das outras, podendo ser executadas juntas ou não e ainda dependendo da intenção do médico, também podem não ser executadas.

É a partir dos resultados auxiliares das regras e somados ao conhecimento e experiência do médico, que se configura o objetivo desta ferramenta de auxílio na obtenção de diagnósticos para hipertensão arterial.

Outra funcionalidade associada a consultas, é a busca por uma consulta anterior de um determinado paciente. Para isso, deve ser informado o nome do paciente, e ao ser encontrado, é retornada uma lista de consultas deste paciente cabendo ao médico selecionar aquela que atende ao seu objetivo.

¹⁹ A MAPA será importada no formato: Hora, Pressão Sistólica, Pressão Diastólica e Atividade. Este formato foi determinado devido as necessidades do sistema, não sendo um padrão adotado no mercado.

Não será possível fazer edição de uma consulta passada, nem excluir uma consulta já cadastrada de um paciente. A exclusão só será possível ao ser excluído o paciente do sistema.



(a)

 A tela de cadastro de nova consulta, intitulada 'Nova Consulta'. Ela contém campos para 'Nome' (preenchido com 'CLAUDIO CAMARGO SOUZA') e 'Data' (preenchido com '05/12/2010'). Há campos para 'Pressão Sistólica Consultorio' e 'Pressão Diastólica Consultorio'. Abaixo, há uma seção para 'Paciente Grávida' com um checkbox desativado e um campo 'Quantidade de Semanas'. Há também uma seção para 'Nível Hipertensão' com um botão 'Avaliar Risco'. Abaixo disso, há uma seção para 'Possui MAPA' com um checkbox desativado e um campo 'Pressões da MAPA'. Há botões 'Importar' e 'Avaliar Pressão'. À direita, há uma seção 'Fatores de Risco' com uma lista de itens: 'Tabagismo', 'Nefropatia', 'Doenças Cardiovasculares' e 'Histórico doença cardiovascular em homens abaixo de'. Abaixo, há uma seção 'Diagnóstico' com quatro opções de radio button: 'Hipertensão', 'Hipertensão Mascarada', 'Normotensão' e 'Hipertensão Avental Branco'. Há também uma seção 'Prescrição' com um campo de texto grande. Na parte inferior, há botões 'Salvar', 'Limpar' e 'Cancelar'.

(b)

Figura 23.(a)Menu de consulta do sistema e opções de cadastrar e exibir consulta (b)Tela de cadastro de nova consulta

Pedindo ajuda

A qualquer momento, o médico poderá retirar suas dúvidas quanto ao sistema, selecionando a opção de ajuda (Figura 24). Um arquivo abrirá, contendo toda a explicação do sistema. Ainda nesse menu, é possível obter outras informações do sistema na opção Sobre.

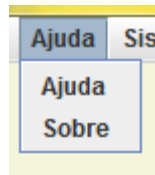


Figura 24. Menu de ajuda do sistema

Encerrando e Fazendo Logoff do PressureTool

Para ambos os tipos de perfil de usuário, é possível fazer logoff e voltar para a tela inicial de login, e encerrar a execução do sistema. Tais opções estão presentes no menu Sistema mostrado na Figura 25.

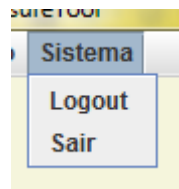


Figura 25. Menu de Sistema: Logout e Encerramento do sistema

CAPÍTULO 5 - TRABALHOS RELACIONADOS

Durante a nossa pesquisa encontramos outros sistemas que possuem características semelhantes àsquelas do *PressureTool*, apresentadas neste trabalho. Dentre eles, destacamos inicialmente [13] que, apesar de possuir uma abordagem voltada para monitorização residencial do paciente, possui a funcionalidade de inferir um resultado que informe a situação real do paciente.

