

UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

ALEXSANDRO BESERRA BASTOS

**Uma Abordagem Ontológica Baseada  
em Informações de Contexto para  
Representação de Conhecimento de  
Monitoramento de Sinais Vitais  
Humanos**

Goiânia  
2013

ALEXSANDRO BESERRA BASTOS

# **Uma Abordagem Ontológica Baseada em Informações de Contexto para Representação de Conhecimento de Monitoramento de Sinais Vitais Humanos**

Dissertação apresentada ao Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

**Área de concentração:** Ciência da Computação.

**Orientador:** Prof. Dr. Renato de Freitas Bulcão Neto

**Co-Orientador:** Prof. Dr. Iwens Gervásio Sene Júnior

Goiânia  
2013

ALEXSANDRO BESERRA BASTOS

# **Uma Abordagem Ontológica Baseada em Informações de Contexto para Representação de Conhecimento de Monitoramento de Sinais Vitais Humanos**

Dissertação defendida no Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 21 de Outubro de 2013, pela Banca Examinadora constituída pelos professores:

---

**Prof. Dr. Renato de Freitas Bulcão Neto**  
Instituto de Informática – UFG  
Presidente da Banca

---

**Prof. Dr. Iwens Gervásio Sene Júnior**  
Instituto de Informática – UFG

---

**Prof. Dr. Sérgio Teixeira de Carvalho**  
Instituto de Informática – UFG

---

**Profa. Dra. Renata Maria Porto Vanni**  
IFSP, Campus Araraquara

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

### **Alexsandro Beserra Bastos**

Graduou-se em Processamento de Dados no Centro Universitário de Goiás - Uni-Anhanguera. Especialização em "Orientação a Objetos e Internet" no Centro Universitário de Goiás - Uni-Anhanguera. Especialização em "Desenvolvimento de Aplicações Web com Interfaces Ricas" pelo Instituto de Informática da UFG. Durante o mestrado participou do projeto "Prontuário Eletrônico do Paciente - PEP", uma parceria entre a Universidade Federal de Goiás com a Dell Computadores. Atualmente, é Analista de Tecnologia da Informação no Hospital das Clínicas da UFG.

À minha família.

---

## Agradecimentos

---

Primeiramente, agradeço a Deus que me protege, ilumina e me guia em todos os meus passos.

Agradeço à minha esposa Romana e aos meus filhos Pedro e Ana Júlia, por me oferecerem o seu amor, carinho e incentivo e por todos os momentos felizes que me proporcionam. Pela compreensão, dos finais de semana em que não passeamos e da viagem de férias que ainda não fizemos em razão do envolvimento com os afazeres das atividades do mestrado. Mas, que no fim, será gratificante para todos nós.

Aos meus pais, por estarem presentes e sempre apoiando nos momentos desafiadores da minha vida. Pelo carinho e lanches que minha mãe me servia nos momentos de estudo em sua casa.

Aos meus irmãos, pelo incentivo, colaboração e conselhos que me impulsionavam a ultrapassar os obstáculos da vida. Também, pelas diversas vezes que estiveram presentes na vida dos meus filhos para acompanhá-los em passeios, atividades escolares, viagens e aconselhamentos.

Às minhas cunhadas e cunhados pela amizade, colaboração e incentivo nos diversos momentos da minha vida.

Ao meu sogro e sogra, pela sabedoria de suas palavras e de seu apoio.

Aos meus sobrinhos, pelo incentivo, amor, carinho e companheirismo.

Ao meu orientador, Prof. Renato, pela confiança em me aceitar como seu orientando. Agradeço por ter, sempre que possível, me acompanhado nas entrevistas realizadas no HC/UFG e que se tornaram produtivas na elaboração deste trabalho. Pelos ensinamentos e orientações dadas que foram de grande importância na realização deste trabalho e no crescimento do meu saber. Além disso, pelos conselhos de incentivo que me encorajaram na realização deste trabalho. Se não fiz mais neste trabalho não foi por falta de seu apoio.

Ao meu co-orientador, Prof. Iwens, pelo incentivo e observações feitas nos encontros dos seminários do grupo de pesquisa que, com certeza, ajudaram na elaboração deste trabalho.

Aos médicos e enfermeiros do HC/UFG, em especial à Dra. Cacilda, por terem reservados algumas horas de sua apertada agenda de compromissos para nos atender nas

entrevistas para a elaboração deste trabalho.

Ao grupo de estudo formado neste mestrado, denominado "Grupo Amendoim", em especial aos amigos Adailton, André e Leonardo, que foi de grande importância na superação dos estudos das disciplinas.

Ao Márcio Vinícius e ao Guilherme Marques que me deram apoio no desenvolvimento das consultas da linguagem SPARQL para validar as questões de competência elaboradas para este trabalho.

Agradeço a Mariana Soller pelo apoio no uso dos comandos do Latex para a escrita desta dissertação.

Aos amigos da T.I do Hospital das Clínicas da UFG, em especial ao meu amigo Lucymar da Cunha, coordenador de T.I. do HC/UFG, que me apoiou para fazer o mestrado.

À Diretoria do HC/UFG, em especial ao Diretor Prof. José Garcia, que concedeu a minha liberação para cursar o mestrado.

Enfim, a todas as pessoas que diretamente ou indiretamente me apoiaram nesta empreitada.

*"Eu aprendi que todos querem viver no topo da montanha, mas toda felicidade e crescimento ocorre quando você a está escalando;"*

***William Shakespeare,***

---

## Resumo

---

Bastos, Alexsandro Beserra. **Uma Abordagem Ontológica Baseada em Informações de Contexto para Representação de Conhecimento de Monitoramento de Sinais Vitais Humanos**. Goiânia, 2013. 128p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

O monitoramento de sinais vitais de pacientes em unidades de terapia intensiva (UTI) é uma atividade que faz parte da rotina de vários profissionais da área da Saúde, envolvendo médicos, enfermeiros, técnicos e auxiliares de enfermagem. Em várias UTIs, o acompanhamento e o registro dos monitoramentos de sinais vitais são realizados de forma manual e em instantes de tempo pré-definidos. Nesse cenário, os registros das medições dos sinais vitais são escritos em formulários pré-impressos, e quando um profissional da saúde deseja obter alguma informação sobre o estado clínico de um paciente, ele tem que recorrer a esses formulários. Além disso, quando ocorre alguma anormalidade no monitoramento do sinal vital do paciente, um monitor multiparamétrico emite alarmes sonoros e, dependendo do fluxo de trabalho dentro da UTI, esse alarme pode não ser percebido prontamente pela equipe médica. Nesse sentido, este trabalho propõe um modelo de representação do conhecimento relativo ao monitoramento de sinais vitais de pacientes em UTIs. O modelo proposto utiliza o potencial de expressividade e formalidade de ontologias, regras e tecnologias correlatas da Web Semântica, o que promove a compreensão consensual, o compartilhamento e a reutilização de informações de sinais vitais de pacientes. Com esse modelo, espera-se também viabilizar o desenvolvimento de aplicações sensíveis a contexto que envolvem o monitoramento de sinais vitais humanos, com armazenamento, consultas e produção de alarmes em casos de anormalidades detectadas nas medições desses sinais.

### Palavras-chave

Ontologia, Monitoramento, Sinal Vital, OWL, SWRL, SeCoM, VSO, Informação de Contexto

---

## Abstract

---

Bastos, Alexsandro Beserra. **An Ontological Approach Based on Context Information for Knowledge Representation and Monitoring of Human Vital Signs**. Goiânia, 2013. 128p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Monitoring vital signs in intensive care units (ICU) is an everyday activity of various health professionals, including doctors, nurses, technicians and nursing assistants. In most ICUs, monitoring and recording vital signs are performed in a manual fashion and in pre-defined time instants. The records of measurements of vital signs in ICUs are generally written on preprinted forms, and a health professional has to re-sort those forms when he wants to get some information about the clinical state of a patient. Besides, when an abnormal measurement of vital sign is detected, a multiparameter monitor triggers audible alarms, and that alarm may not be prompted detected by the medical staff, depending on the workflow within an ICU. In that sense, this work proposes a knowledge representation model of monitoring of vital signs of patients in ICUs. The model proposed exploits the expressiveness and the formality of ontologies, rules and semantic web technologies. This promotes the consensual comprehension, the sharing and the reuse of vital signs of patients. The aim is to develop context-aware applications for monitoring human vital signs, including the storage, query support and semantic alarms triggering.

### Keywords

Ontology, Monitoring, Vital Sign, OWL, SWRL, SeCoM, VSO, Context Information

---

# Sumário

---

Lista de Figuras	14
Lista de Tabelas	15
Abreviações	17
<b>1</b> Introdução	<b>18</b>
1.1 Contexto e Motivação	18
1.2 Objetivos	19
1.3 Metodologia	20
1.4 Organização do Trabalho	21
<b>2</b> Fundamentação Teórica	<b>22</b>
2.1 Ontologia	22
2.1.1 Definições	22
2.1.2 Vantagens da Ontologia	24
2.1.3 Construção de Ontologias	24
2.2 A linguagem OWL	26
2.2.1 OWL 2	28
2.3 SWRL	29
2.3.1 Estrutura das regras em SWRL	29
2.4 Computação sensível ao contexto	33
2.5 Considerações finais	34
<b>3</b> Tecnologias Relacionadas	<b>35</b>
3.1 Modelo SeCoM	35
3.1.1 Ontologia Time	36
3.1.2 Ontologia Temporal Event	37
3.1.3 Ontologia Spatial	38
3.1.4 Ontologia Spatial Event	39
3.1.5 Ontologia Actor	40
3.1.6 Ontologia Activity	41
3.2 Vital Sign Ontology - VSO	42
3.2.1 Ontologias importadas para VSO	44
3.2.2 A representação dos sinais vitais na VSO	44
3.2.3 Representação da localização anatômica do corpo humano em VSO	45
3.2.4 Representação dos valores de dados de medição dos sinais vitais em VSO	46
3.3 Editor de ontologias Protégé-OWL	47
3.4 Raciocinador Pellet	47

3.5	Considerações finais	47
<b>4</b>	<b>Modelo Formal para Monitoramento de Sinais Vitais Humanos</b>	<b>49</b>
4.1	Contextualização	49
4.1.1	Cenário de estudo	49
4.1.2	Delimitação do escopo	50
4.2	Ontologia MSVH	55
4.2.1	Descrição da classe vso:'vital sign'	58
4.2.2	Descrição da classe ParameterizedVitalSign	59
4.2.3	Descrição da classe vso:'Anatomical entity'	60
4.2.4	Descrição da classe vso:'measurement datum'	62
	Descrição da subclasse 'blood pressure measurement datum'	63
	Descrição da subclasse 'body temperature measurement datum'	64
	Descrição da subclasse 'respiratory rate measurement datum'	65
	Descrição da subclasse 'pulse rate measurement datum'	66
	Descrição da subclasse 'oxygen saturation measurement datum'	67
4.2.5	Descrição da classe Alarm	67
	Descrição da subclasse BloodPressureAlarm	69
	Descrição da subclasse BodyTemperatureAlarm	70
	Descrição da subclasse RespiratoryRateAlarm	71
	Descrição da subclasse PulseRateAlarm	71
	Descrição da subclasse OxygenSaturationAlarm	72
4.2.6	Descrição da classe MonitoringVitalSign	72
4.3	Validação do modelo ontológico MSVH	74
4.4	Considerações finais	81
<b>5</b>	<b>Trabalhos Relacionados</b>	<b>82</b>
5.1	Sistema Inteligente de Monitoramento e Envio de Alertas de Pacientes	82
5.2	Arcabouço Health Support in Aware and Ubiquitous Domestic Environments	83
5.3	TeleCardio	84
5.4	Projeto Kamer	85
5.5	Análise dos trabalhos	86
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>88</b>
6.1	Contribuições	88
6.2	Limitações do modelo ontológico MSVH	89
6.3	Trabalhos Futuros	89
	Referências Bibliográficas	<b>91</b>
<b>A</b>	<b>Questões de Competência</b>	<b>95</b>
<b>B</b>	<b>Alarmes Expressos como Regras SWRL</b>	<b>111</b>
B.1	Alarmes para Pressão Arterial	111
B.1.1	Pressão Arterial utilizando a tabela populacional	111
	Pressão Diastólica em categoria diferente, ou seja, alteração da classificação em razão da pressão diastólica ser a de maior valor na classificação	113

	Pressão Sistólica em categoria diferente, ou seja, alteração da classificação em razão da pressão sistólica ser a de maior valor na classificação	114
B.1.2	Pressão Arterial utilizando o valor individualizado	115
	Utilizando valores de medidas das pressões sistólicas e diastólica maiores que os valores máximos individualizados para as pressões sistólica e diastólica	116
	Utilizando valores de medidas das pressões sistólicas e diastólica menores que os valores mínimos individualizados para as pressões sistólica e diastólica – Sem disparar alarme, somente classificado-os	117
	Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica	118
	Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica	119
	Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica menor que o valor individualizado mínimo da diastólica	120
	Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica menor que o valor individualizado mínimo da sistólica	121
	Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica e de maior valor na classificação	122
	Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica e de maior valor na classificação	123
B.2	Alarmes para Temperatura Corpórea	124
B.3	Alarmes para Frequência de Pulso	125
B.3.1	Frequência de Pulso utilizando a tabela populacional	125
B.3.2	Frequência de Pulso utilizando o valor individualizado	125
	Utilizando o valor dentro de cada classificação. Disparando alarme, caso ultrapassem os valores parametrizados para a medição.	125
B.4	Alarmes para Frequência Respiratória	126
B.4.1	Frequência Respiratória utilizando a tabela populacional	126
B.4.2	Frequência Respiratória utilizando o valor individualizado	126
	Utilizando valor individualizado dentro de cada classificação. Disparando alarme, caso ultrapassem os valores parametrizados para a medição.	127

B.5	Alarmes para Saturação de Oxigênio	128
B.5.1	Saturação de Oxigênio utilizando a tabela populacional	128
B.5.2	Saturação de Oxigênio utilizando o valor individualizado	128
	Utilizando valor individualizado dentro de cada classificação. Disparando alarme, caso ultrapassem os valores parametrizados para a medição.	128

---

## Lista de Figuras

---

2.1	Exemplo de uma regra SWRL	32
3.1	Modelo SeCoM [9]	36
3.2	Ontologia Time [9]	37
3.3	Ontologia Temporal Event [9]	38
3.4	Ontologia Spatial [9]	39
3.5	Ontologia Spatial Event [9]	40
3.6	Ontologia Actor [9]	41
3.7	Ontologia Activity [9]	42
3.8	Ontologia VSO [17]	43
4.1	Ficha individual de acompanhamento de pacientes na UTI do HC-UFG (trecho) [5]	50
4.2	Classificação da temperatura corpórea [29]	53
4.3	Classificação da frequência de pulso [30]	53
4.4	Classificação da frequência respiratória [30]	54
4.5	Classificação da saturação de oxigênio [30]	54
4.6	Classificação da pressão arterial de acordo com a medida casual no consultório (> 18 anos) [13]	55
4.7	Principais conceitos da ontologia MSVH	56
4.8	Subclasses da classe vso:'vital sign'	59
4.9	Subclasses da classe ParameterizedVitalSign	60
4.10	Subclasses da classe vso:'Anatomical entity'	61
4.11	Subclasses da classe vso:'measurement datum'	62
4.12	Subclasses de 'blood pressure measurement datum'	64
4.13	Subclasses de 'body temperature measurement datum'	65
4.14	Subclasses de 'respiratory rate measurement datum'	66
4.15	Subclasses de 'pulse rate measurement datum'	66
4.16	Subclasses de 'oxygen saturation measurement datum'	67
4.17	Subclasses da classe Alarm	68
4.18	Subclasses de BloodPressureAlarm	69
4.19	Subclasses de BodyTemperatureAlarm	70
4.20	Subclasses de RespiratoryRateAlarm	71
4.21	Subclasses de PulseRateAlarm	71
4.22	Subclasse de OxygenSaturationAlarm	72
4.23	Subclasses de MonitoringVitalSign	73

---

## Lista de Tabelas

---

2.1	Tipos de átomos da regra SWRL	30
B.1	Pressão Arterial - valor populacional	112
B.2	Pressão Arterial – Pressão Diastólica em categoria diferente	114
B.3	Pressão Arterial – Pressão Sistólica em categoria diferente	114
B.4	Pressão Arterial – valor individualizado	115
B.5	Pressão Arterial – valores de medidas das pressões sistólicas e diastólica maiores que os valores máximos individualizados para as pressões sistólica e diastólica	116
B.6	Pressão Arterial – valores de medidas das pressões sistólicas e diastólica menores que os valores mínimos individualizados para as pressões sistólica e diastólica	117
B.7	Pressão Arterial – valor da medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica	118
B.8	Pressão Arterial – valor da medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica	119
B.9	Pressão Arterial – valor da medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica menor que o valor individualizado mínimo da diastólica	120
B.10	Pressão Arterial – valor da medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica menor que o valor individualizado mínimo da sistólica	121
B.11	Pressão Arterial – valor da medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica e de maior valor na classificação	122
B.12	Pressão Arterial – valor da medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica e de maior valor na classificação	123
B.13	Temperatura Corpórea Axilar	124
B.14	Frequência de Pulso - valor populacional	125
B.15	Frequência de Pulso - valor individualizado sem disparo de alarme	125
B.16	Frequência de Pulso - valor individualizado com disparo de alarme	125
B.17	Frequência respiratória – valor populacional	126
B.18	Frequência respiratória – valor individualizado sem disparar alarme	126

B.19	Frequência respiratória – valor individualizado com disparo de alarme	127
B.20	Saturação de Oxigênio (SpO2) – valor populacional	128
B.21	Saturação de Oxigênio (SpO2) – valor individualizado sem disparo de alarme	128
B.22	Saturação de Oxigênio (SpO2) – valor individualizado com disparo de alarme	128

---

## Abreviações

---

<b>°C</b>	Grau Celsius
<b>BPM</b>	Batimentos Por Minuto
<b>IA</b>	Inteligência Artificial
<b>IBGE</b>	Instituto Brasileiro de Geografia e Estatística
<b>mmHg</b>	Milímetros de Mercúrio
<b>MRPM</b>	Movimento de Respiração Por Minuto
<b>MSVH</b>	Monitoramento de Sinais Vitais Humanos
<b>OWL</b>	Web Ontology Language
<b>PNUD</b>	Programa das Nações Unidas para o Desenvolvimento no Brasil
<b>RNA</b>	Redes Neurais Artificiais
<b>RSSFCH</b>	Redes de Sensores Sem Fio do Corpo Humano
<b>SeCoM</b>	Semantic Context Model
<b>SWRL</b>	Semantic Web Rule Language
<b>UTI</b>	Unidade de Terapia Intensiva
<b>VSO</b>	Vital Sign Ontology
<b>W3C</b>	World Wide Web Consortium

---

## Introdução

---

Em várias Unidades de Terapia Intensiva (UTIs) o acompanhamento e registro dos monitoramentos de sinais vitais são realizados de forma manual e em instantes de tempo pré-definidos.

Para os registros das medições dos sinais vitais, a equipe de saúde da UTI lê os dados apresentados em um monitor multiparamétrico e os registra, geralmente, em formulários pré-impressos contendo informações relacionadas à identificação do paciente e a quais sinais vitais estão sendo monitorados.

Quando algum profissional de saúde deseja obter alguma informação sobre o estado clínico de um paciente, ele tem que recorrer a estes formulários preenchidos manualmente.

Além disso, quando ocorre alguma anormalidade na medição do sinal vital do paciente, o monitor multiparamétrico dispara algum tipo de alarme, sonoro ou luminoso, para alertar a equipe de saúde da UTI, que pode não perceber alerta em um tempo curto. Pode ocorrer uma demora no atendimento ao paciente, caso seja necessário localizar o médico de plantão para realizar alguma intervenção.

Neste capítulo são apresentados o contexto e a motivação que levaram ao desenvolvimento deste trabalho de pesquisa, um modelo do domínio de conhecimento na área da saúde para o monitoramento de sinais vitais humanos, na qual esta pesquisa está inserida no contexto do Projeto CNPq Universal n. 481402/2011-0.

Este capítulo está organizado da seguinte maneira: a Seção 1.1 apresenta o contexto e a motivação da elaboração deste trabalho. A Seção 1.2 apresenta o objetivo a que se propõe este trabalho. A Seção 1.3 apresenta a metodologia do trabalho e por fim, a Seção 1.4 apresenta a estrutura organizacional deste trabalho.

### 1.1 Contexto e Motivação

O monitoramento de sinais vitais de pacientes em Unidades de Terapia Intensiva (UTI) é uma atividade que faz parte da rotina de vários profissionais da área da Saúde, envolvendo médicos, enfermeiros, técnicos e auxiliares de enfermagem. Para um mo-

monitoramento eficiente, além do pessoal envolvido, são necessários equipamentos de alta tecnologia e, por conseguinte, de alto custo, o que pode inviabilizar a existência de uma UTI em hospitais com poucos recursos.

A importância do monitoramento de sinais vitais de pacientes crônicos dentro de UTIs é no sentido de acompanhar o estado de evolução ou deterioração de sua saúde no sentido de conduzir procedimentos adequados para o bem estar do paciente.

Além disso, é comum que pacientes, mesmo fora do hospital, necessitem de cuidados médicos, como o monitoramento de seus sinais vitais, exigindo maquinário e pessoal dedicados. Soma-se a isso, o crescente aumento da fatia da população no Brasil com mais de 65 anos, em torno de 14 milhões de idosos (IBGE, 2010), o que demandará ainda mais investimentos na infraestrutura de atendimento à saúde da população brasileira.

Recentemente, foi lançado o Atlas do Desenvolvimento Humano no Brasil (PNUD, 2013), o qual ressalta o aumento da expectativa de vida do brasileiro, que cresceu no país 14% (9,2 anos) entre 1991 e 2010, o que exigirá o planejamento de políticas públicas adequadas a essa nova população no País.

Em função dos fatores supracitados e de outros fatores correlatos, o atendimento domiciliar, ou *home care*, tem ganhado força em vários países no mundo, inclusive no Brasil, conforme descrito por Bastiani et al [4]. Por essa razão, várias tecnologias têm surgido para apoiar essa modalidade de atendimento a pacientes, como as redes de sensores sem fio do corpo humano (RSSFCH), uma rede de sensores onde cada nó sensor mede um fenômeno (ex: frequência cardíaca, temperatura corpórea) e retransmite esses dados a um observador, conforme descrito por Rodrigues-Filho et al [32].

Nesse ínterim, o acesso remoto às informações adquiridas de uma RSSFCH pode ser interessante para o médico fazer acompanhamento contínuo e em tempo real de um paciente, e receber alertas em caso de distúrbios detectados criados por meio de comparação com valores pré-definidos em relação a uma população, ou para cada paciente tomado individualmente.

Para desenvolver um sistema computacional para esse propósito, é salutar que se conheça a atividade de monitoramento de sinais vitais, tomando como referência as tarefas realizadas por profissionais de saúde, bem como o funcionamento de aparelhos utilizados para fins de monitoramento em uma UTI.

## 1.2 Objetivos

O objetivo deste trabalho é modelar o domínio do conhecimento sobre monitoramento de sinais vitais humanos para que sistemas computacionais possam compartilhar deste conhecimento modelado para oferecer serviços e informações aos seus usuários.

Neste sentido, o propósito deste trabalho de pesquisa envolve a representação de conhecimento relativo ao monitoramento de sinais vitais de pacientes em UTIs, conhecimento este modelado utilizando o potencial de expressividade e formalidade de ontologias, regras e tecnologias correlatas da Web Semântica, conforme descrito por Gruber [18], Bao et al [3] e Horrocks et al [22]. A representação deste conhecimento tem a sua estrutura baseada no dimensionamento de informações de contexto, o que permite que sistemas baseados no conhecimento poderão se beneficiar da modelagem desse domínio de monitoramento de sinais vitais humanos para apoiar equipes de saúde na sua rotina diária de assistência à saúde do paciente em um ambiente sensível ao contexto.

## 1.3 Metodologia

Considerando os objetivos supracitados, foi adotada para o desenvolvimento da ontologia de monitoramento de sinais vitais humanos a metodologia *Ontology development 101* de Noy e McGuinness [25], que trata do desenvolvimento de ontologias através da adoção de sete passos: (a) determinar o domínio e escopo da ontologia, (b) considerar o reuso de ontologias existentes, (c) enumerar termos importantes na ontologia, (d) definir as classes e a hierarquia das classes, (e) Definir as propriedades das classes, (f) Definir as facetas dos *slots* e por fim, (g) criar instâncias. Para cada passo da metodologia foram realizadas as seguintes tarefas:

1. Determinar o domínio e escopo da ontologia: para desenvolver uma ontologia é necessário ter o conhecimento do domínio que será modelado. Portanto, para obter o conhecimento necessário foram realizadas entrevistas, utilizando questionários, com médicos e enfermeiros intensivistas do Hospital das Clínicas da UFG, onde foram coletadas informações importantes que ajudaram a entender os conceitos e as suas relações, bem como, o entendimento sobre quais sinais são monitorados. Além disso, conhecemos os equipamentos que realizam o monitoramento dos pacientes na UTI e a forma em que são registradas as informações das medições pela equipe de saúde. Também, foram indicadas, pela equipe de saúde, referências da literatura médica e de enfermagem para se obter os valores de referência dos sinais vitais humanos. Seguindo a orientação da metodologia para a definição do escopo, foi criada uma lista com mais de 30 questões de competências que a ontologia deve responder.
2. Considerar o reuso de ontologias existentes: seguindo este passo foi reutilizada o modelo SeCoM [9], que é um modelo genérico para aplicações sensíveis ao contexto, a ontologia Activity [9], que descreve atividades como eventos espaço-temporais em um ambiente de computação sensível ao contexto, que por sua vez, importa as demais ontologias do modelo SeCoM, que são as ontologias Actor

(Identidade), Time (Tempo), Time Event (Evento Temporal), Spatial (localização), Spatial Event (evento espacial). Também foi reutilizada a ontologia VSO [17], que é uma ontologia que além de descrever os sinais vitais que são consensuais na área de saúde, descreve a estrutura anatômica humana onde ocorrem as medições dos sinais vitais.

3. Enumerar termos importantes na ontologia: através das entrevistas e da literatura médica consultada foi possível elaborar uma lista com os termos relevantes ao domínio de monitoramento de sinais vitais humanos.
4. Definir as classes e a hierarquia das classes: a partir desta lista de termos foi possível estabelecer as classes e a sua hierarquia.
5. Definir as propriedades e as relações entre as classes definidas no passo anterior.
6. Criar instâncias: para a validação do modelo ontológico.

Para a realização dos passos 4 a 6 foi utilizada a ferramenta de edição e construção de ontologias OWL, denominada Protégé-OWL [31]. O editor de ontologias Protégé-OWL é apresentado na seção 3.3 deste trabalho.

Tendo em vista a dificuldade em se obter dados reais do monitoramento de sinais vitais de pacientes internados em UTI foram criados dados fictícios dentro do modelo ontológico proposto com o objetivo de realizar inferências sobre estes dados, por meio do raciocinador Pellet que é apresentado na seção 3.4.

## 1.4 Organização do Trabalho

No capítulo 2 é apresentada a fundamentação teórica que trata sobre a representação do conhecimento na forma de ontologias, apresentando dentre outras linguagens, a linguagem de construção de ontologias OWL e a sua nova versão OWL 2 que apoiam a infraestrutura da Web Semântica. Também, aborda o uso de regras SWRL para complementar a expressividade de modelos do conhecimento.

No capítulo 3 são apresentadas as tecnologias relacionadas: o modelo de informações de contexto SeCoM descrito por Bulcão-Neto [9], a VSO descrito por Goldfain [17], o editor de ontologias Protégé-OWL apresentado em Protégé [31] e o raciocinador Pellet descrito por Sirin et al [35], que apoiaram na construção da ontologia de monitoramento de sinais vitais humanos.

No capítulo 4 é apresentada a estrutura da ontologia de monitoramento de sinais vitais humanos representada por suas classes, relações e atributos.

No capítulo 5 são apresentadas as limitações do trabalho, conclusão e trabalhos futuros.

---

## Fundamentação Teórica

---

Neste capítulo abordaremos conceitos básicos sobre a representação do conhecimento por meio de ontologias. São apresentadas a linguagem para escrita de ontologias, OWL, e a linguagem de regras SWRL que adiciona mais expressividade às ontologias. No final deste capítulo são apresentadas as considerações finais que relacionam os conceitos discutidos neste capítulo com o trabalho proposto.

### 2.1 Ontologia

#### 2.1.1 Definições

No início dos anos 80, pesquisadores em Inteligência Artificial (IA) e especialmente na representação do conhecimento entenderam que o trabalho realizado com ontologias foi importante no essencial processo de descrever o mundo para que sistemas inteligentes pudessem raciocinar e agir sobre esse mundo, relatado em Welty e Guarino [41]. Sendo assim, a ontologia é uma das técnicas amplamente utilizada para representação do conhecimento e a qual permite o compartilhamento e o reúso do conhecimento formalmente representado de algum domínio de conhecimento. O termo “Ontologia” vem do grego e seu significado na filosofia é sobre o estudo do ser. A palavra ontologia foi emprestada para a computação na qual adquiriu um sentido diferente do usado na filosofia.

O termo Ontologia ficou restrito por muito tempo na esfera filosófica, e agora tem ganhado uma importância específica nos campos de Inteligência Artificial, Linguística Computacional e Teoria de Banco de Dados, como relatado em Guarino [20].

Ontologias permitem a criação de um entendimento comum de um domínio específico que possa ser compartilhado entre pessoas e aplicações de forma a não gerar ambiguidades.

Com o objetivo de facilitar o reúso e o compartilhamento do conhecimento, a área de IA tem lançando mão do desenvolvimento de ontologias, definindo vocabulários comuns para que o conhecimento comum possa ser compartilhado entre agentes (pessoas e sistemas).

São várias as definições dadas ao termo pelos vários pesquisadores da área da computação, mas uma das definições mais aceitas pela comunidade de inteligência artificial é a dada por Gruber [19], no qual “ontologia é uma especificação explícita de uma conceitualização”. O conhecimento formal tem como base a conceitualização. Segundo Gruber [19], a conceitualização é constituída por objetos, conceitos e demais entidades que estão presentes em um domínio de conhecimento e também pelos relacionamentos existentes entre eles.

Borst [8] fez uma leve alteração na definição dada por Gruber [19], definindo a ontologia como “uma ontologia é uma especificação formal de uma conceitualização compartilhada”.

Para este trabalho adotaremos a definição dada por Fensel [15], o qual uniu o conceito de Gruber [19] com o conceito de Borst [8], definindo ontologia como “uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada”. Fensel [15] explicou cada termo da seguinte forma: uma ‘conceitualização’ é referida como um modelo abstrato de algum fenômeno no mundo que identifica os conceitos importantes daquele fenômeno. ‘Explícita’ significa que os tipos de conceitos usados e as restrições em seu uso são explicitamente definidos. ‘Formal’ refere-se ao fato que a ontologia deveria ser compreendida por uma máquina.

Como a ontologia tem o objetivo de disponibilizar um conhecimento comum de um domínio específico, geralmente seu desenvolvimento ocorre com a interação e cooperação entre várias pessoas. O desenvolvimento de ontologias não é um processo fácil, por isso, frequentemente, é necessário o reúso de ontologias já existentes que possam ser utilizadas na ontologia que está sendo desenvolvida, economizando esforço e tempo para definir uma conceitualização que já foi definida por outras pessoas.

Os componentes básicos para descrever o vocabulário de uma ontologia são: *classes, indivíduos, atributos, axiomas e as relações* existentes entre eles.

- *Classes* representam os conceitos de um domínio, são geralmente organizados em taxonomias permitindo que mecanismos de herança possam ser aplicados. Os conceitos representam termos que descrevem a sua semântica em um domínio. As classes podem representar conceitos abstratos (crenças, sentimentos, etc) ou conceitos específicos (pessoas, computadores, etc).
- Os *indivíduos* representam os objetos que são as instâncias das classes e das propriedades.
- Os *atributos* representam as propriedades das classes e das instâncias.
- Os *axiomas* modelam as sentenças que são sempre verdadeiras.
- As *relações* são usadas para modelar os tipos de associações entre as classes. As ontologias geralmente possuem relações binárias, onde se tem dois argumentos um chamado de domínio e o outro de intervalo. As relações são responsáveis pela

hierarquização das classes, criando uma estrutura de árvore que descreve a relação entre objetos.

### 2.1.2 Vantagens da Ontologia

Como visto anteriormente, ontologias visam capturar e modelar o conhecimento de alguma área de domínio para que possa ser reutilizada e compartilhada entre grupos de pessoas e aplicações de software.

Ontologias são baseadas em uma conceitualização de algum domínio específico, tendo esta conceitualização um vocabulário comum, o que evita interpretações ambíguas entre os grupos de pessoas que a utilizam.

O compartilhamento de conceitos formais e comuns de um domínio apoiam o uso de ontologias na interoperabilidade entre aplicações de software, permitindo uma integração transparente de serviços.

Outra vantagem de ontologias é a separação do conhecimento da parte técnica, deixando independente o domínio do conhecimento para que possa ser utilizado em outras aplicações de software.

A linguagem do vocabulário da ontologia pode ser mapeada permitindo que a conceitualização seja expressa em várias línguas, sem que com isso ocorra alteração da conceitualização.

Ontologias genéricas podem ser reusadas por outras ontologias permitindo que seus conceitos possam ser estendidos por uma ontologia de domínio específico.

### 2.1.3 Construção de Ontologias

Para que pessoas necessitariam construir ontologias? Esta é uma pergunta que Noy e McGuinness [25] respondem citando algumas razões:

- Compartilhar o entendimento comum da estrutura de informação entre pessoas ou agentes de software, este é um objetivo da construção de ontologias que permite a interoperabilidade entre aplicações de software, uma vez que os agentes de software terão uma mesma interpretação da semântica do vocabulário de um domínio de conhecimento.
- Permitir o reúso do conhecimento do domínio, isso nos economiza esforço e tempo na definição de conceitos que já foram representados em outra ontologia. Assim, podemos reutilizar, através da importação para a ontologia a ser desenvolvida, os conceitos definidos por esta ontologia existente. É possível também estendemos os conceitos importados para atender à especificação da ontologia que está sendo desenvolvida.

- Tornar explícitas as suposições do domínio, devemos estabelecer um acordo explícito de quais conceitos e suas interpretações semânticas serão utilizados em uma determinada ontologia. Tendo definido explicitamente este vocabulário comum evitaremos ambiguidades no entendimento de tais conceitos por pessoas ou aplicações de software.
- Separar o domínio do conhecimento do conhecimento operacional, esta liberdade proporciona a utilização da ontologia em outras aplicações e vice-versa. Isso deixa o domínio do conhecimento livre das amarras nas implementações das diversas linguagens de programação.
- Analisar o conhecimento do domínio, ontologias estão sendo desenvolvidas para uma grande variedade de aplicações e sendo mantidas por equipes de pessoas distintas que podem reutilizar conceitos de uma variedade de ontologias existentes para a sua ontologia de interesse. Portanto, é importante realizar uma análise formal dos conceitos que serão utilizados.

A metodologia de Noy e McGuinness [25] aborda um processo iterativo que propõe sete passos para o desenvolvimento de ontologias.

**Passo 1:** Determinar o domínio e escopo da ontologia

O passo inicial no desenvolvimento de ontologias é responder questões básicas que ajudem a definir o seu domínio e escopo, questões como "qual é o domínio que a ontologia irá cobrir?". Esboçar uma lista de questões na qual uma ontologia seja capaz de responder ajuda na definição do escopo.

**Passo 2:** Considerar o reuso de ontologias existentes

Geralmente vale a pena verificar se podemos importar ontologias já existentes de forma a estendê-las na ontologia de interesse que está sendo construída. Principalmente quando a aplicação necessita interagir com outras aplicações que utilizam ontologias existentes.

**Passo 3:** Enumerar termos importantes na ontologia

É conveniente a elaboração de uma lista detalhada de termos que consideramos relevantes dentro da área do conhecimento que queremos representar, sem a preocupação de sobreposições de conceitos, as relações entre eles, as propriedades de cada um ou se são classes ou propriedades.

**Passo 4:** Definir as classes e a hierarquia das classes

Podemos definir a hierarquia de classes utilizando abordagem top-down, ou seja, da classe mais geral para a mais específica, como podemos usar a abordagem bottom-up, ou seja, definindo as classes mais específicas para as classes mais gerais, ou podemos mesclar as duas abordagens, conhecida como abordagem de combinação. Qualquer que seja a abordagem utilizada o importante é definirmos as classes. Podemos utilizar a lista de

termos que elaboramos no passo anterior, para capturar os termos com existência própria, os quais serão as classes.

**Passo 5:** Definir as propriedades das classes

Uma vez que já definimos as classes, a partir da lista que elaboramos no passo 3, é importante definirmos a estrutura interna das classes, ou seja, as propriedades, pois apenas definindo as classes não conseguiremos responder à lista de questões de competência que criamos no passo 1. A grande parte dos termos que restaram na lista que elaboramos no passo 3 serão as propriedades que ligaremos nas classes. Como as subclasses herdam as propriedades da superclasse, devemos anexar as propriedades mais gerais nas superclasses.

**Passo 6:** Definir as facetas dos slots

As propriedades têm restrição quanto ao seu tipo (string, numérico, booleano, etc), aos valores que são permitidos, ao número de valores (cardinalidade) e assim por diante. Dessa forma evita que se atribua instâncias de valores não permitidos para alguma propriedade de uma classe.

**Passo 7:** Criar instâncias

E finalmente, criar instâncias que preenchem as classes. As instâncias representam o conjunto de elementos das classes de um domínio.

O desenvolvimento de ontologias compreende a escolha de uma linguagem que representará o domínio de uma área específica. Na seção seguinte será apresentada a linguagem de ontologias OWL, uma recomendação da W3C, e que utilizamos neste trabalho.

## 2.2 A linguagem OWL

Web Ontology Language (OWL) é a principal linguagem para definir e instanciar ontologias na Web, desenvolvida e recomendada pela W3C que a projetou para apoiar a construção de ontologias para a infraestrutura da Web Semântica.

OWL permite que informações contidas em documentos possam ser interpretadas por máquinas e não apenas por humanos, permitindo a interoperabilidade entre aplicações. OWL descreve formalmente e explicitamente os conceitos de um domínio de interesse bem como as relações entre eles.

Geralmente, os documentos OWL são representados pela sintaxe RDF/XML, que são disponibilizados na Web.

A linguagem OWL fornece três sublínguas para o desenvolvimento de ontologias que são incrementalmente expressivas.

OWL Lite é adequada para aqueles usuários que necessitam apenas utilizar uma sintaxe simples e que apoie uma hierarquia de classificação e restrições simples. Uma das vantagens dessa linguagem é ser de mais fácil entendimento, por parte de usuários, e sendo mais fácil de implementar, para desenvolvedores. A sua desvantagem é expressividade menor. Ela possui decidibilidade computacional, ou seja, toda a computação terminará em um tempo finito.

OWL DL é utilizada para aqueles usuários que desejam uma expressividade maior do que a oferecida pelo OWL Lite e que consiga ter decidibilidade computacional. Para manter a decidibilidade da linguagem ela restringe a forma como alguns construtores de OWL e de RDF podem ser utilizados. OWL DL tem como base a lógica descritiva, que pertence à lógica de Primeira Ordem.

OWL Full é toda a linguagem OWL, sendo empregada para se obter a máxima expressividade da linguagem OWL. Essa completude da linguagem a torna indecidível, não permitindo uma inferência computacional para todos os seus recursos.

Existe uma compatibilidade ascendente entre as sublínguas de OWL, pelo fato de que uma linguagem OWL Lite é um subconjunto da linguagem OWL DL e a linguagem OWL DL é um subconjunto da linguagem OWL Full. A relação abaixo é uma verdade, mas a sua inversa não.

- Cada ontologia legal OWL Lite é uma ontologia legal OWL DL.
- Cada ontologia legal OWL DL é uma ontologia legal OWL Full.
- Cada conclusão válida OWL Lite é uma conclusão válida OWL DL.
- Cada conclusão válida OWL DL é uma conclusão válida OWL Full.

A escolha de uma sublíngua de OWL para o desenvolvimento de ontologias, depende do grau de expressividade que o desenvolvedor necessita dar para a construção de sua ontologia.

Os componentes básicos de uma ontologia OWL são: as classes, as propriedades e as instâncias das classes.

As classes OWL são as entidades principais de uma ontologia e representam os diferentes conceitos de uma ontologia, podendo ser organizadas em hierarquias de superclasse-subclasse. Estas classes contêm os indivíduos que fazem parte de uma ontologia. Os indivíduos são as instâncias das classes de uma ontologia OWL. Os indivíduos herdam as propriedades das classes de que são membros.

Uma propriedade é uma relação binária. A linguagem OWL categoriza estas propriedades da seguinte forma:

- *Propriedade objeto* que relaciona indivíduos a indivíduos. Pode ser especificado o domínio e a imagem para o relacionamento das classes. Por exemplo, a propriedade *isMadeBy* tem um domínio *Vinho* e uma imagem *Vinícola*.
- *Propriedade de tipo de dado (datatype)* que relaciona as instâncias de classes com os tipos de dados que podem ser literais RDF e tipos de dados XML schema. Por exemplo, temos os seguintes tipos de dados que podem ser usados com OWL: *string* (*xsd:string*), *número decimal* (*xsd:decimal*), *número inteiro* (*xsd:integer*), *data e hora* (*xsd:dateTime*).

O grupo de trabalho OWL do W3C revisou a linguagem OWL e adicionou novos recursos, o que resultou em uma nova versão da OWL conhecida como OWL 2.

### 2.2.1 OWL 2

O sucesso com experiências práticas no uso da linguagem OWL 1 para o desenvolvimento de ontologias nas diversas áreas, tais como, Engenharia, Medicina, Direito, entre outras, tem revelado a necessidade de melhorias da linguagem, o que resultou na sua revisão e extensão pelo W3C.

A OWL 2 é uma linguagem para o desenvolvimento de ontologias para a Web Semântica com significado definido formalmente (W3C OWL Working Group, 2012) e da mesma forma que o OWL 1, o OWL 2 visar tornar o desenvolvimento de ontologias mais fácil e o compartilhamento do conhecimento na Web com o objetivo de possibilitar a interoperabilidade entre as aplicações de software.

Algumas das novas funcionalidades incorporadas na OWL 2 são:

- Inclusão de novos construtores para as propriedades que aumentam a expressividade da linguagem, por exemplo, *disjunção de propriedades* declara que dois indivíduos não podem estar ligados por duas propriedades diferentes do conjunto.
- Estende a capacidade dos tipos de dados, permite definir novos tipos de dados e restringir intervalos de valores.
- Capacidades de *meta modelagem simples*, permite o uso de mesmo nome para classes e indivíduos.
- Estende o uso de *anotações*, que permite agora *anotações de axiomas*.
- Adição de novas sintaxes para representar ontologias OWL, por exemplo, a sintaxe *Manchester* que é uma sintaxe mais simples para ler.

Com o objetivo de favorecer a eficiência dos raciocinadores, OWL 2 especificou três perfis que são independentes entre si e que são conhecidos como: *OWL 2 EL*, *OWL 2 QL* e *OWL 2 RL*. Os perfis são definidos pelo uso de restrições na estrutura das ontologias OWL 2. Cada perfil oferece um poder diferente de expressividade e

de diferente cenário de aplicação. Abaixo descreveremos brevemente cada um desses profiles.

*OWL 2 EL* é utilizado em aplicações que possuem um grande número de propriedades e/ou classes. O raciocínio para este profile pode ser implementado em tempo polinomial no tamanho da ontologia.

*OWL 2 QL* é empregado em aplicações que trabalham com grandes volumes de instâncias de dados, o objetivo deste profile é responder as questões de consultas usando tecnologia de banco de dados relacional.

*OWL 2 RL* é indicado para aplicações que exigem raciocínio escalável sem que prejudique muito a expressividade da linguagem.

## 2.3 SWRL

Apesar do poder de expressividade que a linguagem OWL oferece, por meio de seus construtores para a definição de ontologias, ela não consegue expressar todo o conhecimento de um domínio de interesse. Por exemplo, para modelar uma estrutura de relacionamento geral, no qual se deseja saber se uma pessoa tem um tio que é irmão de sua mãe utilizando apenas as primitivas de OWL não é possível descrever tal relacionamento. Portanto, para dar mais poder de expressividade para as ontologias OWL foi desenvolvido pelo W3C a Semantic Web Rule Language (SWRL). SWRL é uma linguagem de regras desenvolvida pela combinação das sublínguas da linguagem de ontologias OWL, OWL Lite e OWL DL, com as sublínguas da linguagem de marcação RuleML, Unary/Binary Datalog, conforme descrito por Horrocks et al [22]. A utilização de regras SWRL nos documentos de ontologias OWL permite uma maior expressividade na representação do conhecimento do domínio.