Igualmente como mostrado no *PressureTool*, o público alvo de [13] são pacientes acometidos de hipertensão arterial, cuja monitorização deve ser uma prática constante. Essa monitorização, em [13], consiste de um ambiente composto de sensores que capta três classes de informações: aquelas relativas ao ambiente onde o paciente está presente, outras referentes à fisiologia do paciente e por último, informações comportamentais do mesmo, isto é, quais tipos de atividades estão sendo realizadas por ele. Essas informações são importantes, pois influenciam na oscilação da pressão arterial, tanto de forma regular como também de forma anormal. Após a coleta dos dados relevantes, é possível inferir um resultado sobre a saúde do paciente a partir da regularidade ou alteração da pressão arterial do mesmo.

A inferência feita em [13] é modelada na Lógica Fuzzy sob três graus de pertinência. O resultado apresentado para o paciente pode estar classificado como: **situação normal, em alerta** ou **emergência**. Para cada um desses três níveis, o sistema realiza uma ação sob o paciente, chegando até chamar o serviço de emergência, se o paciente estiver em estado de emergência.

No *PressureTool*, a inferência é realizada pelo médico do paciente, em seu consultório, coletando informações fisiológicas e comportamentais e outras que não são levadas em consideração em [13] como idade, fatores de risco, semanas de gravidez (se for o caso), pois todas essas também contribuem para uma análise individual do paciente. Como informações fisiológicas, o *PressureTool* coleta os dados sobre a pressão arterial, podendo ser aquela medida no consultório, como também, se houver, as medições de pressão arterial realizadas por MAPA. As atividades do paciente só são utilizadas na inferência no *PressureTool* quando houver as pressões da MAPA, pois uma determinada pressão pode ter sido medida como alta, mas se ao levar em consideração a atividade realizada pelo paciente essa medição pode ser considerada normal. É neste ponto que ambos os sistemas se assemelham, no que tange a distinção das reais situações de risco daquelas em que houve uma

alteração de pressão sem necessariamente caracterizar uma situação de risco. As informações do ambiente de [13] são desconsideradas no *PressureTool*.

CAPÍTULO 6 - CONCLUSÃO

Devido não existir sintomas clínicos para a hipertensão, atualmente para se ter um diagnóstico para a mesma é necessário realizar exames complementares, em particular a MAPA. Sendo assim, ressaltamos como conclusão deste trabalho, que o sistema aqui apresentado, o *PressureTool*, pode ser utilizado como um complemento na obtenção de um diagnóstico mais preciso e confiável para a hipertensão, agindo de forma conjunta com a MAPA e a experiência do médico.

Devido sua funcionalidade de análise individual do paciente, que considera as pressões não de forma isolada, mas associada à idade, aos fatores que aumentem os riscos de desenvolver a hipertensão e às atividades cotidianas realizadas pelo paciente, é possível obter um diagnóstico mais abrangente, coerente e consistente, pois se avalia todo o contexto em que o paciente está inserido. No caso das mulheres, ainda é possível fazer uma avaliação quando a mesma se encontra na condição de gestante. Nesta situação, a hipertensão tardiamente identificada, pode trazer riscos a mãe e a criança. Realizando uma avaliação prévia e diferenciada diminuem-se esses riscos, em especial a eclampsia que pode levar ambos a óbito.

Além de se tentar buscar uma avaliação mais individualizada do paciente, o *PressureTool* ainda funciona como um gerenciador de cadastros do pacientes, armazenando-os em uma base local e fazendo a manutenção (inclusão, visualização, edição e exclusão) dos mesmos.