### 2.3.1 Estrutura das regras em SWRL

As regras de SWRL são descritas na forma de implicação (Horn-like), onde temos um antecedente (corpo) e um conseqüente (cabeça). A regra pode ser lida da seguinte forma “se o antecedente for verdadeiro então o conseqüente é verdadeiro”.

$$\text{antecedente (corpo)} \Rightarrow \text{conseqüente (cabeça)}$$

Tanto o antecedente quanto o conseqüente são formados por conjunções positivas denominadas átomos.

$$\text{átomo} \wedge \text{átomo} \wedge \dots \Rightarrow \text{átomo} \wedge \text{átomo}$$

Os átomos podem ser representados da seguinte forma:

$$p(\text{argum1}, \text{argum2}, \dots, \text{argumx})$$

onde  $p$  significa o predicado do átomo e  $\text{argum1}$ ,  $\text{argum2}$ ,  $\text{argumx}$  são os argumentos deste predicado.

Existem sete tipos de átomos que podem ser utilizados nas regras SWRL, cada tipo pode ter um número definido de argumentos, na Tabela 2.1 são ilustrados estes tipos.

**Tabela 2.1:** *Tipos de átomos da regra SWRL*

Átomo	Descrição
class	Tipo de átomo que representa classes da ontologia OWL.
individual property	Tipo de átomo que representa propriedades de objetos que relacionam os indivíduos da ontologia OWL.
datavalued property	Tipo de átomo que representa propriedades que relacionam dados aos indivíduos da ontologia OWL.
data range	Tipo de átomo que representa o tipo de dado definido na ontologia.
sameAs	Tipo de átomo que determina se dois indivíduos são os mesmos.
differentFrom	Tipo de átomo que determina se dois indivíduos são diferentes um do outro.
built-in	Tipo de átomo que utiliza de funções existentes ou definidas pelo usuário que retorna o valor verdadeiro se os argumentos são satisfeitos.

Os argumentos dos átomos são os indivíduos ou valores de dados da ontologia OWL, ou variáveis que expressam eles. As variáveis são precedidas do sinal de interrogação (?).

É importante ressaltar que apenas variáveis que se encontram no antecedente devem aparecer no consequente.

Em relação aos argumentos dos átomos temos as seguintes informações.

O átomo class é representado pelo nome da classe OWL e por um argumento que é um indivíduo desta classe.

$$\text{Person}(?x) \text{ (a)}$$

$$\text{Person}(\text{Roberto}) \text{ (b)}$$

Nesses dois exemplos é apresentada uma classe OWL chamada Person, sendo que o argumento de (a) é uma variável (?x) e o argumento de (b) é um indivíduo específico (Roberto). A classe Person será considerada verdadeira se seus argumentos forem instâncias desta classe.

O átomo individual property é formado por uma propriedade objeto e por dois argumentos representando indivíduos OWL.

parent(?x,?y) (a)  
parent(Tiago, Renato) (b)

Onde, parent representa uma individual property OWL e os argumentos de (a) são duas variáveis ?x e ?y e os argumentos de (b) são os indivíduos OWL “Tiago” e “Renato”. A individual property parent será considerada verdadeira se o relacionamento de seus argumentos for mantido por essa propriedade.

O átomo datavalued property é constituído por uma propriedade data e por dois argumentos que representam, respectivamente, um indivíduo e um valor de dado OWL. Este valor de dado é relacionado ao indivíduo na datavalued property de que fazem parte.

hasAge(?x, ?y) (a)  
hasAge(Renato, 22) (b)  
hasAge(?x, 40) (c)

Para estes exemplos temos o seguinte, uma datavalued property nomeada de hasAge, sendo que no exemplo (a) ?x e ?y são variáveis que representam, respectivamente, um indivíduo e um valor de dado. No exemplo (b), um indivíduo chamado “Renato” está relacionado ao valor 22. Por fim, no exemplo (c) existe uma variável ?x que representa um indivíduo que está relacionado ao valor de dado 40.

O átomo data range é formado por um tipo de dado usado na ontologia e um argumento que representa um valor de dado.

xsd:string(?x)

Para esse exemplo temos um tipo de dados no formato XML schema do tipo string e uma variável ?x que representa um valor de dado. Este átomo será considerado verdadeiro se o argumento for uma instância de seu tipo de dado.

O átomo sameAs é constituído por um termo sameAs com dois argumentos que representam indivíduos OWL. Caso o átomo sameAs determine que os dois argumentos sejam iguais então será mantida a verdade deste átomo.

sameAs(?x, ?y) (a)  
sameAs(Roberto, Carlos) (b)

Nesses dois exemplos temos o termo sameAs, sendo que no exemplo (a) aparecem duas variáveis, ?x e ?y, que representam indivíduos OWL e no exemplo (b) temos dois indivíduos chamados, respectivamente, Roberto e o outro Carlos. Caso se trate da mesma instância OWL então será mantida a verdade deste átomo.

O átomo differentFrom é constituído por um termo differentFrom com dois argumentos que representam indivíduos OWL. Caso o átomo differentFrom determine que os dois argumentos sejam diferentes então será mantida a verdade deste átomo.

differentFrom(?x,?y) (a)  
differentFrom(Marcos, Fernando) (b)

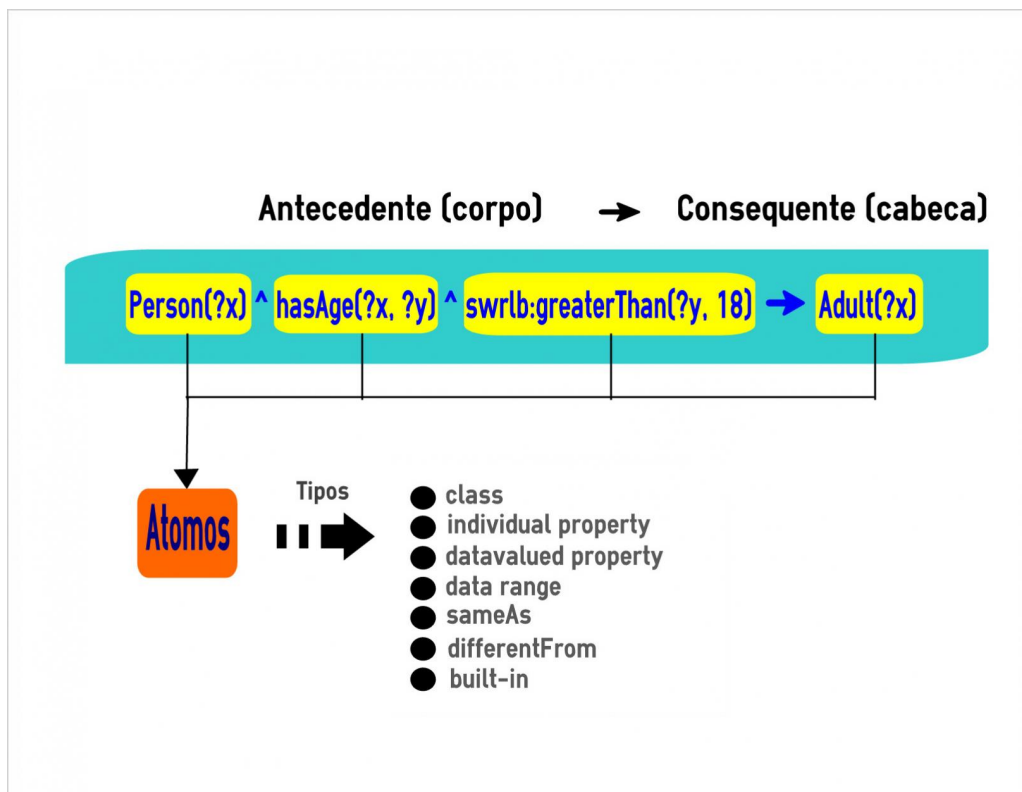
Nesses exemplos temos o átomo differentFrom, no qual o exemplo (a) possui duas variáveis ?x e ?y que representam indivíduos OWL e no exemplo (b) temos dois indivíduos específicos um chamado Marcos e o outro chamado Fernando. Em ambos exemplos será determinado se os argumentos são distintos.

O átomo built-in pode ser representado por suas várias funções (que podem ser classificadas como sendo de comparação, funções matemática, valores lógicos, string, data, tempo, duração, URI e listas) e que podem possuir um ou mais argumentos dependendo da função utilizada.

swrlb:lessThanOrEqual(?x, 40)

No exemplo acima apresentado é utilizada uma função de comparação, swrlb:lessThanOrEqual, e dois argumentos, uma variável ?x e um número 40. Caso a variável ?x seja menor ou igual ao número 40 então este átomo manterá o valor verdadeiro.

Na Figura 2.1 é apresentado um exemplo de uma regra SWRL que considera que uma pessoa é adulta quando possui idade igual ou maior de 18 anos.



**Figura 2.1:** Exemplo de uma regra SWRL

É importante destacar que as regras SWRL são armazenadas no próprio documento da ontologia OWL e que podem inferir em novas informações ao conhecimento da ontologia OWL.

## 2.4 Computação sensível ao contexto

O contexto é considerado muito útil, uma vez que ele fornece informações relevantes acerca da situação de um indivíduo, lugar e objetos.

Segundo a definição de Dey [14], Informação de contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade, sendo que uma entidade pode ser uma pessoa, um lugar ou um objeto considerados relevantes para a interação entre um usuário e uma aplicação, incluindo o usuário e a própria aplicação. Assim, se alguma informação caracterizar a situação de uma entidade em uma interação, esta informação é considerada de contexto.

Desenvolvedores de aplicações sensíveis ao contexto tendo ciência do que é contexto podem selecionar as informações de contexto que são relevantes para a sua aplicação.

Dey [14] também definiu que um sistema é considerado sensível ao contexto quando ele utiliza as informações de contexto para oferecerem serviços e informações relevantes a usuários ou a outras aplicações com o intuito de facilitar as suas atividades.

Abowd e Mynatt [1] definem um conjunto mínimo de informações de contexto para auxiliar projetistas e desenvolvedores de aplicações sensíveis ao contexto nas suas atividades de modelar e desenvolver suas aplicações. Estas informações de contexto são:

- Who (quem) - Como seres humanos nós lembramos de atividades e eventos passados baseados na presença de outras pessoas. Esta informação de contexto está relacionada a identidade de outras pessoas a quem um indivíduo interage.
- What (o quê) - Esta informação de contexto está associada com a atividade desempenhada em uma interação.
- Where (onde) - É a informação de contexto que está relacionada onde ocorreu a interação.
- When (quando) - É a informação de contexto que está relacionada ao tempo em que ocorreu a interação.
- Why (por quê) - Esta informação de contexto está relacionada ao por quê está sendo realizada uma determinada ação dentro de uma interação.

Além dessas informações de contexto Truong et al [39] definem o contexto How (como) que é uma tarefa relacionada à captura e ao acesso de informações de contexto.

## 2.5 Considerações finais

Neste capítulo foram apresentadas informações sobre ontologias, a linguagem para ontologias OWL e a linguagem de regras SWRL. Estes conceitos são utilizados no desenvolvimento deste trabalho.

Ontologia é o alicerce para a representação do conhecimento deste trabalho que visa representar o domínio de monitoramento de sinais vitais humanos.

O vocabulário comum oferecido pela ontologia é formalizado e expressado semanticamente através do uso da linguagem OWL. Dessa forma, evita-se a ambiguidade do entendimento dos conceitos representados nesse trabalho e permite a inferência e interoperabilidade entre as aplicações que tenham interesse no monitoramento de sinais vitais humanos.

As regras escritas em SWRL possibilitaram a inferência de informações relacionadas às notificações relacionadas ao estado clínico do paciente.

---

## Tecnologias Relacionadas

---

Neste capítulo apresentaremos as ontologias importadas, SeCoM e VSO, para a ontologia de monitoramento de sinais vitais humanos, bem como, a ferramenta Protégé que é utilizada para a edição e construção do modelo e o raciocinador Pellet que realiza as inferências do modelo ontológico proposto.

### 3.1 Modelo SeCoM

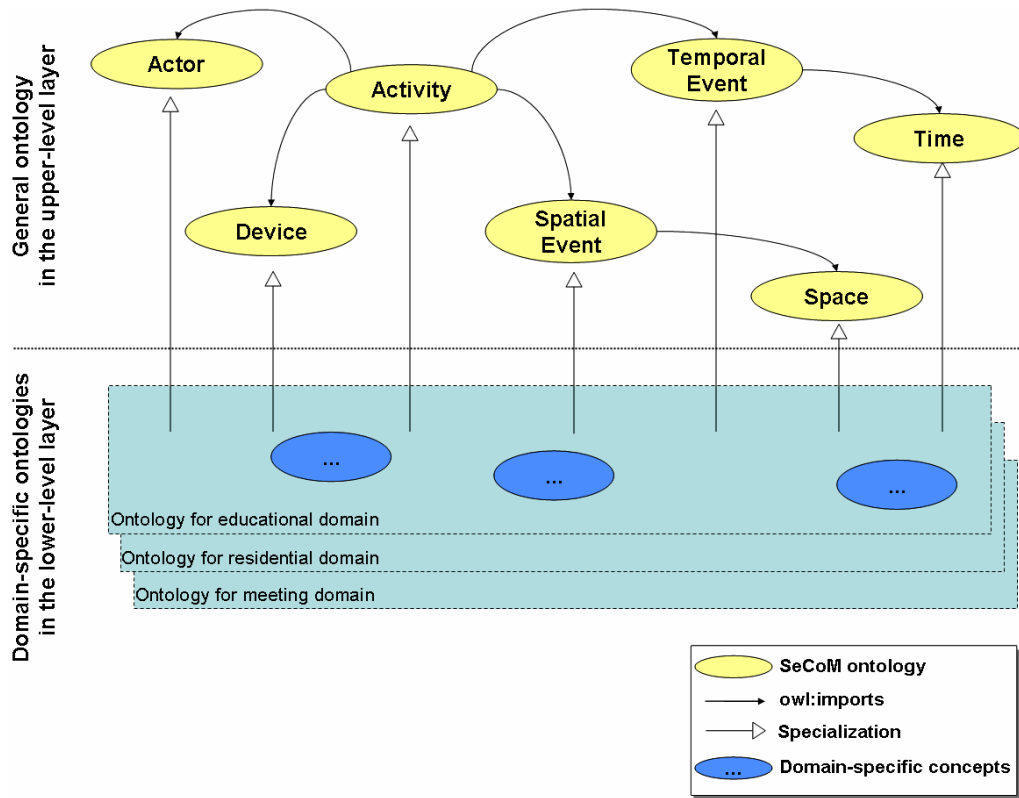
O modelo *Semantic Context Model - SeCoM* é um modelo de propósito geral que apoia o desenvolvimento de aplicações sensíveis ao contexto baseadas em ontologias. Este modelo foi proposto por Bulcão-Neto [9] em sua abordagem de engenharia de software para apoiar o desenvolvimento de aplicações sensíveis ao contexto empregando tecnologia de Web Semântica [7].

Esse modelo é formado por um conjunto de ontologias em que cada uma dessas ontologias está relacionada a uma dimensão semântica da informação de contexto, conforme relatado por Bulcão-Neto e Pimentel [11].

As dimensões semânticas da informação de contexto estão divididas e relacionadas com as ontologias da seguinte maneira: When correspondem as interações de tempo (ontologia Time), Where estão relacionadas com as interações de localização (ontologia Spatial), Who corresponde as interações dos participantes (ontologia Actor), How são as interações que correspondem à captura e ao acesso das informações de contexto (ontologia Device) e What que correspondem as interações dos eventos (ontologia Activity).

As ontologias *Temporal Event* e *Spatial Event* são extensões das ontologias *Time* e *Spatial* que representam eventos de extensão espaciais e temporais, respectivamente.

Na Figura 3.1 é apresentado o modelo SeCoM.



**Figura 3.1:** Modelo SeCoM [9]

Como apresentado na Figura 3.1, o modelo SeCoM segue uma abordagem em duas camadas, de forma que na camada superior se encontram as ontologias que compõem o modelo e as relações entre elas e a camada inferior representa as ontologias que poderão ser estendidas do modelo principal para atender às necessidades de um projetista no desenvolvimento de uma aplicação sensível a contexto.

O modelo SeCoM, por ser genérico, foi desenvolvido para oferecer características de modularidade e extensibilidade que atendam aos requisitos de diversos domínios de interesse. Por ser baseada em ontologias o modelo SeCoM se beneficia da formalização e expressividade oferecida pela linguagem de ontologias OWL.

A seguir faremos uma apresentação de forma geral e sucinta das ontologias que compõem o modelo SeCoM.

### 3.1.1 Ontologia Time

A ontologia *Time* representa através de suas classes, atributos e relações o conhecimento sobre o domínio de tempo. Nela estão caracterizadas informações referentes a intervalo e instante de tempo. As entidades temporais podem ser representadas através das diversas unidades que descrevem o tempo: segundo, minuto, hora, dia, semana, mês, ano e assim por diante.

O diagrama UML da Figura 3.2 apresenta a ontologia Time com suas classes, atributos e relações.

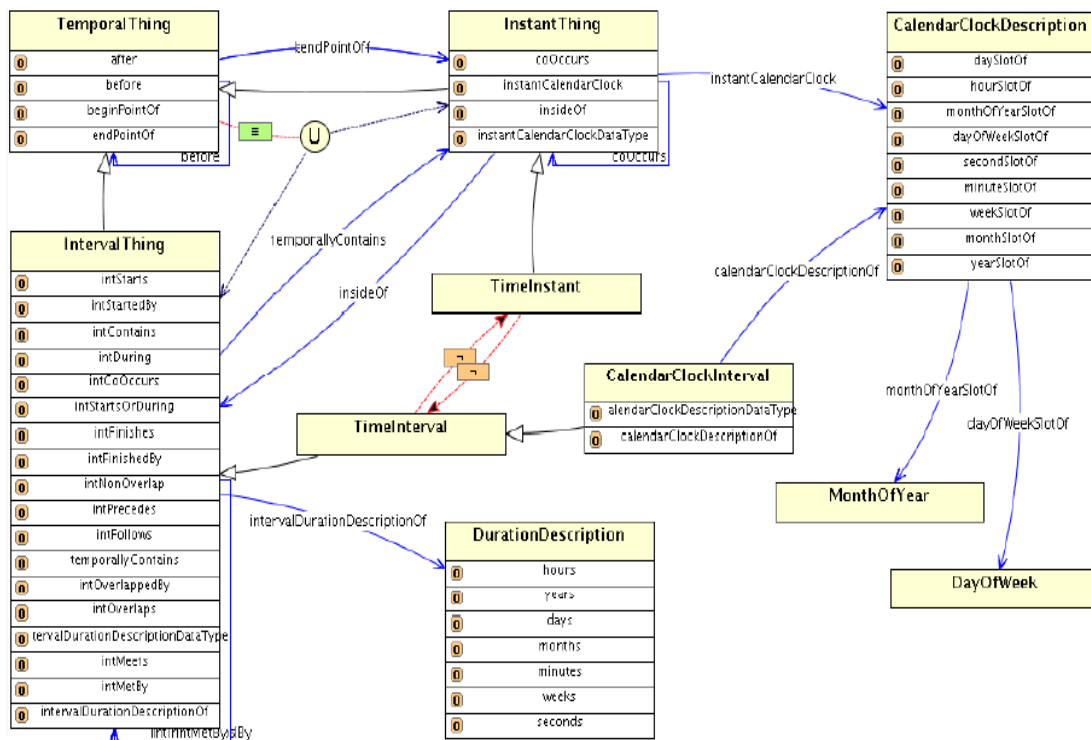


Figura 3.2: Ontologia Time [9]

Por meio da ontologia Time que está presente no modelo SeCoM é possível representar qualquer entidade que apresente uma extensão temporal por meio de intervalo de tempo ou de instante de tempo.

Informações mais detalhadas sobre a ontologia Time podem ser encontradas em Bulcão-Neto [9].

### 3.1.2 Ontologia Temporal Event

A ontologia *Temporal Event* é uma complementação da ontologia Time que modela eventos relacionados a extensão temporal quanto a instantes e intervalos de tempo, conforme descrito por Bulcão-Neto et al [10]. Na ontologia Temporal Event são acrescentadas novas informações de contexto que representam o início e fim de um evento, assim como, a periodicidade e a frequência de um evento.

Esta ontologia adiciona mais expressividade em eventos de extensão temporal ao modelo SeCoM. O diagrama UML da Figura 3.3 ilustra a ontologia Temporal Event, as classes com o espaço de nome time: representam as classes da ontologia Time.

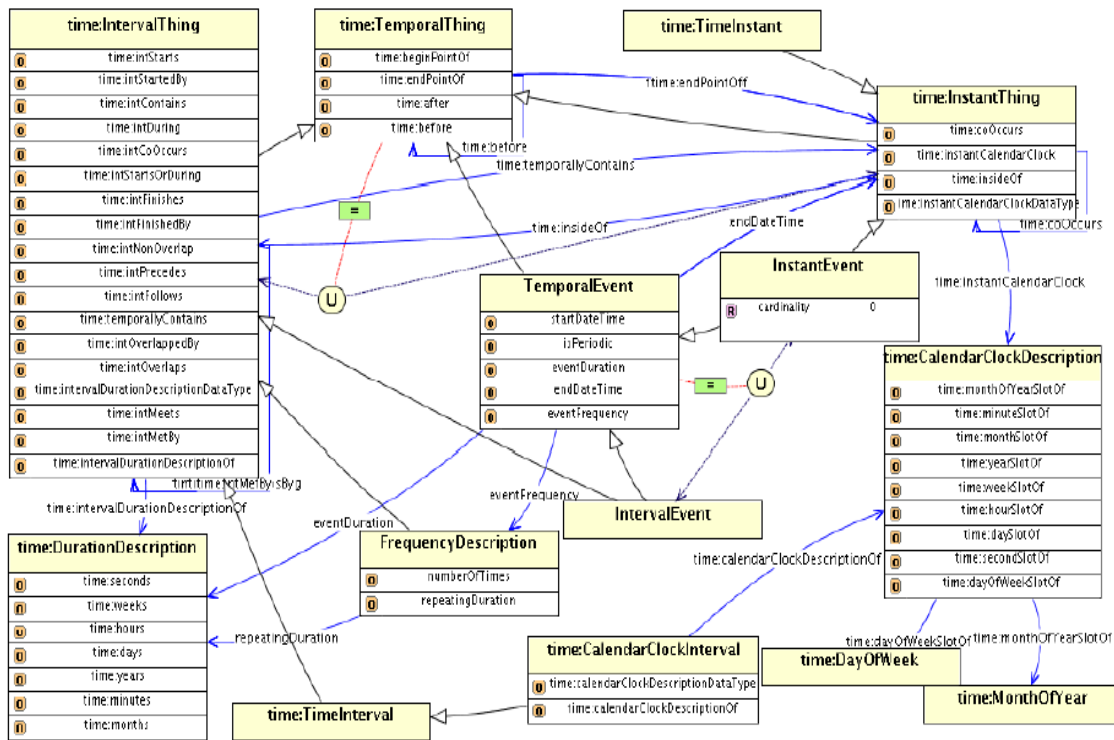


Figura 3.3: Ontologia Temporal Event [9]

Maiores informações sobre a ontologia Temporal Event podem ser encontradas relatadas por Bulcão-Neto [9].

### 3.1.3 Ontologia Spatial

A ontologia *Spatial* é uma ontologia que representa informações referentes a elementos que possuem uma extensão espacial descritas tanto como um espaço físico ou virtual e as relações existentes entre elas.

Na representação de informações contextuais de espaço físico podem ser modeladas relações parte-todo, como por exemplo, descrever que uma sala pertence a um andar de um edifício, a sua localização geográfica como a latitude, longitude e altitude e também informações sobre divisões administrativas como cidade, estado e país.

A ontologia Spatial é uma das ontologias que compõem o modelo SeCoM e portanto, capacita o modelo a descrever qualquer elemento que possua extensão temporal, por exemplo, espaço físico (sala de aula) ou virtual (sala de videoconferência).

O diagrama UML da Figura 3.4 apresenta a ontologia Spatial do modelo SeCoM.

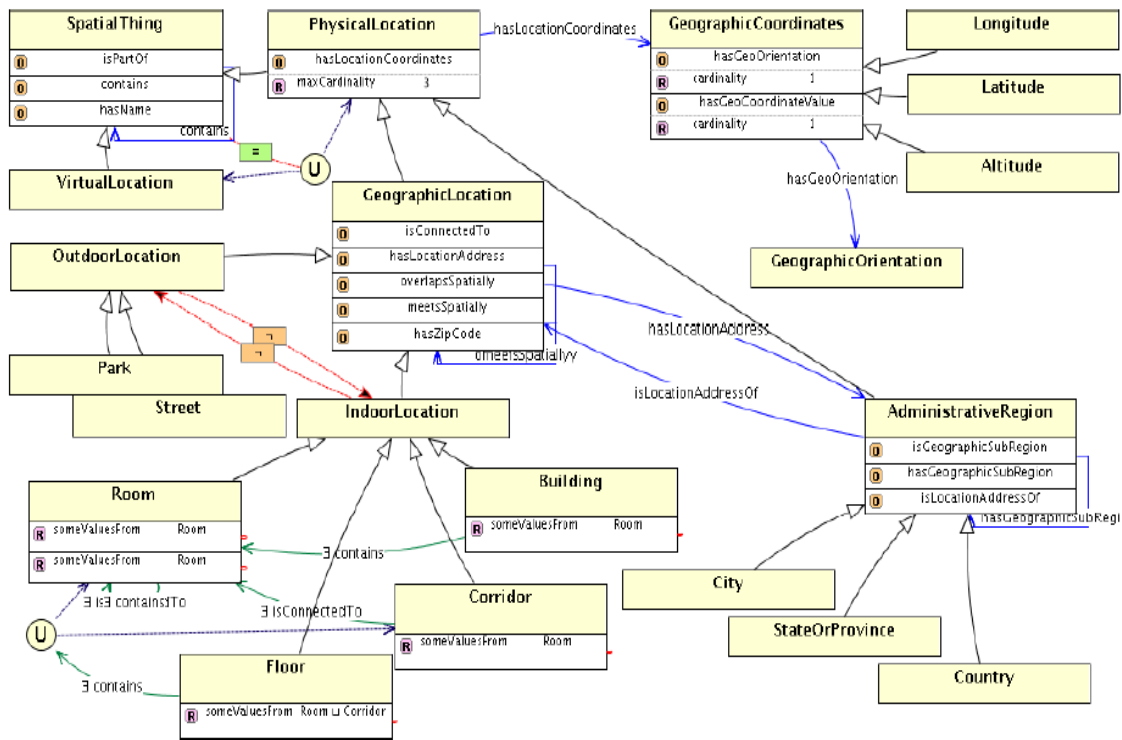


Figura 3.4: *Ontologia Spatial* [9]

Informações detalhadas sobre a ontologia Temporal Event podem ser encontradas relatadas por Bulcão-Neto [9].

### 3.1.4 Ontologia Spatial Event

Da mesma maneira que a ontologia Temporal Event representa uma extensão da ontologia Time que modela elementos que contêm informações de extensão temporal, a ontologia *Spatial Event* é um complemento da ontologia Spatial que modela elementos que contêm informações de extensão espacial.

Na ontologia Spatial Event são acrescentadas novas classes, atributos e relações que permitem relacionar eventos a localizações físicas e virtuais.

A separação de elementos que possuem informações de extensão espacial dos elementos que representam eventos que ocorrem em localizações físicas e virtuais permite ao modelo SeCoM, através da ontologia Spatial Event, modelar eventos que ocorrem em localizações espaciais tais como, uma reunião agendada na sala da diretoria ou arquivos de imagens enviadas de uma sala de videoconferência.

O diagrama UML da Figura 3.5 ilustra a ontologia Spatial Event no modelo SeCoM, onde espaço de nome loc: está relacionado a ontologia Spatial.



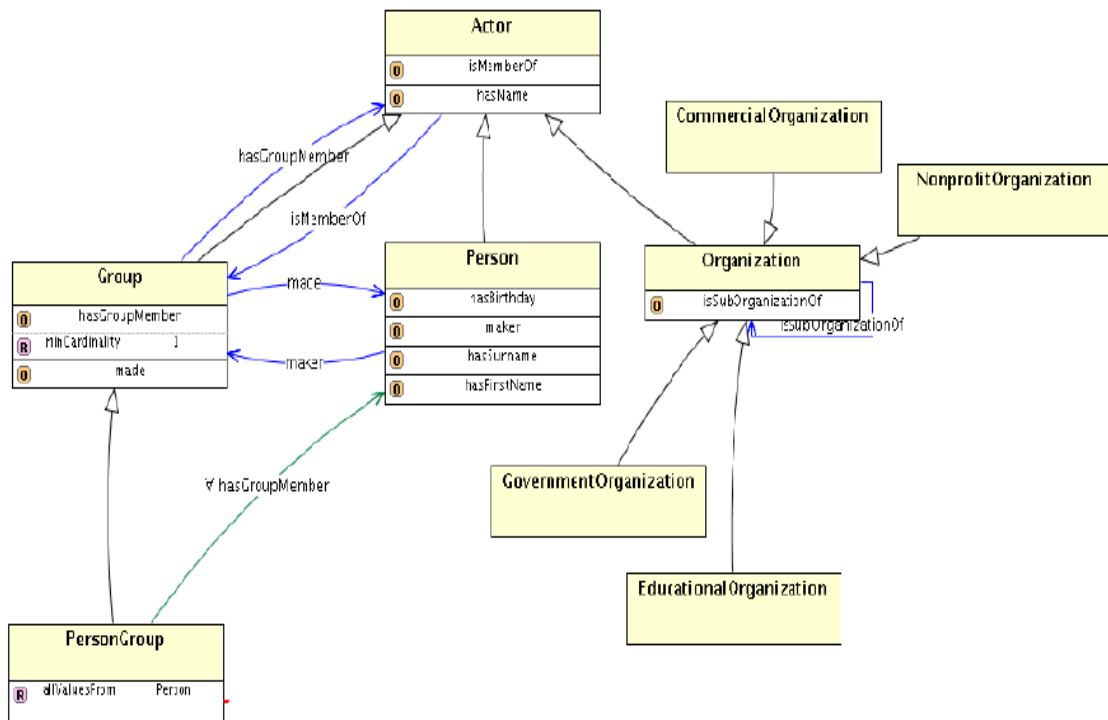


Figura 3.6: Ontologia Actor [9]

Informações mais detalhadas sobre a ontologia Actor é relatada por Bulcão-Neto [9].

### 3.1.6 Ontologia Activity

A ontologia *Activity* foi a última ontologia a ser desenvolvida para compor o modelo SeCoM, pois necessitaria das informações das ontologias já mencionadas. O objetivo da ontologia Activity foi descrito por Bulcão-Neto [9] como "o objetivo da ontologia Activity é descrever ações que atores realizam ou fazem acontecer em um ambiente de computação sensível a contexto".

Para a construção da ontologia Activity foi realizada a importação das ontologias Temporal Event, Spatial Event, Actor e Device. Consequentemente, as ontologias Time e Spatial também foram importadas uma vez que fazem parte das ontologias Temporal Event e Spatial Event, respectivamente.

Para o desenvolvimento da ontologia Activity foram seguidas certas orientações de projeto que descrevem atividades em relação: ao tempo em que ocorrem, à localização em que ocorrem, aos atores/participantes envolvidos, aos dispositivos computacionais presentes na interação usuário-máquina e à diferenciação das atividades que ocorrem de forma espontânea ou prevista.

As atividades são descritas como eventos espaço-temporais, visto que são baseadas nas ontologias importadas Temporal Event e Spatial Event.

Atores participam das atividades por meio das relações existentes entre a classe de atividade e de ator a qual é importada da ontologia Actor.

Nas atividades podem ser expressas que atores utilizam dispositivos computacionais que foram importados da ontologia Device.

É possível distinguir quando uma atividade acontece espontaneamente ou de forma planejada através de atributos e conceitos que estão relacionados com as atividades.

O diagrama UML da Figura 3.7 apresenta a ontologia Activity do modelo SeCoM.

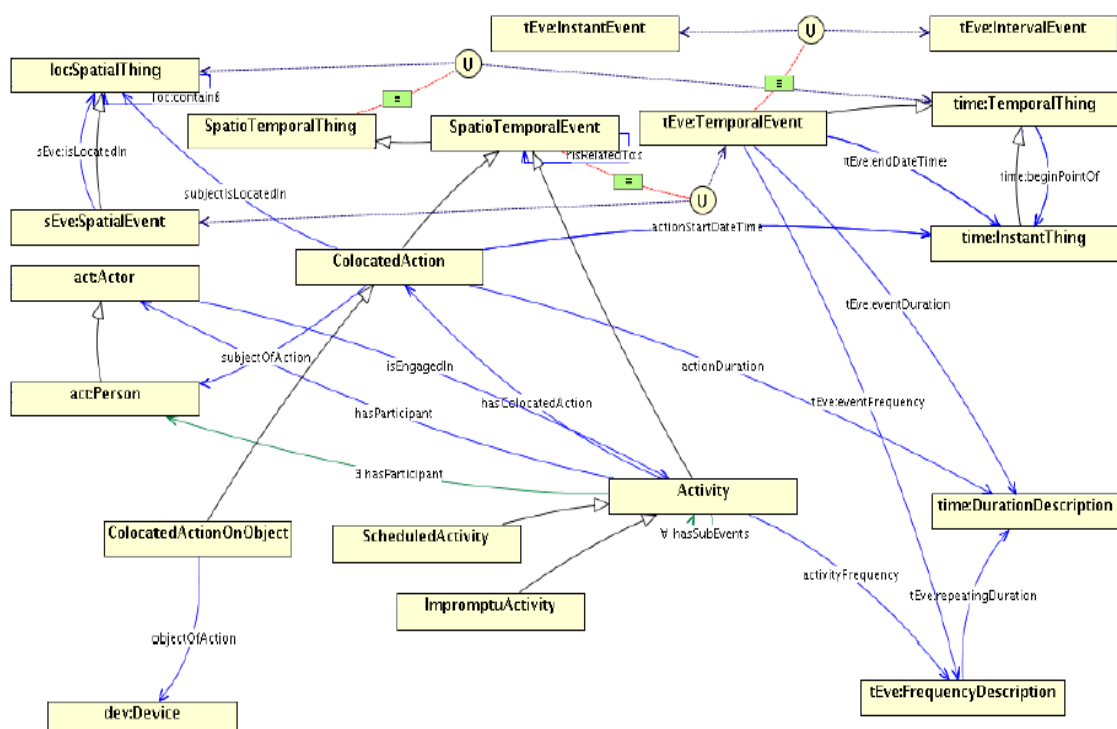
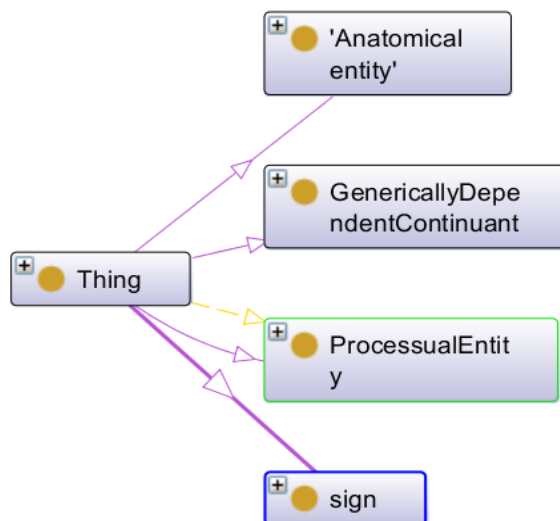


Figura 3.7: Ontologia Activity [9]

## 3.2 Vital Sign Ontology - VSO

A ontologia VSO é uma ontologia que representa o conhecimento do domínio sobre sinais vitais humanos, ou seja, os sinais vitais: pressão arterial, temperatura corpórea, frequência respiratória e frequência de pulso. Além de descrever os sinais vitais acima citados, a ontologia VSO também descreve vários processos de realização de medições, dados de medição dos sinais vitais e várias entidades anatômicas que participam destas medições, conforme descrito por Goldfain [17]. A Figura 3.8 ilustra a ontologia VSO.



**Figura 3.8:** *Ontologia VSO* [17]

VSO foi desenvolvida empregando a linguagem de ontologias OWL, especificamente a sublíngua OWL-DL, o que a faz ter decidibilidade computacional, e definições de linguagem natural concisas.

Projetistas podem se beneficiar da ontologia VSO no desenvolvimento de aplicações que utilizem ontologias que representem domínios de interesse na área de saúde e que necessitem de informações sobre sinais vitais humanos. As ontologias a serem desenvolvidas podem importar a ontologia VSO para serem estendidas no seu domínio de interesse.

Segundo Goldfain et al [17], o objetivo em desenvolver a ontologia VSO é fornecer um vocabulário controlado extensível, rigoroso cientificamente, consistente e computável para facilitar a troca de dados e anotações entre aplicações que necessitem de informações sobre sinais vitais.

Embora exista uma classe (vital sign) na ontologia VSO que descreve apenas os quatro sinais vitais que são consensos universalmente, ela é extensível para acomodar outros sinais vitais que podem estar previstos na especificação de requisitos de projetistas para o desenvolvimento de aplicações específicas.

A ontologia VSO é uma extensão da *Ontology for General Medical Science* (OGMS). OGMS é uma ontologia desenvolvida com base no conhecimento da teoria geral de doenças descrita no artigo de Scheuermann et al [33] e que representa entidades que estão envolvidas em uma consulta médica, a qual descreve termos como: *disease*, *disorder*, *sign*, *diagnosis* entre outros termos relacionados a doenças e diagnósticos.

Com o desenvolvimento de VSO cobriu-se o espaço vazio na cobertura de sinais clínicos da ontologia de *Open Biomedical Ontologies* (OBO).

### 3.2.1 Ontologias importadas para VSO

Segundo Goldfain et al [17], para a construção da ontologia VSO foram importadas para seu projeto os conceitos das seguintes ontologias:

- Basic Formal Ontology (BFO): na qual utiliza e estende os conceitos: *continuants* e *occurents*, referentes, respectivamente, aos objetos, tal como um coração, e as entidades processuais, tal como o processo de respiração, conforme descrito por Spear [38].
- Ontology for General Medical Science (OGMS): a qual utiliza e estende o conceito *sign* para o conceito de vital sign dentro de VSO.
- Ontology for Biomedical Investigations (OBI): utilizando e estendendo os conceitos: *measurement process* e *measurement datum*, as quais se referem aos processos e valores de dados das medições relacionadas aos sinais vitais de VSO.
- Phenotypic Qualities Ontology (PATO): no qual utiliza e estende os conceitos *rate*, *temperature* e *pressure* referenciando aos sinais vitais dentro de VSO: frequência de pulso, frequência respiratória, temperatura corpórea e pressão arterial que são consenso universal na área de saúde.
- Foundational Model of Anatomy (FMA): em que utiliza e estende o conceito *anatomical entities* dentro de VSO referenciando a localização no corpo humano na qual as medições dos sinais vitais ocorrem.
- Gene Ontology (GO): na qual utiliza e estende os conceitos *biological processes* e *regulation of a biological quality*.
- A organização relacional utilizada na ontologia VSO são importadas da ontologia OBO Relation Ontology (RO) e de relações estendidas desta ontologia. As ontologias que são construídas baseadas nas boas práticas de OBO utilizam as relações da ontologia OBO-RO.

### 3.2.2 A representação dos sinais vitais na VSO

Os sinais vitais são representados em VSO como subclasses da classe 'vital sign', que conseqüentemente é uma subclasse de *sign* importada da OGMS.

*Pressão sanguínea*, representada pela classe 'blood pressure', descreve intervalos temporais no ciclo cardíaco e está dividida em seu primeiro nível em duas pressões: sistólica (que representa o período de contração do coração), representada pela classe 'diastolic blood pressure', e diastólica (que representa o período de relaxamento do coração) representada pela classe 'systolic blood pressure'. Outros intervalos do ciclo cardíaco derivados das pressões sistólica e diastólica são representadas na VSO. Dessa forma VSO cobre a representação de termos do ciclo cardíaco que o OBO não cobre.

Entidades da classe pressão arterial se relacionam com as entidades da classe 'Anatomical entity' nas localizações em que são pertinentes o sinal vital.

*Temperatura corpórea* é representada pela classe 'body temperature', as suas subclasses são descritas indicando em que parte do corpo a medição da temperatura corpórea ocorre.

As temperaturas corpóreas são obtidas a partir de buracos, cavidades ou lúmens existentes no corpo humano.

A temperatura corpórea é relativamente igual em toda a região corporal, mas é necessário indicar a região do corpo onde foi realizada a medição, uma vez que os valores normais para cada região diferem de forma significativa clinicamente e porque o monitoramento para cada paciente pode ser realizado de formas diferentes.

As entidades da classe de temperatura corpórea se relacionam com as entidades das classes 'Anatomical entity' informando a localização do corpo em que o sinal vital participa.

*Frequência de Pulso* é representada pela classe 'pulse rate', as suas subclasses se diferenciam na sua representação pela localização da artéria na qual ocorre a medição.

As entidades pertencentes à classe frequência de pulso se relacionam com as entidades das classes 'Anatomical entity' para informar em qual artéria ocorreu o processo de pulsação.

*Frequência respiratória* é representada pela classe 'respiratory rate' a qual não possui subclasses na versão atual de VSO.

Ambas as frequências acima apresentadas, especificamente, dependem simultaneamente do tempo e da localização para realizar suas medições, conforme descrito por Goldfain et al [17].

### **3.2.3 Representação da localização anatômica do corpo humano em VSO**

Na ontologia VSO são representadas as localizações anatômicas do corpo humano em que os sinais vitais ocorrem. As entidades instanciadas para esta representação pertencem às subclasses da classe 'Anatomical entity' que foi importada da ontologia Foundational Model of Anatomy (FMA) pela VSO.

As subclasses de 'Anatomical entity' representam várias partes anatômicas do corpo humano, por exemplo, a classe 'Axillary fossa' que representa a localização do

corpo humano onde o sinal vital de temperatura axilar ocorre.

FMA é um domínio de conhecimento da área de biomedicina em evolução baseada em computador. Seu objetivo é a representação de classes e os relacionamentos entre elas para a representação da estrutura fenotípica do corpo humano de maneira que seja interpretável por humanos e também navegável, analisável e interpretável por aplicações baseadas por máquinas. Resumidamente, FMA é uma ontologia que representa um coerente corpo do conhecimento declarativo explícito sobre a anatomia humana, conforme descrito em SIG-FMA [34].

O relacionamento entre a localização da anatomia humana e os sinais vitais é realizado através dos relacionamentos importados da ontologia OBO-RO pela VSO.

### **3.2.4 Representação dos valores de dados de medição dos sinais vitais em VSO**

As entidades que representam os valores de dados de medição dos sinais vitais são representados através das subclasses da classe 'measurement datum', que é uma subclasse de *GenericallyDependentContinuant*, a qual foi importada da *Ontology for Biomedical Investigations (OBI)* pela VSO.

Suas subclasses representam os valores das medições de cada sinal vital apresentado anteriormente.

- '*blood pressure measurement datum*' representa a classe das entidades de medidas que ocorrem para um sinal vital pressão arterial.
- '*body temperature measurement datum*' representa a classe das entidades de medidas que ocorrem para um sinal vital temperatura corpórea.
- '*pulse rate measurement datum*' representa a classe das entidades de medidas que ocorrem para um sinal vital frequência de pulso.
- '*respiratory rate measurement datum*' representa a classe das entidades de medidas que ocorrem para um sinal vital frequência respiratória.

O relacionamento entre os valores de dados de medições e os sinais vitais é realizados através dos relacionamentos importados da ontologia OBO-RO pela VSO.

### 3.3 Editor de ontologias Protégé-OWL

*Protégé* é considerada uma das melhores ferramentas utilizadas para a construção e a edição de ontologias. Protégé foi originalmente desenvolvido pelo grupo de modelagem do conhecimento em informática médica da universidade de Stanford, segundo relatado por Martins [24]. Protégé foi desenvolvido como solução para aquisição de conhecimento para aplicações biomédicas, porém, tornou-se uma plataforma independente de domínio de aplicação, sendo um ambiente extensível e integrado para edição e construção de ontologias para sistemas baseados no conhecimento, conforme descrito por Genari et al [16].

Protégé é apoiada por uma comunidade de desenvolvedores e usuários ligados à academia, ao governo e corporativos que estão desenvolvendo aplicações do conhecimento em diversas áreas tais como a biomedicina, engenharia e modelagem corporativa, conforme relatado em [31].

### 3.4 Raciocinador Pellet

Pellet é um raciocinador utilizado para raciocinar sobre ontologias descritas na linguagem OWL e que apoia a completa expressividade da linguagem OWL DL, bem como, o raciocínio sobre construtores nominais (classes enumeradas). Ele é um raciocinador de código aberto implementado na linguagem de programação Java, conforme relatado por Sirin [35]. Pellet fornece funcionalidades para validação de ontologias, checa a consistência de ontologias, classificação de taxonomia, verifica implicações e responde a consultas estabelecidas através das propriedades objeto ou de tipos de dados (datatype) [28].