Esperamos, com este trabalho, contribuir para um melhor atendimento aos pacientes que possuem esta doença, e com isso melhorar a qualidade de vida dos mesmos. Porém, sabemos que o sistema ainda possui algumas limitações e que é possível dar continuidade. Para isso propomos uma interface com layout mais moderno e consolidação dos requisitos não funcionais listados neste trabalho. Visando uma comunicação completa com o SCIADS, deve ser aumentado o domínio do *PressureTool* de forma que ele possa receber os dados do monitoramento do plano de cuidado do paciente e alterá-lo de forma dinâmica se necessário. Tendo um domínio maior, propomos também a construção de novas regras que possam considerar esses novos dados recebidos e que essas regras possam ser implementadas utilizando agentes de software especializados de forma a automatizar a execução das regras. Por fim, sabemos que ainda existe o desafio da resistência a adoção dessa tecnologia por parte do público-alvo ao qual o sistema foi destinado. Assim, propomos que seja realizada uma pesquisa com médicos, paciente e até mesmos os estudantes de medicina para que seja avaliada a aceitação e viabilidade de implementação desse sistema.

REFERÊNCIAS BIBLIOGRAFICAS

- [1] *Jess, the Rule Engine for the Java™ Platform*. Disponível em: <<http://www.jessrules.com>>. Acessado em: março de 2010.
- [2] SOCIEDADE BRASILEIRA DE CARDIOLOGIA. *IV Diretriz para Uso da Monitorização Ambulatorial da Pressão Arterial, II Diretriz para uso da Monitorização Residencial da Pressão Arterial, IV MAPA / II MRPA*. Disponível em: <<http://publicacoes.cardiol.br/consenso>>. Acessado em: março de 2010.
- [3] SOCIEDADE BRASILEIRA DE CARDIOLOGIA. *V Diretrizes Brasileiras de Hipertensão Arterial – 2006*. Disponível em: <<http://publicacoes.cardiol.br/consenso>>. Acessado em: abril de 2010.
- [4] MANUAL de assistência domiciliar na atenção primária à saúde, Porto Alegre, Novembro 2003.
- [5] S. T. Carvalho, A. Copetti, A. Sztajnberg, M. Erthal, R. Santos, O. Loques. Sistema de Assistência Domiciliar à Saúde Telemonitorada. Relatório Técnico, Universidade Federal Fluminense- Niterói, 2009
- [6] *Wikipedia*. Disponível em: <<http://pt.wikipedia.org/wiki/Telemedicina>>. Acessado em Novembro de 2010.
- [7] SCIADS: Sistema Computacional Inteligente de Assistência Domiciliar à Saúde. Projeto de Pesquisa, UFF- Niterói, Fevereiro de 2010
- [8] M. X. Py. Sistemas Especialistas: uma introdução. Universidade do Rio Grande do Sul- Porto Alegre. 2007. Disponível em: <<http://www.inf.ufrgs.br/procpar/disc/cmp135/trabs/mpy/sistemaspecialistas.pdf>>. Acessado em Agosto de 2010.
- [9] G. F. Budke, D. Mayer. *Jess. The Rule Engine for the Java Platform*. Universidade Federal de Santa Catarina. Florianópolis, 2008. Disponível em: <http://www.das.ufsc.br/~gb/pg-ia/Jess08/Artigo_Jess_Daniel_Gerson.pdf>
- [10] *H2*. Disponível em: < <http://www.h2database.com/html/main.html>>. Acessado em Setembro de 2010
- [11] *Wikipedia*. Disponível em: < <http://pt.wikipedia.org/wiki/Singleton> >. Acessado em Novembro de 2010.
- [12] *Wikipedia*. Disponível em: < <http://pt.wikipedia.org/wiki/Fa%C3%A7ade>>. Acessado em Novembro de 2010.
- [13] A. Copetti; O. Loques; J.C.B Leite; T. Barbosa; A. Nóbrega. Intelligent Context-Aware Monitoring of Hypertensive Patients. In: Pervasive Health Workshop, London, UK, 2009. Disponível em: < <http://www.tempo.uff.br/?codpag=4> >. Acessado em Setembro 2010
- [14] H. M. Deitel, P. J. Deitel. *Java, Como programar*. Prentice Hall, 6º edição, 2005