O raciocinador Pellet pode ser integrado ao editor de ontologias Protégé, através da instalação do plug-in do Protégé. Dessa forma é possível executar as funcionalidades do raciocinador para garantir a integridade da ontologia. Pellet oferece suporte à linguagem de regras SWRL.

### 3.5 Considerações finais

O modelo de informações de contexto SeCoM é composto por ontologias que representam as dimensões semânticas da informação de contexto. A ontologia VSO representa os sinais vitais que são consensuais na comunidade médica. Um dos passos relatados pela metodologia de Noy e McGuinness [25], é considerar o reúso de ontologias existentes, dessa forma, o trabalho proposto importa do modelo SeCoM a ontologia Activity, que representa atividades como eventos espaço-temporais. Assim, além da

expressividade e formalismo oferecidos pela ontologia, é adicionado ao modelo proposto as características relacionadas à informação de contexto.

A ontologia VSO oferece ao modelo proposto, além dos conceitos dos sinais vitais consensuais, a representação das medições de cada sinal, bem como, a estrutura anatômica onde ocorrem as medições dos sinais vitais.

Através da ferramenta de edição e construção de ontologias Protégé é realizada a criação das classes, atributos e relações do modelo ontológico desenvolvido.

Para a inferência e checagem lógica do modelo desenvolvido é utilizado o raciocinador Pellet que oferece um suporte completo para a linguagem OWL e as regras SWRL.

# Modelo Formal para Monitoramento de Sinais Vitais Humanos

---

Este capítulo descreve em detalhes o modelo formal MSVH (Monitoramento de Sinais Vitais Humanos) desenvolvido neste trabalho de pesquisa. O conhecimento representado no modelo MVSH, por meio de ontologias e regras da Web Semântica [7], tem como fundamentação entrevistas e questionários aplicados a médicos intensivistas do Hospital das Clínicas da Universidade Federal de Goiás (HC-UFG), bem como livros das áreas médica e de enfermagem sugeridos pelos entrevistados.

## 4.1 Contextualização

### 4.1.1 Cenário de estudo

Em cada leito de paciente há vários sensores (com fio), como os de frequências de pulso e respiratória, pressão arterial, temperatura, saturação da hemoglobina do sangue arterial, dentre outros. O controle e a parametrização de cada sensor do paciente são realizados pelo médico plantonista em um monitor multiparamétrico, no qual cadastram-se dados do paciente, inclusive valores de referência de sinais daquele paciente. É também por meio da tela desse monitor que o plantonista acessa medições de sinais vitais dos pacientes internados.

Os registros de sinais vitais de pacientes, entretanto, são feitos, na maioria dos casos, de forma manual por enfermeiros ou auxiliares de enfermagem, e em períodos pré-definidos, uma vez que a maioria dos equipamentos de sensoriamento existentes não realiza o armazenamento desses dados. Essas anotações são realizadas em uma ficha impressa individual, cujo conteúdo, dentre outras informações, inclui uma matriz de horários de registro e os tipos de sinais vitais monitorados desse paciente, como descrito na Figura 4.1.

UTI		SISTEMÁTICA DA ASSISTÊNCIA DE ENFERMAGEM														
ADULTO HOSPITAL DAS CLÍNICAS - UFG -		NOME:														
		DIAGNÓSTICO:														
		CIRURGIA:														
DATA: _/_/		DIA DE UTI:				ENF:				MANHÃ:						
		DIA DE PO:				ENF:				TARDE:						
		REGISTRO:				ENF:				NOITE:						
SINAIS VITAIS								Glicemia (mg/dL)	FIO <sub>2</sub> /O <sub>2</sub> (% / L/m)	SaO <sub>2</sub> (%)	Dose de:	Dose de:	Dose de:	Dose de:	Mudança Decúbito	
Hora	T °c	FC bpm	FR ipm	PA mmHg	PVC cmH <sub>2</sub> O	PAM mmHg	PIA cmH <sub>2</sub> O									
07																
08																
09																
10																
11																
12																

**Figura 4.1:** Ficha individual de acompanhamento de pacientes na UTI do HC-UFG (trecho) [5]

Em resposta a anormalidades detectadas nos sinais captados pelos sensores de um paciente, alarmes são produzidos de forma a requisitar a presença da equipe de atendimento para o caso ou chamar um profissional responsável por isso, que deve analisar os sinais vitais exibidos no monitor multiparamétrico daquele paciente, bem como os registros manuais realizados na ficha do paciente.

Algumas vezes podem ocorrer problemas dentro deste contexto, uma vez que funciona de forma manual, por exemplo, a equipe de atendimento sempre se encontra sobrecarregada com as atividades da UTI e podem não perceber os alarmes e também o tempo para localizar um profissional responsável para o atendimento.

#### 4.1.2 Delimitação do escopo

O passo inicial foi escolher uma metodologia para o desenvolvimento de ontologias, a qual foi selecionada a *Ontology Development 101* de Noy e McGuinness [25]. Seguindo a metodologia, como resultado deste trabalho foram definidas mais de 30 questões de competência (vide Apêndice A) para a delimitação do escopo da ontologia, ou seja, questões que a ontologia deveria responder, dentre as quais destacam-se:

1. Que sinais humanos são consensualmente classificados na literatura médica como sinais vitais?

2. Que faixas de valores consensuais para cada sinal vital significam normalidade ou anormalidade?
3. É possível determinar faixas de valores personalizados para cada sinal vital de um paciente?
4. Quais as medições de sinal vital de um paciente em um dado período de tempo?
5. Em que instante de tempo um paciente teve uma medição de sinal vital fora da normalidade individual?

Segundo Potter e Perry [30], “os sinais vitais são uma forma rápida e eficiente de monitorizar a condição de um paciente ou identificar problemas e avaliar a resposta do paciente à intervenção”. Alguma alteração do sinal vital pode indicar alguma alteração na função fisiológica do paciente. Muitos fatores, tais como, temperatura do ambiente, esforço físico, doenças, emoção influenciam na alteração dos sinais vitais, sendo que algumas vezes, fora da faixa de normalidade.

Para responder à questão de competência 1, foram realizadas entrevistas com médicos, enfermeiros intensivistas e a literatura, tendo como resultado a identificação de cinco sinais vitais: temperatura corpórea, frequência de pulso, frequência respiratória, saturação de oxigênio no sangue e pressão arterial.

*Temperatura corpórea* é a diferença entre o calor produzido pelos processos corporais do corpo e o calor perdido pela troca com o ambiente externo.

A temperatura central do corpo, que é a temperatura dos tecidos profundos, se mantém relativamente constante, mesmo com variações da temperatura ambiente e de esforço físico, graças ao aparelho termorregulador. Contudo, a temperatura superficial, temperatura cutânea, sofre alterações dependendo do fluxo sanguíneo da pele e também da perda de calor para o ambiente externo.

A unidade padrão de medida utilizada para a temperatura corpórea, no Brasil, é a de grau Celsius (°C). Uma temperatura considerada aceitável para os seres humanos varia entre 36 à 38 °C, conforme relatado por Potter e Perry [30].

Os locais do corpo para a aferição da temperatura podem sofrer algumas diferenças de valores, estes locais podem ser: oral, retal, axilar, membrana timpânica, arterial pulmonar, esofágico, nasofaringiano e vesical, conforme descrito por Porto e Porto [29].

*Frequência de pulso* é o fluxo sanguíneo sentido pela palpação em várias partes do corpo. Ele indica a circulação sanguínea no corpo que ocorre pelo bombeamento do sangue pelo coração.

Através do pulso pode ser aferido a frequência cardíaca do paciente. Podem ser aferida a frequência cardíaca em qualquer artéria, mas as normalmente indicadas, por serem de fácil palpação, são a radial e a carótida.

A frequência de pulso é medida pelo tempo de 1 minuto que indicará a frequência cardíaca do paciente. A unidade de medida utilizada para a frequência cardíaca é a de batimentos por minuto (bpm).

*Frequência respiratória* é o movimento respiratório que o paciente realiza na inspiração e expiração. O tempo para realizar a aferição da frequência respiratória é de 1 minuto, sendo realizado através da observação e palpação do movimento da parede torácica.

A unidade de medida utilizada para a frequência respiratória é a de movimento de respiração por minuto (mrpm).

*Saturação de oxigênio* no sangue é o percentual de oxigênio encontrado nas moléculas de hemoglobina do sangue.

A aferição da saturação de oxigênio é realizado pelo aparelho conhecido como oxímetro de pulso, que é uma sonda com um diodo que emite luz (DEL) e fotodetector que está conectado por um cabo a um oxímetro. O DEL emite comprimentos de luz que são absorvidos diferentemente pelas moléculas de hemoglobinas oxigenadas e desoxigenadas e, então, o fotodetector obtém a diferença da absorção da luz, e o oxímetro calcula a saturação de oxigênio.

O local para fazer a aferição da saturação de oxigênio, pelo oxímetro de pulso, pode ser os dedos ou os lóbulos da orelha. A unidade de medida utilizada para a saturação de oxigênio é a de percentual (%).

*Pressão arterial* é a força exercida nas paredes das artérias pelo sangue que pulsa em decorrência do bombeamento do coração fazendo o sangue circular por todo o corpo.

Quando o coração faz o movimento de contração, ele expulsa o sangue de seu interior, e é nesse momento que a pressão nas artérias atinge o seu valor máximo, conhecida como *pressão máxima* ou *sistólica*.

Quando o coração faz o movimento de relaxamento da musculatura para permitir que o sangue volte a encher o seu interior a pressão nas artérias diminui atingindo valores mínimos, conhecida como *pressão mínima* ou *diastólica*.

A aferição da pressão arterial é realizada, normalmente, de forma não invasiva, por um aparelho chamado esfigmomanômetro colocado no braço do paciente. A leitura da pressão arterial é apresentada com o valor da pressão sistólica antes da pressão diastólica, por exemplo, 120/80. A unidade padrão de medida utilizada para a pressão arterial é a de milímetros de mercúrio (mmHg).

Concluído esse levantamento, os profissionais de Saúde entrevistados recomendaram o estudo de livros clássicos de Medicina de Porto e Porto [29] e de Enfermagem de Potter e Perry [30] para que se pudesse responder à questão de competência 2. Para cada um dos sinais vitais identificados, foram coletados os valores de referência utilizados pelas comunidades mencionadas.

Abaixo apresentamos, ilustrada através da Figura 4.2, a classificação da temperatura corpórea para adultos, segundo descrito por Porto e Porto [29], tomando como referência o nível da temperatura axilar.

Classificação	Temperatura (°C)
Temperatura axilar (Normal)	35,5 – 37
Febre leve ou febrícula	37,1 – 37,5
Febre moderada	37,6 – 38,5
Febre alta	> 38,5
Hipotermia leve	32 – 35
Hipotermia moderada	30 – 31,9
Hipotermia grave	< 30

**Figura 4.2:** *Classificação da temperatura corpórea [29]*

Segundo Potter e Perry [30], na Figura 4.3 é apresentada a classificação da frequência de pulso para adultos.

Classificação	Batimentos por minuto (bpm)
Normal	60 - 100
Bradycardia	< 60
Taquicardia	> 100

**Figura 4.3:** *Classificação da frequência de pulso [30]*

Na Figura 4.4 é apresentada a classificação da frequência respiratória para adultos, segundo Potter e Perry [30].

Classificação	Movimento de respiração por minuto (mrpm)
Eupnéia	12 - 20
Bradipnéia	< 12
Taquipnéia	> 20
Apnéia	= 0

**Figura 4.4:** Classificação da frequência respiratória [30]

Na Figura 4.5 é ilustrada a classificação da saturação de oxigênio para adultos, segundo Potter e Perry [30].

Classificação	Percentagem da Saturação de Oxigênio (%)
Normal	95 – 100
Hipoxemia	< 95

**Figura 4.5:** Classificação da saturação de oxigênio [30]

Na Figura 4.6 é apresentada a tabela da classificação da pressão arterial para adultos, segundo as VI Diretrizes Brasileiras de Hipertensão Arterial de 2010 da Sociedade Brasileira de Cardiologia et al [13].

Através da interação com profissionais da saúde constatou-se também que, com exceção da temperatura corpórea, as faixas de valores de sinais vitais podem variar de um paciente para outro, objeto da questão de competência 3. Por exemplo, um alarme não deveria ser emitido se um paciente pré-hipertenso tiver leituras contínuas de 130-85 mmHg, respectivamente, para pressão arterial sistólica e diastólica, já que essas leituras são consideradas normais para esse paciente. O mesmo se aplica a pacientes com longo histórico de tabagismo, que podem apresentar saturação de oxigênio no sangue inferior a 95%, valor dado como normal para esses pacientes.

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

Quando as pressões sistólica e diastólica situam-se em categorias diferentes, a maior deve ser utilizada para classificação da pressão arterial.

\* Pressão normal-alta ou pré-hipertensão são termos que se equivalem na literatura.

**Figura 4.6:** Classificação da pressão arterial de acordo com a medida casual no consultório (> 18 anos) [13]

As questões de competência 4 e 5 reforçam que a ontologia necessita registrar temporalmente cada medição de sinal vital de um paciente para posterior consulta, o que pode apoiar a tomada de decisão de intensivistas, por exemplo, principalmente para os instantes em que houve alguma medição cujo valor é considerado anormal para um dado paciente.

A seção seguinte descreverá em detalhes a construção do modelo MSVH.

## 4.2 Ontologia MSVH

A característica de formalidade atribuída ao modelo MSVH foi dada pela utilização de ontologias e regras da lógica de predicados. Após a delimitação do escopo da ontologia, como propõe a metodologia seguida por Noy e McGuinness [25], foram reutilizadas ontologias existentes, aplicáveis ao mesmo domínio de aplicação, e que reduzem o tempo e o esforço da construção, em geral. A ontologia MSVH importou do modelo SeCoM a ontologia Activity para ser estendida pelo conceito de monitoramento de sinais vitais. A ontologia Activity importa as demais ontologias do modelo SeCoM para descrever ações realizadas por atores em um ambiente de computação sensível ao contexto.

Após estudo das ontologias a serem reutilizadas e da elaboração das questões de competência, apresentadas no Apêndice A, para a ontologia MSVH, iniciou-se a

construção da ontologia proposta com suas classes e respectivas propriedades, relações, axiomas e indivíduos, como sugere a metodologia 101, seguindo a sintaxe da linguagem OWL [36] [6], que é um padrão W3C para construção de ontologias.

O objetivo da ontologia MSVH é descrever atividades de monitoramento de sinais vitais de pacientes, tais como, o registro das medições dos sinais vitais, o acompanhamento automatizado destas medições, que possam apoiar o trabalho de assistência dos profissionais da área de saúde ao cuidado de pacientes monitorados.

No diagrama UML da Figura 4.7 estão ilustradas as classes principais, atributos e relações da ontologia MSVH. As demais classes serão apresentadas no decorrer desta seção. É possível verificar na Figura 4.7 a importação da classe *sActivity:Activity* que pertence à ontologia Activity, ilustrada na Figura 3.7 da Subseção 3.1.6, e da classe *sActor:Actor* e sua subclasse *sActor:Person* que pertencem a ontologia Actor, ilustrada na Figura 3.6 da Subseção 3.1.5, ambas as ontologias são relacionadas ao modelo SeCoM, e das classes: *vso:'vital sign'*, *vso:'Anatomical entity'* e *vso:'measurement datum'*, que pertencem à ontologia importada VSO. Os espaços de nomes *sActivity:*, *sActor:* e *vso:*, respectivamente, correspondem às ontologias Activity e Actor do modelo SeCoM e à ontologia VSO.

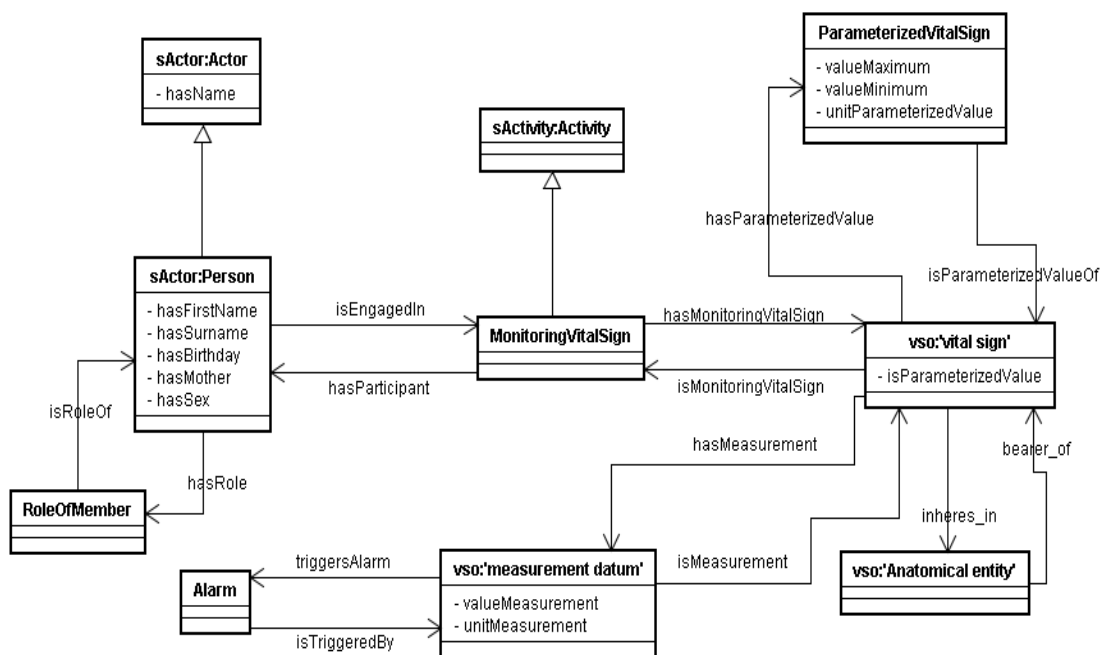


Figura 4.7: Principais conceitos da ontologia MSVH

Na ontologia MSVH, ilustrada na Figura 4.7, a principal classe é a *MonitoringVitalSign*, que é uma extensão da classe *sActivity:Activity*. A classe *sActivity:Activity* modela entidades de atividades como eventos espaço-temporais. Dessa forma, *MonitoringVitalSign*, que herda de *sActivity:Activity* todos os atributos e relacionamentos espaço-temporais, consegue descrever entidades de atividades de monitoramento de sinais vitais

do paciente em relação ao tempo, através da classe *InstantEvent* da Figura 3.3, e à localização, através das subclasses de *PhysicalLocation* da Figura 3.4, em que o monitoramento de sinais vitais ocorre.

Além disso, como a ontologia Activity importa todas as outras ontologias do modelo SeCoM, ela possui entidades relacionadas aos atores participantes das atividades modeladas, assim, por meio da relação *hasParticipant* e de sua relação inversa *isEngagedIn*, permite que indivíduos da classe *MonitoringVitalSign* possam se relacionar com indivíduos da subclasse de *sActor:Actor* chamada *sActor:Person*, que descreve os atores do tipo pessoa por meio de seus atributos *hasName*, *hasSurname*, *hasFirstName*, *hasBirthday*, *hasMother* e *hasSex*, para representar os atores que participam do monitoramento de sinais vitais do paciente, sendo que os dois últimos atributos foram incluídos na ontologia MSVH e portanto, não estão presentes na ontologia Actor original. Portanto, é possível relacionar cada medição de sinal vital a um paciente, em um dado instante de tempo, em uma dada localização, e a equipe de profissionais de saúde que assiste esse paciente. Dessa forma, são atendidas as questões de competência 23, 35 e 37 do Apêndice A.

A classe *RoleOfMember* descreve os papéis, tais como, médico, enfermeiro e paciente. O relacionamento *hasRole* e sua relação inversa *isRoleOf* entre as classes *sActor:Person* e *RoleOfMember*, permitem representar quais papéis que os atores assumem dentro do monitoramento de sinais vitais.

A classe *vso:'vital sign'* é uma classe da ontologia VSO que descreve indivíduos que representam os sinais vitais que são consenso entre os membros da comunidade da área da Saúde. Por meio dessa classe é possível atender a questão de competência 21.

A classe *MonitoringVitalSign* está relacionada por meio da relação *hasMonitoringVitalSign* e da sua relação inversa *isMonitoringVitalSign*, com a classe *vso:'vital sign'*. Dessa forma, é possível saber qual sinal vital do paciente está sendo monitorando. Além disso, através do relacionamento *inheres\_in* e da sua relação inversa *bearer\_of* que descreve a estrutura anatômica do corpo humano em que o sinal vital ocorre por meio da classe *vso:'Anatomical entity'*, é possível saber em qual local do corpo foi realizada a medição do sinal vital, assim, são atendidas as questões de competência 31, 32, 33, 34 e 36.

A classe *vso:'measurement datum'*, que pertence à ontologia VSO e que descreve indivíduos que reapresentam os valores das medições através de seus atributos *valueMeasurement* e *unitMeasurement xsd : string*, também mantém relacionamentos, por meio da relação *hasMeasurement* e da sua relação inversa *isMeasurement*, com a classe *'vital sign'*. Com isso, é possível associar as medições com os sinais vitais que estão sendo monitorados. Dessa forma, são atendidas as questões de competência 1, 2, 3, 4, 5, 16, 17, 18, 19, 20, 24, 26, 27, 28, 29 e 30.

Ainda é possível verificar na Figura 4.7 o relacionamento, por meio da relação

*triggersAlarm* e da sua relação inversa *isTriggeredBy*, entre a classe *vso:'measurement datum'* e a classe *Alarm*, que representa os indivíduos que tiveram os valores de medições considerados anormais dentro de uma classificação pré-definida pela comunidade da área de saúde. Assim, por meio dessas relações, é possível que alarmes sejam disparados quando os valores das medições ficarem fora dos intervalos de normalidade das tabelas de classificações dos sinais vitais, e dessa forma, são atendidas as questões de competência 6, 7, 8, 9, 10, 11, 12, 13, 14 e 15.