- [15] R. R. Serson. Programação Orientada a Objetos com Java 6 Curso Universitário, Brasport, 2007
- [16] *DevMedia*. Disponível em: < <http://www.devmedia.com.br/post-16783-Design-Patterns-State-Parte-4.html>>. Acessado em Novembro de 2010

APÊNDICE I

IMPLEMENTAÇÃO DA REGRAS DE INFERÊNCIA EM JESS

1. Regra de avaliação de pressão arterial do paciente

Inicialização dos templates

```
(deftemplate paciente
  (slot id)
  (slot nome)
  (slot idade)
  (slot tipo)
  (slot status))

(deftemplate mulher_gravida
  (slot id)
  (slot nome)
  (slot idade)
  (slot semanas_gestacao)
  (slot periodo)
  (slot status))

(deftemplate atividade
  (slot tipo)
  (slot s)
  (slot d))

(deftemplate pressao
  (slot id_paciente)
  (slot PAC_S)
  (slot PAC_D)
  (slot MAPA_S)
  (slot MAPA_D)
  (slot atividade))

(deffacts inicio
  (phase inclui_atividade))

(defrule inclui_atividade
  ?inicio <- (inclui_atividade)
  =>
  (assert (atividade (tipo "assistiTV") (s 0.3) (d 1.1)))
  (assert (atividade (tipo "caminhar") (s 12) (d 5.5)))
  (assert (atividade (tipo "conversar") (s 6.7) (d 6.7)))
  (assert (atividade (tipo "dormir") (s -10) (d -7.6)))
  (assert (atividade (tipo "repouso") (s 0) (d 0)))
  (assert (atividade (tipo "reuniao") (s 20.2) (d 15)))
  (assert (atividade (tipo "tarefasDomesticas") (s 10.7) (d 6.7)))
  (assert (atividade (tipo "telefonar") (s 9.5) (d 7.2)))
  (assert (atividade (tipo "trabalho") (s 16) (d 13)))
  (assert (atividade (tipo "vestir") (s 11.5) (d 9.7)))

  (retract ?inicio)
)
```

Funções e regras de uso geral

```
(deffunction verifica_idade_mulher_gravida (?idade)
  (if (or (<= ?idade 10) (>= ?idade 40)) then
    (printout t "Idade incompativel" crlf)
    (printout t "Deseja continuar mesmo assim? (s/n) ")
    (bind ?resp (read))
    (if (or (eq ?resp s) (eq ?resp S)) then
      (return true)
    else
      (return false)))
)
```