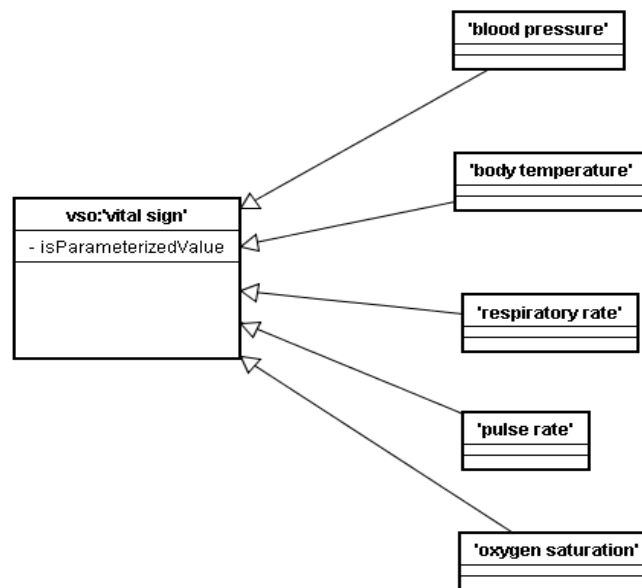
A classe *ParameterizedVitalSign* representa, por meio de seus atributos *valueMaximum xsd : nonNegativeInteger*, *valueMinimum xsd : nonNegativeInteger* e *unitParameterizedValue xsd : string*, valores de referência individualizados. A classe *'vital sign'* associada à classe *ParameterizedVitalSign*, por meio da relação *hasParameterizedValue* e da sua relação inversa *isParameterizedValue*, representa valores de referência individualizados para o sinal vital. Para a classe *'vital sign'* foi criado um atributo chamado *isParameterizedValue* cujo valor assume a notação do tipo de dados booleano *xsd : boolean*, este atributo não existe na classe *'vital sign'* da VSO original. Quando esse atributo é setado com o valor de verdadeiro, significa que o sinal vital não terá como parâmetro apenas os valores de referência das tabelas de classificações pré-definidas pela comunidade da área da Saúde como normais, mas também os valores individualizados da classe *ParameterizedVitalSign*. Dessa forma, apenas serão disparados alarmes quando as medições não se encontrarem dentro do intervalo de valores parametrizados, e assim, são atendidas as questões de competência 22 e 25.

## Descrição das classes da ontologia de MSVH

Nesta seção, são descritas as classes, atributos e relações que foram desenvolvidas/reutilizadas conforme as questões de competência do Apêndice A elaboradas para atender os objetivos da ontologia MSVH.

### 4.2.1 Descrição da classe *vso:'vital sign'*

A classe *vso:'vital sign'* descreve os sinais vitais que são consenso na literatura médica e que representam o sinal vital que está envolvido no monitoramento. Na ontologia MSVH foi criada uma subclasse de *vso:'vital sign'*, chamada *'oxygen saturation'*, para descrever o sinal vital de saturação de oxigênio e que não existe na ontologia VSO original. A Figura 4.8 ilustra as subclasses da classe *vso:'vital sign'*.



**Figura 4.8:** Subclasses da classe *vso:'vital sign'*

As subclasses *'blood pressure'*, *'body temperature'*, *'respiratory rate'*, *'pulse rate'* e *'oxygen saturation'* descrevem, respectivamente, os sinais vitais que representam a pressão arterial, a temperatura corpórea, frequência respiratória e de pulso e a saturação de oxigênio.

Para todas as subclasses de *vso:'vital sign'* foram criadas subpropriedades derivadas das relações que *vso:'vital sign'* mantém com as outras classes, conforme está ilustrada na Figura 4.7. Por exemplo, se algum sinal vital de frequência respiratória tiver o atributo *isParameterizedValue* marcado como verdadeiro, então, este sinal vital terá um relacionamento, *hasParameterizedValueRespiratoryRate*, com um indivíduo da subclasse *RespiratoryRate* de *ParameterizedVitalSign*, e assim, poderá ocorrer com os demais sinais vitais.

## 4.2.2 Descrição da classe *ParameterizedVitalSign*

A classe *ParameterizedVitalSign* descreve valores individualizados e suas unidades de medida utilizadas como referência de normalidade para as medições dos sinais vitais. Algumas pessoas possuem doenças preexistentes que poderiam disparar alarmes falsos para a equipe de saúde, por exemplo, pessoas que fumaram durante um longo período de tempo podem ter o percentual de saturação de oxigênio abaixo de 95%, o que tornar-se um valor de medição normal para aquela pessoa. A Figura 4.9 apresenta as subclasses da classe *ParameterizedVitalSign*. Foram criadas para as subclasses subpropriedades derivadas das relações que *ParameterizedVitalSign* mantém com *vso:'vital sign'*.

As subclasses de *ParameterizedVitalSign* são representadas por seus atributos de valor máximo, valor mínimo e unidade de medida. Por exemplo, através do relaciona-

mento, *isParameterizedValueOfPulseRate*, com a subclasse 'pulse rate' de *vso:'vital sign'* é possível individualizar o valor de referência para o sinal vital de frequência de pulso para um determinado paciente, para isso, deverão ser atribuídos valores para os atributos *valueMaximumPulseRate*, *valueMinimumPulseRate* e *unitParameterizedValuePulseRate* da subclasse *PulseRate* de *ParameterizedVitalSign*. A individualização dos valores de referência poderá ocorrer para os sinais vitais da classe *vso:'vital sign'*, com exceção do sinal vital 'body temperature', que não apresenta alteração em relação à classificação da temperatura corpórea ilustrada na Figura 4.2.

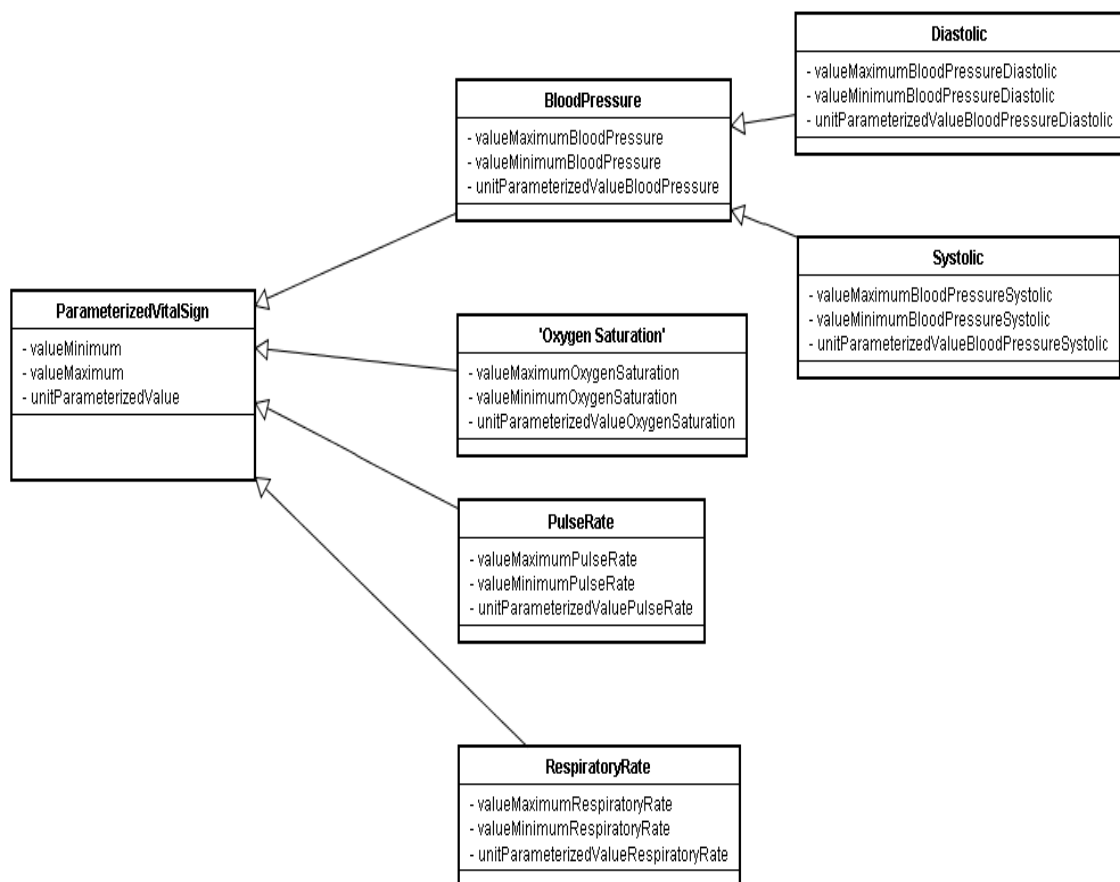
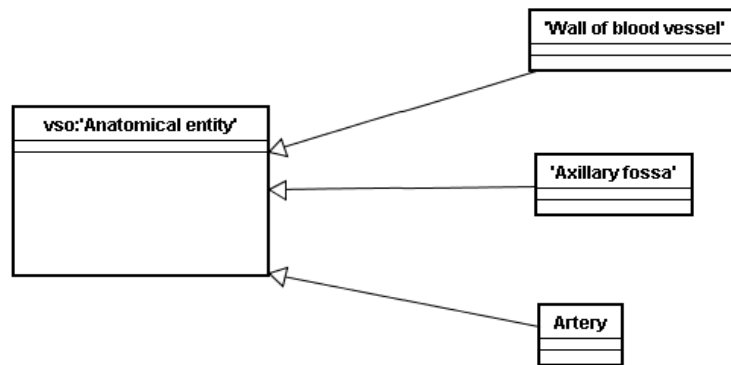


Figura 4.9: Subclasses da classe *ParameterizedVitalSign*

Os atributos das subclasses *BloodPressure*, *Diastolic*, *Systolic*, *RespiratoryRate*, *PulseRate* e '*OxygenSaturation*' são subatributos dos atributos da classe *ParameterizedVitalSign*.

### 4.2.3 Descrição da classe *vso:'Anatomical entity'*

A classe *vso:'Anatomical entity'* descreve a localização da estrutura anatômica humana em que a medição do sinal vital ocorre. A Figura 4.10 apresenta as subclasses da classe *vso:'Anatomical entity'* que foram utilizadas na ontologia MSVH.



**Figura 4.10:** Subclasses da classe *vso: 'Anatomical entity'*

A subclasse *'Wall of blood vessel'* descreve localizações da estrutura anatômica humana em que a medição do sinal vital da pressão arterial pode ocorrer. Ela mantém o seguinte relacionamento:

- relacionamento *bearer\_of* e sua relação inversa *inheres\_in* com a subclasse *blood pressure* de *vso: 'vital sign'*, que representa a localização da estrutura anatômica humana que ocorre a medição para o sinal vital de pressão arterial.

A subclasse *'Axillary fossa'* descreve a localização da estrutura anatômica humana em que a medição do sinal vital da temperatura corpórea da axila pode ocorrer. Ela mantém o seguinte relacionamento:

- relacionamento *bearer\_of* e sua relação inversa *inheres\_in* com a subclasse *'axillary temperature'* de *'body temperature'*, que representa a localização da estrutura anatômica humana que ocorre a medição para o sinal vital de temperatura corpórea axilar.

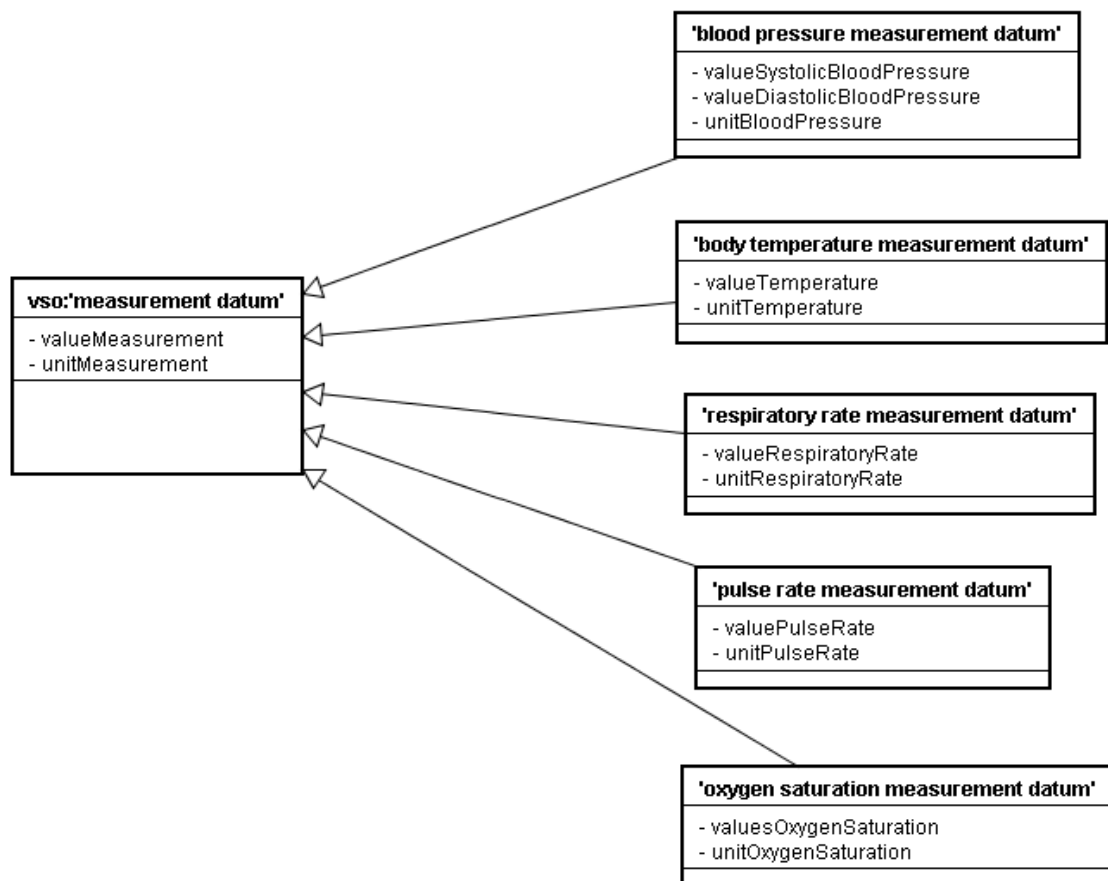
A subclasse *'Artery'* descreve localizações da estrutura anatômica humana em que a medição do sinal vital da frequência de pulso pode ocorrer. Ela mantém o seguinte relacionamento:

- relacionamento *bearer\_of* e sua relação inversa *inheres\_in* com a subclasse *'pulse rate'* de *vso: 'vital sign'*, que representa a localização da estrutura anatômica humana que ocorre a medição para o sinal vital de frequência de pulso.

Outras localizações da classe *vso: 'Anatomical entity'* podem ser utilizadas da classe para expressar as localizações em que ocorrem as medições dos sinais vitais.

#### 4.2.4 Descrição da classe *vso:'measurement datum'*

A classe *vso:'measurement datum'* descreve valores e unidades de medida para as medições de sinais vitais que ocorrem em um monitoramento por meio de seus atributos *valueMeasurement* e *unitMeasurement* (xsd:string). Na ontologia MSVH foi criada uma subclasse de *vso:'measurement datum'*, chamada '*oxygen saturation measurement datum*', para descrever valores e unidades de medida para a medição do sinal vital de saturação de oxigênio e que não existe na ontologia VSO original. A Figura 4.11 ilustra as subclasses da classe *vso:'measurement datum'*. Foram criadas para as subclasses subpropriedades derivadas das relações que *vso:'measurement datum'* mantém com as classes *vso:'vital sign'* e *Alarm*, a qual está representada na Figura 4.7.



**Figura 4.11:** Subclasses da classe *vso:'measurement datum'*

A subclasse '*blood pressure measurement datum*' descreve as medições para o sinal vital pressão arterial por meio de seus atributos *valueDiastolicBloodPressure* (xsd:nonNegativeInteger), *valueSystolicBloodPressure* (xsd:nonNegativeInteger) e *unitBloodPressure* (xsd:string).

A subclasse '*body temperature measurement datum*' descreve as medições para o sinal vital temperatura corpórea por meio de seus atributos *valueTemperature* (xsd:float)

e *unitTemperature* (xsd:string).

A subclasse '*respiratory rate measurement datum*' descreve as medições para o sinal vital frequência respiratória por meio de seus atributos *valueRespiratoryRate* (xsd:nonNegativeInteger) e *unitRespiratoryRate* (xsd:string).

A subclasse '*pulse rate measurement datum*' descreve as medições para o sinal vital frequência de pulso por meio de seus atributos *valuePulseRate* (xsd:nonNegativeInteger) e *unitPulseRate* (xsd:string).

A subclasse '*oxygen saturation measurement datum*' descreve as medições para o sinal vital saturação do oxigênio por meio de seus atributos *valueOxygenSaturation* (xsd:nonNegativeInteger) e *unitOxygenSaturation* (xsd:string).

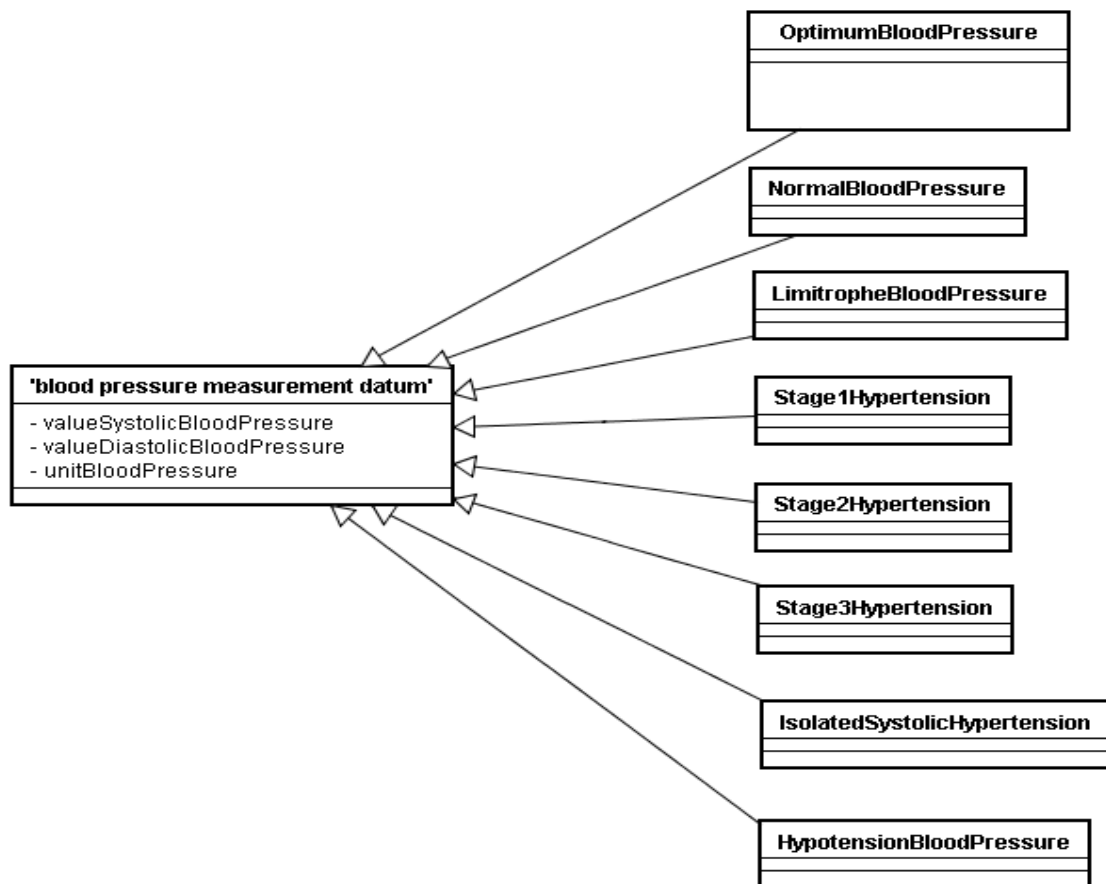
Os atributos das subclasses '*blood pressure measurement datum*', '*body temperature measurement datum*', '*respiratory rate measurement datum*', '*pulse rate measurement datum*' e '*oxygen saturation measurement datum*' são subatributos derivados dos atributos da classe *vso:'measurement datum'*.

Portanto, para cada sinal vital de *vso:'vital sign'* haverá um relacionamento com as subclasses de *vso:'measurement datum'*. Por exemplo, para um sinal vital de pressão arterial existe um relacionamento, *hasMeasurementBloodPressure*, com um indivíduo da subclasse '*blood pressure measurement datum*' de *vso:'measurement datum'* que irá guardar o valor e a unidade da medição do sinal vital de pressão arterial.

Cada subclasse de *vso:'measurement datum'* possui subclasses que representam a classificação das medições conforme os valores de referências estabelecido pela literatura médica.

### **Descrição da subclasse '*blood pressure measurement datum*'**

As subclasses de '*blood pressure measurement datum*' foram definidas conforme as categorias da tabela de classificação da pressão arterial apresentadas na Figura 4.6. A Figura 4.12 ilustra estas subclasses.



**Figura 4.12:** Subclasses de *'blood pressure measurement datum'*

Cada subclasse de *'blood pressure measurement datum'* representa as medições que se encontram dentro do intervalo de valor para cada categoria da tabela de classificação da pressão arterial, ilustrada na Figura 4.6. Por exemplo, a medição do sinal vital de pressão arterial que se encontra com o valor da pressão sistólica entre 140 e 149 mmHg e o valor da pressão diastólica entre 90 a 99 mmHg, esta medição será classificada dentro da subclasse *Stage1Hypertension*.

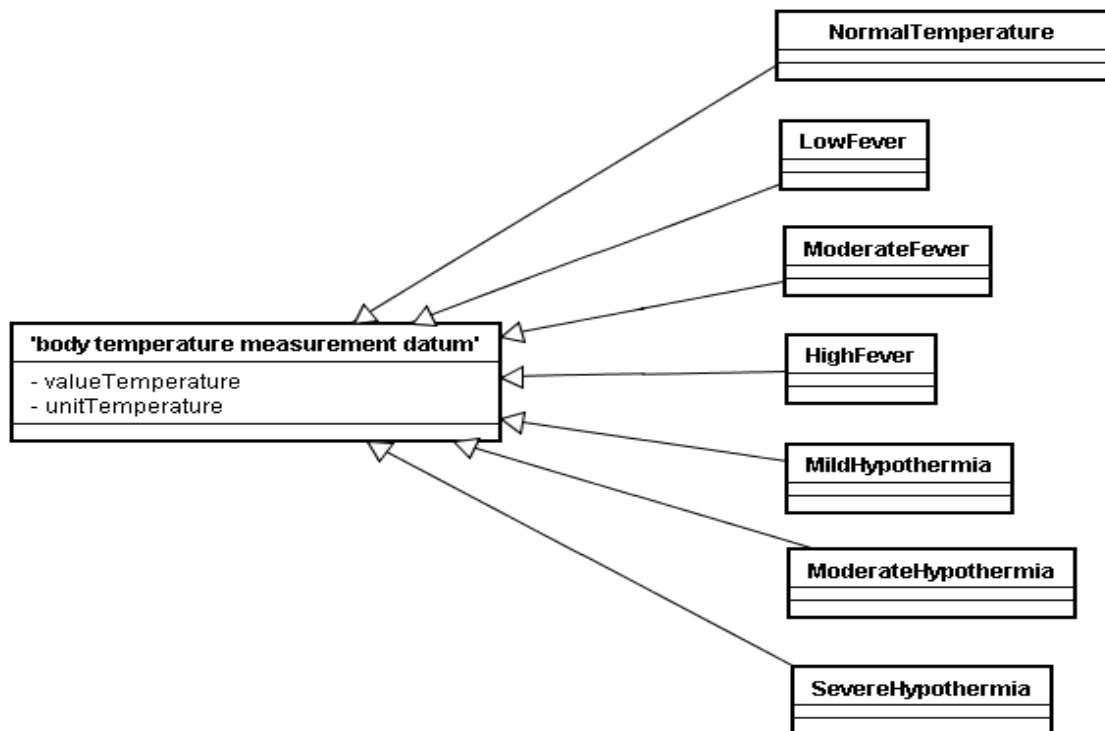
A subclasse *HypotensionBloodPressure* representa os indivíduos de medição classificados com a pressão arterial hipotensão e que tem os seguintes valores de medição: *pressão sistólica menor que 90 mmHg e pressão diastólica menor que 60 mmHg*.

Quando as medições das pressões sistólica ou diastólica se situarem em categorias diferentes, a pressão que apresentar o maior valor irá definir em quais categorias será classificada a medição.

### Descrição da subclasse *'body temperature measurement datum'*

As subclasses de *'body temperature measurement datum'* foram definidas com base nas categorias da tabela de classificação da temperatura corpórea apresentadas na

Figura 4.2, tomando como referência o nível da temperatura axilar. A Figura 4.13 ilustra estas subclasses.

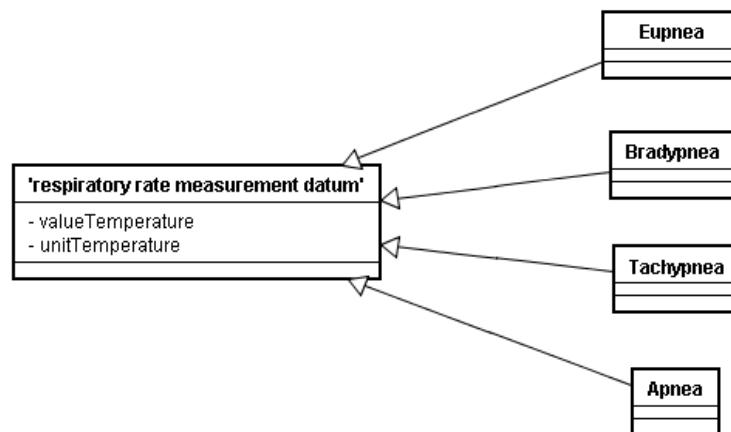


**Figura 4.13:** Subclasses de *'body temperature measurement datum'*

As subclasses de *'body temperature measurement datum'* representam as medições que se encontram dentro do intervalo de valor para cada categoria da tabela de classificação da temperatura corpórea, ilustrada na Figura 4.2. Por exemplo, quando algum valor da medição do sinal vital de temperatura corpórea for maior que 38,5 °C esta medição será classificada dentro da subclasse *HighFever*.

### **Descrição da subclasse *'respiratory rate measurement datum'***

As subclasses de *'respiratory rate measurement datum'* foram definidas conforme as categorias da tabela de classificação da frequência respiratória ilustradas pela Figura 4.4. A Figura 4.14 ilustra estas subclasses.

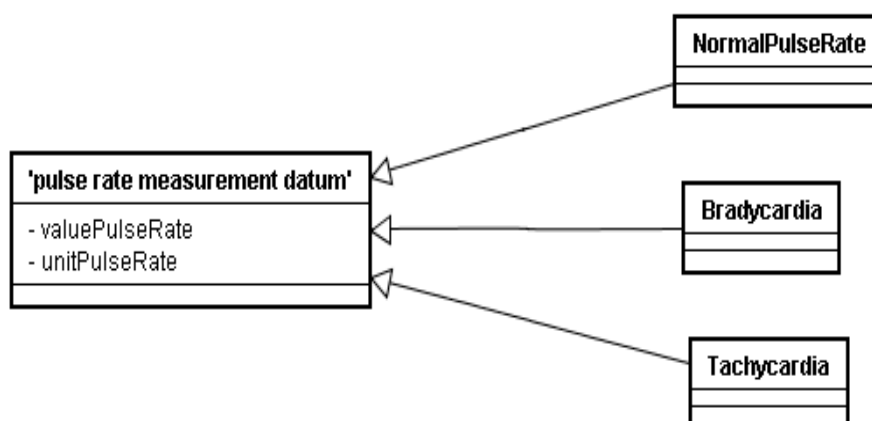


**Figura 4.14:** Subclasses de *'respiratory rate measurement datum'*

As subclasses de *'respiratory rate measurement datum'* representam as medições que se encontram dentro do intervalo de valor para cada categoria da tabela de classificação da frequência respiratória, ilustrada pela Figura 4.4. Por exemplo, se o valor da medição do sinal vital de frequência respiratória for menor que 12 mrpm esta medição será classificada dentro da subclasse *Bradypnea*.

#### Descrição da subclasse *'pulse rate measurement datum'*

As subclasses de *'pulse rate measurement datum'* foram baseadas conforme as categorias da tabela de classificação da frequência de pulso ilustradas na Figura 4.3. A Figura 4.15 ilustra estas subclasses.



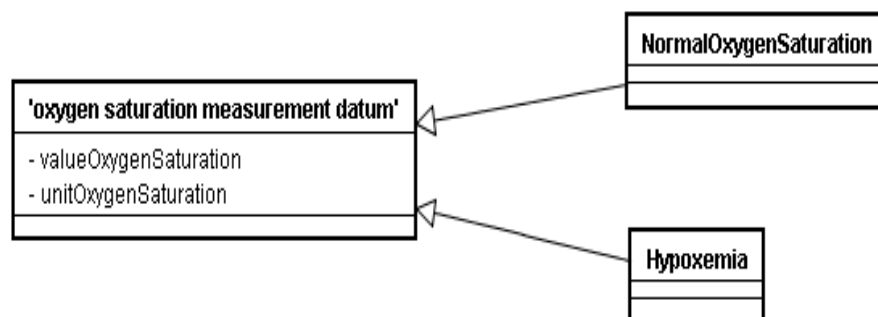
**Figura 4.15:** Subclasses de *'pulse rate measurement datum'*

As subclasses de *'pulse rate measurement datum'* representam as medições que se encontram dentro do intervalo de valor para cada categoria da tabela de classificação da frequência respiratória, ilustrada pela Figura 4.3. Por exemplo, se o valor da medição

do sinal vital de frequência de pulso for maior que 100 bpm esta medição será classificada dentro da subclasse *Tachycardia*.

### Descrição da subclasse 'oxygen saturation measurement datum'

As subclasses de 'oxygen saturation measurement datum' foram definidas conforme as categorias da tabela de classificação da saturação do oxigênio apresentadas pela Figura 4.5. A Figura 4.16 ilustra estas subclasses.

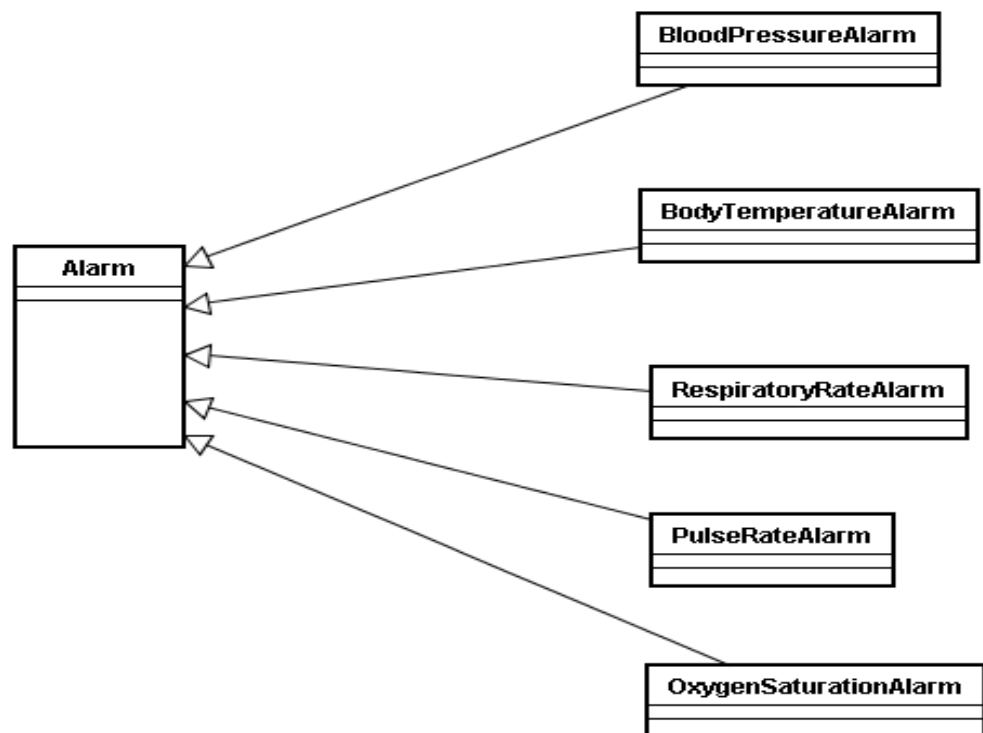


**Figura 4.16:** Subclasses de 'oxygen saturation measurement datum'

As subclasses de 'oxygen saturation measurement datum' representam as medições que se encontram dentro do intervalo de valor para cada categoria da tabela de classificação da saturação de oxigênio, ilustrada pela Figura 4.5. Por exemplo, se o valor da medição do sinal vital de saturação de oxigênio for menor que 95 % esta medição será classificada dentro da subclasse *Hypoxemia*.

### 4.2.5 Descrição da classe Alarm

A classe *Alarm* que descreve os alarmes que são disparados quando a medição do sinal vital se encontra fora do intervalo de normalidade, possui subclasses que descrevem os alarmes para cada medição do sinal vital. O disparo de alarmes é para alertar a equipe de saúde sobre possíveis ocorrências anormais dos valores de referência, que implicam em alguma mudança na função fisiológica do paciente, e assim, a equipe intervir através de ações de procedimentos para retornar os sinais vitais para valores aceitáveis. A Figura 4.17 ilustra as subclasses da classe *Alarm*.



**Figura 4.17:** Subclasses da classe *Alarm*

A subclasse *BloodPressureAlarm* descreve os alarmes que são disparados quando a medição do sinal vital se encontram fora do intervalo de normalidade para o sinal vital pressão arterial.

A subclasse *BodyTemperatureAlarm* descreve os alarmes que são disparados quando a medição do sinal vital se encontra fora do intervalo de normalidade para o sinal vital temperatura corpórea.

A subclasse *RespiratoryRateAlarm* descreve os alarmes que são disparados quando a medição do sinal vital se encontra fora do intervalo de normalidade para o sinal vital frequência respiratória.

A subclasse *PulseRateAlarm* descreve os alarmes que são disparados quando a medição do sinal vital se encontra fora do intervalo de normalidade para o sinal vital frequência de pulso.

A subclasse *OxygenSaturationAlarm* descreve os alarmes que são disparados quando a medição do sinal vital se encontra fora do intervalo de normalidade para o sinal vital saturação do oxigênio.

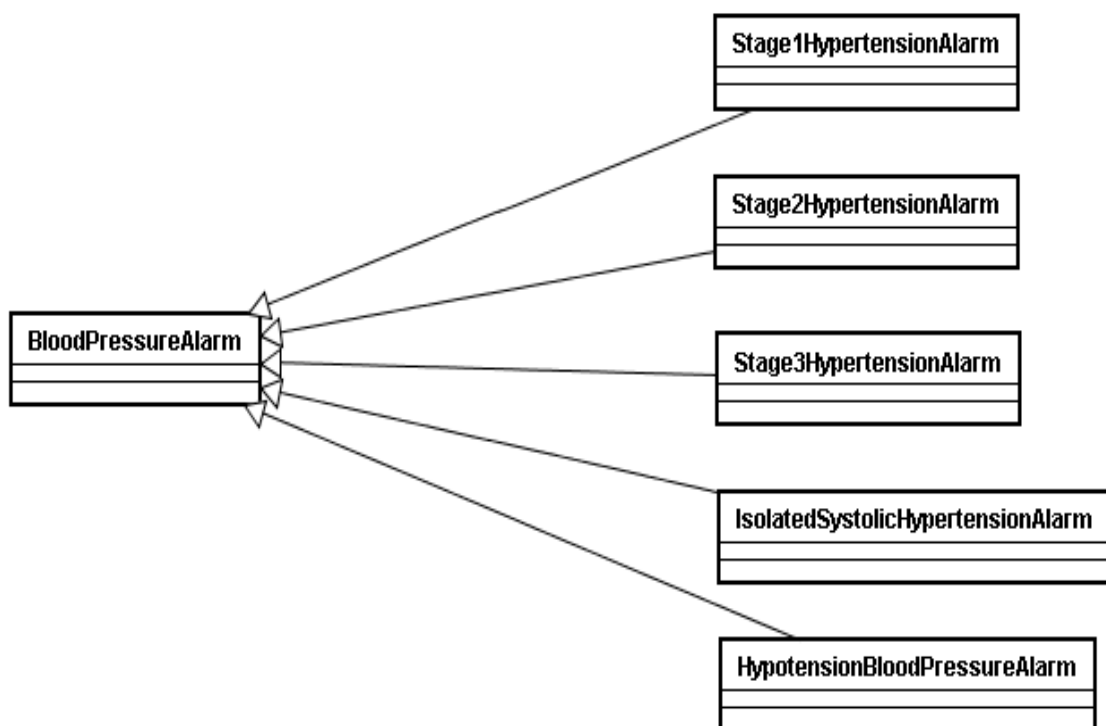
As subclasses de *Alarm* mantêm relacionamento, por meio da relação *isTriggeredBy*, com as subclasses de *vso: 'measurement datum'* que descreve as medições dos sinais vitais que ficaram fora dos valores de normalidade apresentados nas tabelas de classificações ou parametrizados individualmente por paciente. Por exemplo, se uma medição de temperatura corpórea de um paciente estiver fora do valor de normalidade será dispa-

rado um alarme, representado através de uma regra SWRL do Apêndice B , que irá inferir o alarme na subclasse *BodyTemperatureAlarm* de *Alarm*.

Cada subclasse de *Alarm* possui subclasses que representam a classificação dos alarmes conforme os valores de referências estabelecidos pela literatura médica.

### Descrição da subclasse *BloodPressureAlarm*

As subclasses de *BloodPressureAlarm* representam as categorias da tabela de classificação da pressão arterial apresentadas na Figura 4.6. As subclasses de *BloodPressureAlarm* são apresentadas na Figura 4.18.



**Figura 4.18:** Subclasses de *BloodPressureAlarm*

A subclasse *HypotensionBloodPressureAlarm* representa os indivíduos de alarme classificados com a pressão arterial hipotensão e que tem os seguintes valores de medição: *pressão sistólica menor que 90 mmHg e pressão diastólica menor que 60 mmHg*.

As subclasses de *BloodPressureAlarm* representam as categorias da tabela de classificação da pressão arterial cujos intervalos de valores se encontram fora da faixa de normalidade, conforme ilustrada pela Figura 4.6. Por exemplo, uma medição de sinal vital de pressão arterial que se encontra com o valor da pressão sistólica entre 160 e 179 mmHg e o valor da pressão diastólica entre 100 a 109 mmHg, esta medição irá disparar um alarme, representado através de uma regra SWRL do Apêndice B , que irá classificar o alarme na subclasse *Stage2Hypertension* de *BloodPressureAlarm*.

Quando os alarmes das pressões sistólica ou diastólica se situarem em categorias diferentes, a pressão que apresentar o maior valor irá definir em qual categoria será classificado o alarme.

### Descrição da subclasse *BodyTemperatureAlarm*

As subclasses de *BodyTemperatureAlarm* foram definidas conforme as categorias da tabela de classificação da temperatura corpórea apresentadas na Figura 4.2. A Figura 4.19 ilustra estas subclasses.

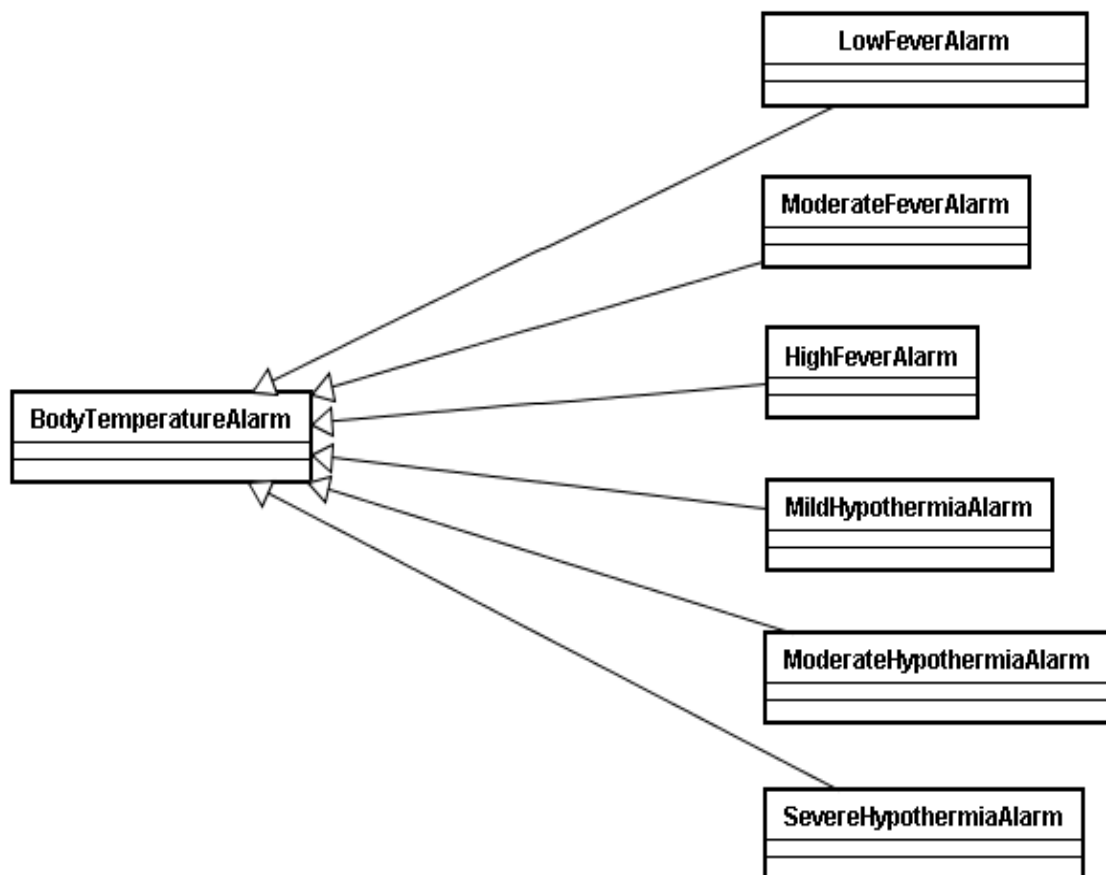


Figura 4.19: Subclasses de *BodyTemperatureAlarm*

As subclasses de *BodyTemperatureAlarm* foram criadas para representar as categorias da tabela de classificação da temperatura corpórea cujos intervalos de valores se encontram fora da faixa de normalidade, conforme ilustrada pela Figura 4.2. Por exemplo, uma medição de sinal vital de temperatura corpórea que se encontra com o valor maior que 38,5 °C, esta medição irá disparar um alarme, representado através de uma regra SWRL do Apêndice B, que irá classificar o alarme na subclasse *HighFeverAlarm* de *BodyTemperatureAlarm*.

### Descrição da subclasse *RespiratoryRateAlarm*

As subclasses de *RespiratoryRateAlarm* representam as categorias da tabela de classificação da frequência respiratória apresentadas na Figura 4.4. As subclasses de *RespiratoryRateAlarm* são apresentadas na Figura 4.20.

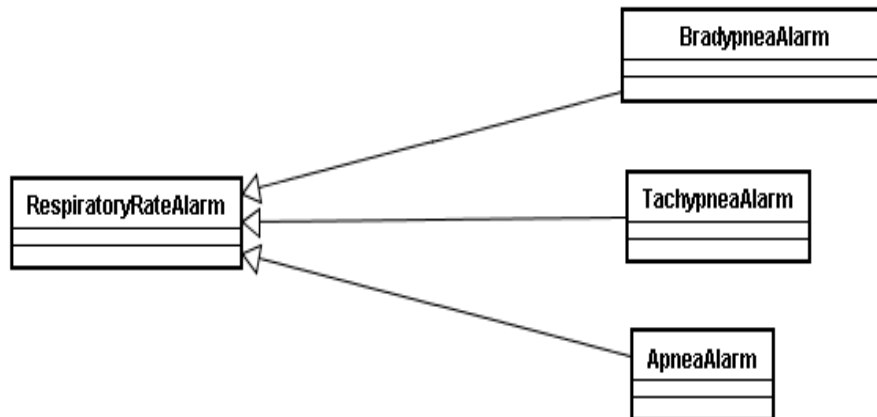


Figura 4.20: Subclasses de *RespiratoryRateAlarm*

As subclasses de *RespiratoryRateAlarm* representam as categorias da tabela de classificação da frequência respiratória cujos intervalos de valores se encontram fora da faixa de normalidade, conforme ilustrada pela Figura 4.4. Por exemplo, uma medição de sinal vital de frequência respiratória que se encontra com o valor menor que 12 mrpm, esta medição irá disparar um alarme, representado através de uma regra SWRL do Apêndice B , que irá classificar o alarme na subclasse *BradypneaAlarm* de *RespiratoryAlarm*.

### Descrição da subclasse *PulseRateAlarm*

As subclasses de *PulseRateAlarm* foram criadas conforme as categorias da tabela de classificação da frequência de pulso apresentadas na Figura 4.3. A Figura 4.21 ilustra estas subclasses.