Regras

```
(defrule regra_crianca_jovem
?paciente <- (or (paciente (tipo "crianca")) (paciente (tipo "jovem")))
?atividade <- (atividade)
?pressao <- (pressao)
=>
(if (= (fact-slot-value ?paciente id) (fact-slot-value ?pressao
id_paciente)) then
  (if (= (fact-slot-value ?pressao atividade) (fact-slot-value
?atividade tipo)) then
    (bind ?s (fact-slot-value ?atividade s))
    (bind ?d (fact-slot-value ?atividade d))

    (if (and (>= (fact-slot-value ?pressao PAC_S) (+ 133 ?s)) (and (> (fact-
slot-value ?pressao MAPA_S) (+ 124 ?s)) (<= (fact- slot-value ?pressao
MAPA_S) (+ 129 ?s)))) then
      (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
    else
      (if (and (and (< (fact-slot-value ?pressao PAC_S) (+ 133 ?s))
(<= (fact-slot-value ?pressao MAPA_S) (+ 124 ?s)))
        (and (< (fact-slot-value ?pressao PAC_D) (+ 86 ?d)) (<= (fact-
slot-value ?pressao MAPA_D) (+ 76 ?d))) ) then
          (modify ?paciente (status "NORMOTENSÃO")))

        (if (and (and (>= (fact-slot-value ?pressao PAC_S) (+ 133 ?s))
(> (fact-slot-value ?pressao MAPA_S) (+ 124 ?s)))
          (and (>= (fact-slot-value ?pressao PAC_D) (+ 86 ?d)) (> (fact-
slot-value ?pressao MAPA_D) (+ 86 ?d))) ) then
            (modify ?paciente (status "HIPERTENSÃO")))

          (if (and (and (>= (fact-slot-value ?pressao PAC_S) (+ 133 ?s))
(<= (fact-slot-value ?pressao MAPA_S) (+ 129 ?s)))
            (and (>= (fact-slot-value ?pressao PAC_D) (+ 86 ?d)) (<= (fact-
slot-value ?pressao MAPA_D) (+ 81 ?d))) ) then
              (modify ?paciente (status "HIPERTENSÃO DO AVENTAL
BRANCO")))

            (if (and (and (< (fact-slot-value ?pressao PAC_S) (+ 133 ?s))
(> (fact-slot-value ?pressao MAPA_S) (+ 129 ?s)))
              (and (< (fact-slot-value ?pressao PAC_D) (+ 86 ?d)) (> (fact-
slot-value ?pressao MAPA_D) (+ 81 ?d))) ) then
                (modify ?paciente (status "HIPERTENSÃO MASCARADA"))) )
          )
        )
      )
    )
  )
)
```

```

(defrule regra_adulto_idoso
  ?pressao <- (pressao)
  ?atividade <- (atividade)
  ?paciente <- (or (paciente (tipo "adulto")) (paciente (tipo "idoso")))
=>
  (if (= (fact-slot-value ?paciente id) (fact-slot-value ?pressao
id_paciente)) then
    (if (= (fact-slot-value ?pressao atividade) (fact-slot-value ?atividade
tipo)) then
      (bind ?s (fact-slot-value ?atividade s))
      (bind ?d (fact-slot-value ?atividade d))

      (if (and (>= (fact-slot-value ?pressao PAC_S) (+ 140 ?s))
        (and (> (fact-slot-value ?pressao MAPA_S) (+ 130 ?s)) (<= (fact-slot-
value ?pressao MAPA_S) (+ 135 ?s)))) then
        (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
      else

        (if (and (and (< (fact-slot-value ?pressao PAC_S) (+ 140 ?s))
          (<= (fact-slot-value ?pressao MAPA_S) (+ 130 ?s)))
          (and (< (fact-slot-value ?pressao PAC_D) (+ 90 ?d)) (<= (fact-
slot-value ?pressao MAPA_D) (+ 80 ?d))) ) then
            (modify ?paciente (status "NORMOTENSÃO")))

          (if (and (and (>= (fact-slot-value ?pressao PAC_S) (+ 140 ?s))
            (> (fact-slot-value ?pressao MAPA_S) (+ 130 ?s))) (and (>=
(fact-slot-value ?pressao PAC_D) (+ 90 ?d)) (> (fact-slot-value
?pressao MAPA_D) (+ 90 ?d))) ) then
              (modify ?paciente (status "HIPERTENSÃO")))

            (if (and (and (>= (fact-slot-value ?pressao PAC_S) (+ 140 ?s))
              (<= (fact-slot-value ?pressao MAPA_S) (+ 135 ?s))) (and (>=
(fact-slot-value ?pressao PAC_D) (+ 90 ?d)) (<= (fact-slot-
value ?pressao MAPA_D) (+ 85 ?d))) ) then
                (modify ?paciente (status "HIPERTENSÃO DO AVENTAL
BRANCO")))

              (if (and (and (< (fact-slot-value ?pressao PAC_S) (+ 140 ?s))
                (> (fact-slot-value ?pressao MAPA_S) (+ 135 ?s))) (and (< (fact-
slot-value ?pressao PAC_D) (+ 90 ?d)) (> (fact-slot-value
?pressao MAPA) (+ 85 ?d))) ) then
                  (modify ?paciente (status "HIPERTENSÃO MASCARADA"))) )
                )
              )
            )

  (defrule regra_mulher_gravida
    ?paciente <- (mulher_gravida (status "HIPERTENSÃO"))
    ?atividade <- (atividade)
    ?pressao <- (pressao)
    =>
    (if (eq (fact-slot-value ?paciente periodo) vigilia) then
      (try
        (if (= (fact-slot-value ?paciente id) (fact-slot-value ?pressao
id_paciente)) then

          (if (= (fact-slot-value ?pressao atividade) (fact-slot-value
?atividade tipo)) then
            (modify ?paciente (status "HIPERTENSÃO MASCARADA")))
          )
        )
      )
    )
  )

```

```

(bind ?s (fact-slot-value ?atividade s))
(bind ?d (fact-slot-value ?atividade d))

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 9) (<=
(fact-slot-value ?paciente semanas_gestacao) 16)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ (+ 115 8)
?s)) (>= (fact-slot-value ?pressao MAPA_S) (+ (- 115 8) ?s))) (and
(<=(fact-slot-value ?pressao MAPA_D) (+ (+ 70 7) ?d)) (>=(fact-
slot-value ?pressao MAPA_D) (+ (- 70 7) ?d)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
else

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 17) (<=
(fact-slot-value ?paciente semanas_gestacao) 24)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ (+ 115 8)
?s)) (>= (fact-slot-value ?pressao MAPA_S) (+ (- 115 8)
?s))) (and(<=(fact-slot-value ?pressao MAPA_D) (+ (+ 69 6) ?d))
(>=(fact-slot-value ?pressao MAPA_D) (+ (- 69 6) ?d)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
else

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 25) (<=
(fact-slot-value ?paciente semanas_gestacao) 32)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ (+ 116 9)
?s)) (>= (fact-slot-value ?pressao MAPA_S) (+ (- 116 9)
?s))) (and(<=(fact-slot-value ?pressao MAPA_D) (+ (+ 70 7) ?d))
(>=(fact-slot-value ?pressao MAPA_D) (+ (- 70 7) ?d)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
else

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 33) (<=
(fact-slot-value ?paciente semanas_gestacao) 40)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ (+ 119 9)
?s)) (>= (fact-slot-value ?pressao MAPA_S) (+ (- 119 9)
?s))) (and(<=(fact-slot-value ?pressao MAPA_D) (+ (+ 74 7) ?d))
(>=(fact-slot-value ?pressao MAPA_D) (+ (- 74 7) ?d)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))))))

catch)
else
(try

(if (= (fact-slot-value ?paciente id) (fact-slot-value ?pressao
id_paciente)) then

```

```

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 9) (<=
(fact-slot-value ?paciente semanas_gestacao) 16)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ 100 7)) (>=
(fact-slot-value ?pressao MAPA_S) (- 100 7))) (and(<=(fact-slot-
value ?pressao MAPA_D) (+ 55 5)) (>=(fact-slot-value ?pressao
MAPA_D) (- 55 5)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
else

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 17) (<=
(fact-slot-value ?paciente semanas_gestacao) 24)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ 99 8)) (>=
(fact-slot-value ?pressao MAPA_S) (- 99 8))) (and(<=(fact-slot-value
?pressao MAPA_D) (+ 54 6)) (>=(fact-slot-value ?pressao MAPA_D) (-
54 6)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
else

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 25) (<=
(fact-slot-value ?paciente semanas_gestacao) 32)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ 101 8)) (>=
(fact-slot-value ?pressao MAPA_S) (- 101 8))) (and(<=(fact-slot-
value ?pressao MAPA_D) (+ 55 6)) (>=(fact-slot-value ?pressao
MAPA_D) (- 55 6)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))
else

(if (and (>= (fact-slot-value ?paciente semanas_gestacao) 33) (<=
(fact-slot-value ?paciente semanas_gestacao) 40)) then

  (if (and (and (<=(fact-slot-value ?pressao MAPA_S) (+ 108 8)) (>=
(fact-slot-value ?pressao MAPA_S) (- 108 8))) (and(<=(fact-slot-
value ?pressao MAPA_D) (+ 58 7)) (>=(fact-slot-value ?pressao
MAPA_D) (- 58 7)))) then
    (modify ?paciente (status "NORMOTENSÃO"))
  else
    (modify ?paciente (status "HIPERTENSÃO ou HIPERTENSÃO DO
AVENTAL BRANCO")))))))
catch
))
)
(run)

```

2. Regra de avaliação dos riscos de hipertensão do paciente

```

(deftemplate paciente
  (slot id)
  (slot nome)
  (slot idade)
  (slot tipo)
  (multislot fatores_risco (type string))
  (slot status))

(deftemplate pressao
  (slot id_paciente)
  (slot PAC_S)
  (slot PAC_D))

(deftemplate alarme
  (slot id_paciente)
  (slot mensagem))

(defun verifica_fator_risco (?lista ?fator)
  (bind ?i 1)
  (bind ?qte (length$ ?lista))
  (while (<= ?i ?qte)
    (bind ?lista_fator (nth$ ?i ?lista))
    (if (eq ?lista_fator ?fator) then
      (return true))
    (bind ?i (+ ?i 1)))
  )

(defrule verifica_status_paciente
?paciente <- (or (paciente (tipo "adulto")) (paciente (tipo "idoso")))
?pressao <- (pressao)
=>
  (if (= (fact-slot-value ?paciente id) (fact-slot-value ?pressao
id_paciente)) then
    (if (and (< (fact-slot-value ?pressao PAC_S) 120) (< (fact-slot-value
?pressao PAC_D) 80)) then
      (modify ?paciente (status "OTIMA"))
    else
      (if (and (< (fact-slot-value ?pressao PAC_S) 130) (< (fact-slot-value
?pressao PAC_D) 85)) then
        (modify ?paciente (status "NORMAL"))
      else (if (and (and(>= (fact-slot-value ?pressao PAC_S) 130) (<= (fact-
slot-value ?pressao PAC_S) 139))
        (and(>= (fact-slot-value ?pressao PAC_D) 85) (<= (fact-
slot-value ?pressao PAC_D) 89))) then
          (modify ?paciente (status "LIMITROFE"))
        else (if (and (and(>= (fact-slot-value ?pressao PAC_S) 140) (<=
(fact-slot-value ?pressao PAC_S) 159))
          (and(>= (fact-slot-value ?pressao PAC_D) 90) (<=
(fact-slot-value ?pressao PAC_D) 99))) then
            (modify ?paciente (status "HIPERTENSAO_N1"))
          else (if (and (and(>= (fact-slot-value ?pressao PAC_S)
160) (<= (fact-slot-value ?pressao PAC_S) 179))
            (and(>= (fact-slot-value ?pressao PAC_D)
100) (<= (fact-slot-value ?pressao PAC_D) 109))) then
              (modify ?paciente (status "HIPERTENSAO_N2"))
            else (if (and (>= (fact-slot-value ?pressao PAC_S)
180) (>= (fact-slot-value ?pressao PAC_D) 110)) then

```



```

        (modify ?paciente (status "HIPERTENSAO_N3"))))))))
    )
)

(defrule analisa_risco_normal
  ?paciente <- (paciente (status "NORMAL"))
  =>
  (bind ?qte_fatores (length$ (fact-slot-value ?paciente fatores_risco)))
  (if (= ?qte_fatores 0) then
    (assert (alarme (id_paciente (fact-slot-value ?paciente id))
(mensagem "Sem Risco Adicional"))))
    else (if (verifica_fator_risco (fact-slot-value ?paciente
fatores_risco) "doencas_cardiovasculares") then
      (assert (alarme (id_paciente (fact-slot-value ?paciente
id)) (mensagem "Risco Alto"))))
      else (if (or (>= ?qte_fatores 3) (or (verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "lesao_de_orgaos")

(verifica_fator_risco (fact-slot-value ?paciente fatores_risco)
"diabetes_melito"))) then
      (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Médio"))))
      else (if (and (> ?qte_fatores 0) (<= ?qte_fatores 2)) then
        (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Baixo"))))
      ))))
)

(defrule analisa_risco_limitrofe
  ?paciente <- (paciente (status "LIMITROFE"))
  =>
  (bind ?fatores (length$ (fact-slot-value ?paciente fatores_risco)))
  (if(= ?fatores 0) then
    (assert (alarme (id_paciente (fact-slot-value ?paciente id))
(mensagem "Sem Risco Adicional"))))
    else (if (verifica_fator_risco (fact-slot-value ?paciente
fatores_risco) "doencas_cardiovasculares") then
      (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Muito Alto"))))
      else (if (or (>= ?fatores 3) (or (verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "lesao_de_orgaos")

(verifica_fator_risco (fact-slot-value ?paciente fatores_risco)
"diabetes_melito"))) then
      (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Alto"))))
      else (if (and (> ?fatores 0) (<= ?fatores 2)) then
        (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Baixo"))))))))
)

(defrule analisa_risco_hipertenso_n1
  ?paciente <- (paciente (status "HIPERTENSAO_N1"))
  =>
  (bind ?fatores (length$ (fact-slot-value ?paciente fatores_risco)))
  (if(= ?fatores 0) then
    (assert (alarme (id_paciente (fact-slot-value ?paciente id))
(mensagem "Risco Baixo"))))
    else (if (verifica_fator_risco (fact-slot-value ?paciente
fatores_risco) "doencas_cardiovasculares") then

```

```

        (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Muito Alto"))))
        else (if (or (>= ?fatores 3) (or (verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "lesao_de_orgaos")
(verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "diabetes_melito")))) then
        (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Alto"))))
        else (if (and (> ?fatores 0) (<= ?fatores 2)) then
        (assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Médio"))))))))
)

(defrule analisa_risco_hipertenso_n2
?paciente <- (paciente (status "HIPERTENSAO_N2"))
=>
(bind ?fatores (length$ (fact-slot-value ?paciente fatores_risco)))
(if(= ?fatores 0) then
(assert (alarme (id_paciente (fact-slot-value ?paciente id))
(mensagem "Risco Médio")))
else (if (verifica_fator_risco (fact-slot-value ?paciente
fatores_risco) "doencas_cardiovasculares") then
(assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Muito Alto")))
else (if (or (>= ?fatores 3) (or (verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "lesao_de_orgaos")
(verifica_fator_risco (fact-slot-value ?paciente fatores_risco)
"diabetes_melito")))) then
(assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Alto")))
else (if (and (> ?fatores 0) (<= ?fatores 2)) then
(assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Médio"))))))))
)

(defrule analisa_risco_hipertenso_n3
?paciente <- (paciente (status "HIPERTENSAO_N3"))
=>
(bind ?fatores (length$ (fact-slot-value ?paciente fatores_risco)))
(if(= ?fatores 0) then
(assert (alarme (id_paciente (fact-slot-value ?paciente id))
(mensagem "Risco Alto")))
else (if (verifica_fator_risco (fact-slot-value ?paciente
fatores_risco) "doencas_cardiovasculares") then
(assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Muito Alto")))
else (if (or (>= ?fatores 3) (or (verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "lesao_de_orgaos")
(verifica_fator_risco
(fact-slot-value ?paciente fatores_risco) "diabetes_melito")))) then
(assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Muito Alto")))
else (if (and (> ?fatores 0) (<= ?fatores 2)) then
(assert (alarme (id_paciente (fact-slot-value
?paciente id)) (mensagem "Risco Muito Alto"))))))))
)
(run)

```