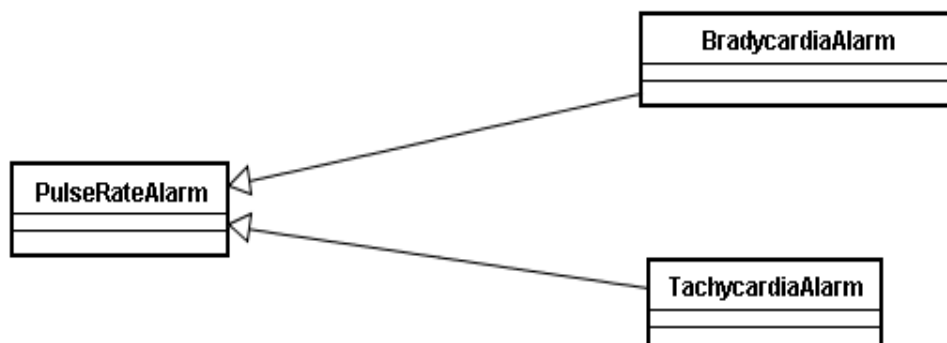


Figura 4.21: Subclasses de *PulseRateAlarm*

As subclasses de *PulseRateAlarm* representam as categorias da tabela de classificação da frequência de pulso cujos intervalos de valores se encontram fora da faixa de normalidade, conforme ilustrada pela Figura 4.3. Por exemplo, uma medição de sinal vital de frequência de pulso que se encontra com o valor maior que 100 bpm, esta medição irá disparar um alarme, representado através de uma regra SWRL do Apêndice B , que irá classificar o alarme na subclasse *TachycardiaAlarm* de *PulseRateAlarm*.

### Descrição da subclasse *OxygenSaturationAlarm*

A subclasse de *OxygenSaturationAlarm* representa a categoria da tabela de classificação da saturação do oxigênio da Figura 4.5. A Figura 4.22 ilustra esta subclasse.

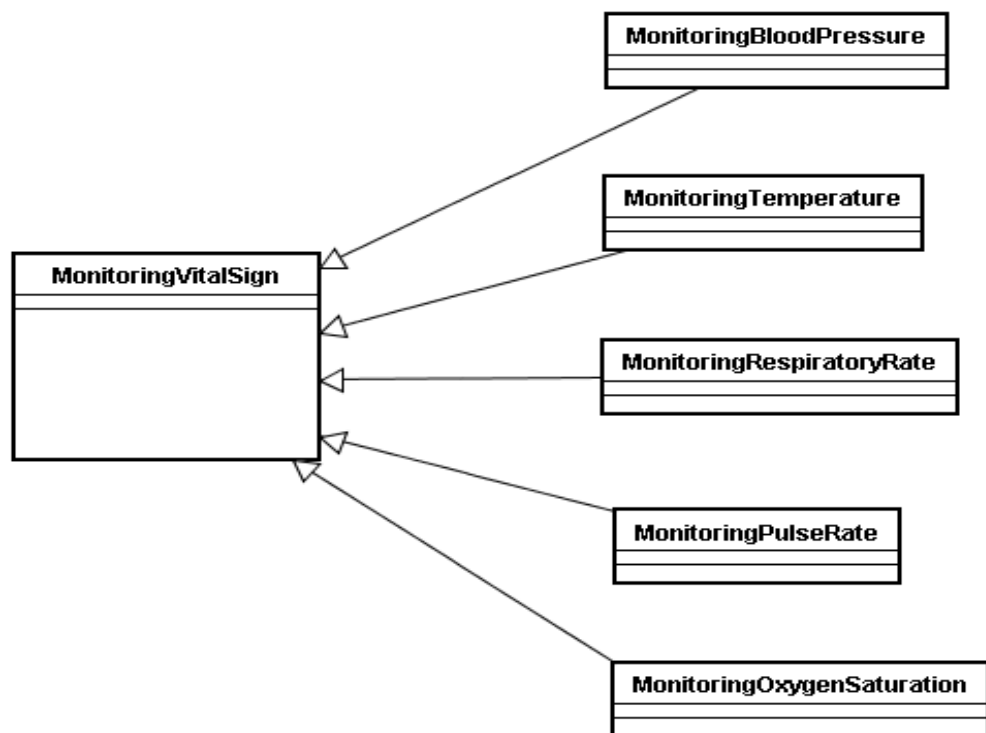


Figura 4.22: Subclasse de *OxygenSaturationAlarm*

A subclasse *HypoxemiaAlarm* representa os indivíduos de alarme classificados com a saturação do oxigênio hipoxemia e que tem os seguintes valores de medição: *saturação do oxigênio menor que 95 %*.

### 4.2.6 Descrição da classe *MonitoringVitalSign*

A classe *MonitoringVitalSign* é uma subclasse da classe *sActivity:Activity*, que é uma ontologia que compõe o modelo SeCoM. A Figura 4.23 ilustra as subclasses da classe *MonitoringVitalSign*. Foram criadas para as subclasses subpropriedades derivadas das relações que *MonitoringVitalSign* mantém com as outras classes, a qual está representada na Figura 4.7.



**Figura 4.23:** *Subclasses de MonitoringVitalSign*

As subclasses *MonitoringBloodPressure*, *MonitoringTemperature*, *MonitoringRespiratoryRate*, *MonitoringPulseRate* e *MonitoringOxygenSaturation* descrevem as atividades espaço-temporais de monitoramento dos sinais vitais pressão arterial, temperatura corpórea, frequência respiratória e de pulso e saturação de oxigênio, respectivamente.

Para todas as subclasses de *MonitoringVitalSign* foram criadas subpropriedades derivadas das relações que *MonitoringVitalSign* mantém com as outras classes, conforme está ilustrada na Figura 4.7. Por exemplo, a subclasse *MonitoringBloodPressure* representará o monitoramento de um sinal vital pressão arterial de um paciente, tendo uma equipe de saúde que assiste a este monitoramento e registrando a data e a localização deste monitoramento. Dessa forma, todos os sinais vitais da classe *vso: 'vital sign'* serão monitorados pelas subclasses de *MonitoringVitalSign*.

Foram construídas regras, por meio da linguagem de regras SWRL, que ao realizar a inferência do modelo ontológico MSVH, pelo raciocinador Pellet, é executada a classificação das medições do monitoramento dos sinais vitais humanos e disparos de alarmes caso ocorram anormalidades nas medições de algum sinal vital. Estas classificações são baseadas nas VI Diretrizes Brasileiras de Hipertensão Arterial, 2010 [13] e na literatura médica [29] e de enfermagem [30], como já mencionadas. As regras construídas estão descritas no Apêndice B deste trabalho.

### 4.3 Validação do modelo ontológico MSVH

SPARQL [21] [40] é uma linguagem de consulta para RDF recomendada pelo W3C. Dessa forma, ela permite que os dados instanciados em ontologias possam ser consultados. A linguagem é semelhante à linguagem SQL usada para consultas em bancos de dados relacionais.

O modelo ontológico MSVH poderia ter sido validado por aplicações no qual o seu domínio de interesse contemplasse o monitoramento de sinais vitais humanos, contudo, é possível realizar a validação do modelo ontológico MSVH através da criação de consultas utilizando a linguagem SPARQL para responder às questões de competência do Apêndice A. Por meio dessas consultas os dados instanciados na ontologia MSVH poderão ser consultados da mesma maneira que profissionais de saúde consultariam os dados armazenados em um ambiente real.

Abaixo são apresentadas consultas SPARQL que respondem algumas questões de competência do Apêndice A.

Questões de competência.

1. Quais são as medições de temperatura de um determinado paciente em um período de tempo específico?

```

1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2 PREFIX var: <http://www.semanticweb.org/ontologies/2013/1/
Ontology1361391792831.owl#>
3 PREFIX actor: <http://linkserver.icmc.usp.br/ckonto/actor#>
4 PREFIX acti: <http://linkserver.icmc.usp.br/ckonto/activity#>
5 PREFIX tEvent: <http://linkserver.icmc.usp.br/ckonto/tEvent#>
6 PREFIX time: <http://linkserver.icmc.usp.br/ckonto/time#>

7 SELECT ?person ?valor ?unidade ?data
8 WHERE {?monitoring acti:hasParticipant ?person .
9 ?person var:hasRole var:patient .
10 ?monitoring var:hasMonitoringTemperature ?axiliary .
11 ?monitoring tEvent:startDateTime ?date .
12 ?temperature var:isMeasurementTemperature ?axiliary .
13 ?temperature var:valueTemperature ?valor .
14 ?temperature var:unitTemperature ?unidade .
15 ?date time:instantCalendarClockDataType ?data .
16 FILTER (?data >= "2013-02-19T10:00:00"^^xsd:dateTime && ?date <=
"2013-02-20T11:00:00"^^xsd:dateTime)

```

}

Linhas	Significado
1 a 6	Representam os prefixos de nomes de URIs que serão utilizados na consulta.
7	Representa os dados que serão retornados pela consulta, neste caso, as variáveis ?person, ?valor, ?unidade e ?data, respectivamente, retornam o paciente, o valor de medição e a unidade da medição e a data e hora que ocorreu a medição.
8	Representa o relacionamento (acti:hasParticipant) entre indivíduos da classe MonitoringTemperature e indivíduos da classe Person.
9	Representa o relacionamento (var:hasRole) entre indivíduos da classe Person e indivíduos da classe RoleOfMember.
10	Representa o relacionamento (var:hasMonitoringTemperature) entre indivíduos da classe MonitoringTemperature e indivíduos da classe 'body temperature'.
11	Representa o relacionamento (tEvent:startDateTime) entre indivíduos da classe MonitoringTemperature e indivíduos da classe InstantEvent.
12	Representa o relacionamento (var:isMeasurementTemperature) entre indivíduos da classe 'body temperature measurement datum' e indivíduos da classe 'body temperature'.
13	Representa o relacionamento entre indivíduos da classe 'body temperature measurement datum' e seu atributo valueTemperature, que representa o valor de medição, que será retornado para a variável ?valor.
14	Representa o relacionamento entre indivíduos da classe 'body temperature measurement datum' e seu atributo unitTemperature, que representa a unidade de medição, que será retornado para a variável ?unidade.
15	Representa o relacionamento entre indivíduos da classe InstantEvent e seu atributo instantCalendarClockDataType, que representa a data e a hora da medição, que será retornado para a variável ?data.
16	Seleciona o período da consulta.

## 2. Quais são as medições de pressão arterial de um determinado paciente em um período de tempo específico?

```

1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2 PREFIX var: <http://www.semanticweb.org/ontologies/2013/1/
Ontology1361391792831.owl#>
3 PREFIX actor: <http://linkserver.icmc.usp.br/ckonto/actor#>
4 PREFIX acti: <http://linkserver.icmc.usp.br/ckonto/activity#>
5 PREFIX tEvent: <http://linkserver.icmc.usp.br/ckonto/tEvent#>
6 PREFIX time: <http://linkserver.icmc.usp.br/ckonto/time#>

7 SELECT ?person ?valorSistolica ?valorDiastolica ?unidade ?data
8 WHERE {?monitoring acti:hasParticipant ?person .
9 ?person var:hasRole var:patient .
10 ?monitoring var:hasMonitoringBloodPressure ?bloodPressure .
11 ?monitoring tEvent:startDateTime ?date .
12 ?pressure var:isMeasurementBloodPressure ?bloodPressure .
13 ?pressure var:valueSystolicBloodPressure ?valorSistolica .
14 ?pressure var:valueDiastolicBloodPressure ?valorDiastolica .
15 ?pressure var:unitBloodPressure ?unidade .
16 ?date time:instantCalendarClockDataType ?data .
17 FILTER (?data >= "2013-02-19T10:00:00"^^xsd:dateTime && ?data <=
"2013-02-20T12:00:00"^^xsd:dateTime)
}

```

Linhas	Significado
7	Representa os dados que serão retornados pela consulta, neste caso, as variáveis ?person, ?valorSistolica, ?valorDiastolica, ?unidade e ?data, respectivamente, retornam o paciente, o valor de medição da pressão sistólica, o valor de medição da pressão diastólica, a unidade da medição e a data e hora que ocorreu a medição.
8	Representa o relacionamento (acti:hasParticipant) entre indivíduos da classe MonitoringBloodPressure e indivíduos da classe Person.
10	Representa o relacionamento (var:hasMonitoringBloodPressure) entre indivíduos da classe MonitoringBloodPressure e indivíduos da classe 'blood pressure'.
11	Representa o relacionamento (tEvent:startDateTime) entre indivíduos da classe MonitoringBloodPressure e indivíduos da classe InstantEvent.

12	Representa o relacionamento (var:isMeasurementBloodPressure) entre indivíduos da classe 'blood pressure measurement datum' e indivíduos da classe 'blood pressure'.
13	Representa o relacionamento entre indivíduos da classe 'blood pressure measurement datum' e seu atributo valueSystolicBloodPressure, que representa o valor de medição da pressão sistólica, que será retornado para a variável ?valorSistolica.
14	Representa o relacionamento entre indivíduos da classe 'blood pressure measurement datum' e seu atributo valueDiastolicBloodPressure, que representa o valor de medição da pressão diastólica, que será retornado para a variável ?valorDiastolica.
15	Representa o relacionamento entre indivíduos da classe 'blood pressure measurement datum' e seu atributo unitBloodPressure, que representa a unidade de medição, que será retornado para a variável ?unidade.

3. Quais são as medições da frequência de pulso de um determinado paciente em um período de tempo específico?

```

1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2 PREFIX var:
<http://www.semanticweb.org/ontologies/2013/1/Ontology1361391792831.owl#>
3 PREFIX actor: <http://linkserver.icmc.usp.br/ckonto/actor#>
4 PREFIX acti: <http://linkserver.icmc.usp.br/ckonto/activity#>
5 PREFIX tEvent: <http://linkserver.icmc.usp.br/ckonto/tEvent#>
6 PREFIX time: <http://linkserver.icmc.usp.br/ckonto/time#>

7 SELECT ?person ?valor ?unidade ?data
8 WHERE {?monitoring acti:hasParticipant var:person .
9 ?person var:hasRole var:patient .
10 ?monitoring var:hasMonitoringPulseRate ?pulseRate .
11 ?monitoring tEvent:startDateTime ?date .
12 ?pulso var:isMeasurementPulseRate ?pulseRate .
13 ?pulso var:valuePulseRate ?valor .
14 ?pulso var:unitPulseRate ?unidade .
15 ?date time:instantCalendarClockDataType ?data .
16 FILTER (?date >= "2013-02-19T10:00:00"^^xsd:dateTime && ?date <=
"2013-02-20T11:00:00"^^xsd:dateTime)
}

```

Linhas	Significado
7	Representa os dados que serão retornados pela consulta, neste caso, as variáveis ?person, ?valor, ?unidade e ?data, respectivamente, retornam o paciente, o valor de medição e a unidade da medição e a data e hora que ocorreu a medição.
8	Representa o relacionamento (acti:hasParticipant) entre indivíduos da classe MonitoringPulseRate e indivíduos da classe Person.
10	Representa o relacionamento (var:hasMonitoringPulseRate) entre indivíduos da classe MonitoringPulseRate e indivíduos da classe 'pulse rate'.
11	Representa o relacionamento (tEvent:startDateTime) entre indivíduos da classe MonitoringPulseRate e indivíduos da classe InstantEvent.
12	Representa o relacionamento (var:isMeasurementPulseRate) entre indivíduos da classe 'pulse rate measurement datum' e indivíduos da classe 'pulse rate'.
13	Representa o relacionamento entre indivíduos da classe 'pulse rate measurement datum' e seu atributo valuePulseRate, que representa o valor de medição, que será retornado para a variável ?valor.
14	Representa o relacionamento entre indivíduos da classe 'pulse rate measurement datum' e seu atributo unitPulseRate, que representa a unidade de medição, que será retornado para a variável ?unidade.

4. Quais são as medições da frequência respiratória de um determinado paciente em um período de tempo específico?

```

1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2 PREFIX var:
<http://www.semanticweb.org/ontologies/2013/1/Ontology1361391792831.owl#>
3 PREFIX actor: <http://linkserver.icmc.usp.br/ckonto/actor#>
4 PREFIX acti: <http://linkserver.icmc.usp.br/ckonto/activity#>
5 PREFIX tEvent: <http://linkserver.icmc.usp.br/ckonto/tEvent#>
6 PREFIX time: <http://linkserver.icmc.usp.br/ckonto/time#>

7 SELECT ?person ?valor ?unidade ?data
8 WHERE {?monitoring acti:hasParticipant var:person .
9 ?person var:hasRole var:patient .
10 ?monitoring var:hasMonitoringRespiratoryRate ?respRate .
11 ?monitoring tEvent:startDateTime ?date .

```

```

12 ?respiratory var:isMeasurementRespiratoryRate ?respRate .
13 ?respiratory var:valueRespiratoryRate ?valor .
14 ?respiratory var:unitRespiratoryRate ?unidade .
15 ?date time:instantCalendarClockDataType ?data .
16 FILTER (?data >= "2013-02-19T10:00:00"^^xsd:dateTime && ?data <=
"2013-02-20T11:00:00"^^xsd:dateTime)
}

```

Linhas	Significado
7	Representa os dados que serão retornados pela consulta, neste caso, as variáveis ?person, ?valor, ?unidade e ?data, respectivamente, retornam o paciente, o valor de medição e a unidade da medição e a data e hora que ocorreu a medição.
8	Representa o relacionamento (acti:hasParticipant) entre indivíduos da classe MonitoringRespiratoryRate e indivíduos da classe Person.
10	Representa o relacionamento (var:hasMonitoringRespiratoryRate) entre indivíduos da classe MonitoringRespiratoryRate e indivíduos da classe 'respiratory rate'.
11	Representa o relacionamento (tEvent:startDateTime) entre indivíduos da classe MonitoringRespiratoryRate e indivíduos da classe InstantEvent.
12	Representa o relacionamento (var:isMeasurementRespiratoryRate) entre indivíduos da classe 'respiratory rate measurement datum' e indivíduos da classe 'respiratory rate'.
13	Representa o relacionamento entre indivíduos da classe 'respiratory rate measurement datum' e seu atributo valueRespiratoryRate, que representa o valor de medição, que será retornado para a variável ?valor.
14	Representa o relacionamento entre indivíduos da classe 'respiratory rate measurement datum' e seu atributo unitRespiratoryRate, que representa a unidade de medição, que será retornado para a variável ?unidade.

5. Quais são as medições de saturação de oxigênio de um determinado paciente em um período de tempo específico?

```

1 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
2 PREFIX var:
<http://www.semanticweb.org/ontologies/2013/1/Ontology1361391792831.owl#>
3 PREFIX actor: <http://linkserver.icmc.usp.br/ckonto/actor#>
4 PREFIX acti: <http://linkserver.icmc.usp.br/ckonto/activity#>

```

```

5 PREFIX tEvent: <http://linkserver.icmc.usp.br/ckonto/tEvent#>
6 PREFIX time: <http://linkserver.icmc.usp.br/ckonto/time#>

7 SELECT ?person ?valor ?unidade ?data
8 WHERE {?monitoring acti:hasParticipant var:person .
9        ?person var:hasRole var:patient .
10       ?monitoring var:hasMonitoringOxygenSaturation ?oxygenSaturation .
11       ?monitoring tEvent:startDateTime ?date .
12       ?oxygen var:isMeasurementOxygenSaturation ?oxygenSaturation .
13       ?oxygen var:valueOxygenSaturation ?valor .
14       ?oxygen var:unitOxygenSaturation ?unidade .
15       ?date time:instantCalendarClockDataType ?data .
16 FILTER (?date >= "2013-02-19T10:00:00"^^xsd:dateTime && ?date <=
"2013-02-20T11:00:00"^^xsd:dateTime)
}

```

Linhas	Significado
7	Representa os dados que serão retornados pela consulta, neste caso, as variáveis ?person, ?valor, ?unidade e ?data, respectivamente, retornam o paciente, o valor de medição e a unidade da medição e a data e hora que ocorreu a medição.
8	Representa o relacionamento (acti:hasParticipant) entre indivíduos da classe MonitoringOxygenSaturation e indivíduos da classe Person.
10	Representa o relacionamento (var:hasMonitoringOxygenSaturation) entre indivíduos da classe MonitoringOxygenSaturation e indivíduos da classe 'oxygen saturation'.
11	Representa o relacionamento (tEvent:startDateTime) entre indivíduos da classe MonitoringOxygenSaturation e indivíduos da classe InstantEvent.
12	Representa o relacionamento (var:isMeasurementOxygenSaturation) entre indivíduos da classe 'oxygen saturation measurement datum' e indivíduos da classe 'oxygen saturation'.
13	Representa o relacionamento entre indivíduos da classe 'oxygen saturation measurement datum' e seu atributo valueOxygenSaturation, que representa o valor de medição, que será retornado para a variável ?valor.

14	Representa o relacionamento entre indivíduos da classe 'oxygen saturation measurement datum' e seu atributo unitOxygenSaturation, que representa a unidade de medição, que será retornado para a variável ?unidade.
----	---

## 4.4 Considerações finais

Este capítulo apresentou a ontologia MSVH que teve como base para o seu desenvolvimento a ontologia Activity que compõem o modelo de informação de contexto SeCoM e a ontologia de sinais vitais VSO.

A ontologia MSVH busca ser uma solução computacional para aplicações do conhecimento para área de saúde, especificamente para o monitoramento de sinais vitais humanos em um ambiente de computação sensível ao contexto.

Dessa forma, a ontologia MSVH modela o conhecimento sobre monitoramento de sinais vitais humanos fornecendo um vocabulário consensual que possa ser compartilhado por aplicações que tenham interesse neste conhecimento para prover informações e serviços relevantes aos seus usuários, tais como, acesso remoto às medições de sinais vitais em tempo real e ao histórico de armazenamento dessas medições, bem como, disponibilizar um mecanismo de alertas para informa a equipe de saúde quando ocorrer possíveis desvios de normalidade dos sinais vitais que estão sendo monitorados, permitindo assim, uma intervenção mais rápida para o restabelecimento de valores aceitáveis dos sinais vitais do paciente.

---

## Trabalhos Relacionados

---

Este capítulo descreve trabalhos que estão inseridos no mesmo contexto de apoiar o procedimento de monitoramento do estado de saúde de pacientes, envolvendo sinais vitais humanos. No final do capítulo, é apresentado um estudo comparativo dos esforços de pesquisa mencionados com o presente trabalho.

### 5.1 Sistema Inteligente de Monitoramento e Envio de Alertas de Pacientes

O Sistema Inteligente de Monitoramento e envio de Alertas de Pacientes (SIMAp), descrito por Leite [23], é um sistema computacional baseado em técnicas de sistemas inteligentes, lógica nebulosa e redes neurais artificiais, aplicadas no monitoramento de pacientes em UTIs.

O objetivo do SIMAp é capturar dados do monitor multiparamétrico para transformar em informações relevantes, através da formalização do conhecimento dos especialistas, bem como dos valores de referência de normalidade dos sinais vitais.

Para a formalização do conhecimento dos especialistas foi utilizada a lógica nebulosa, que realiza a extração (inferência) das informações a respeito do quadro clínico dos pacientes. Através da lógica nebulosa é realizado um pré-diagnóstico das informações capturadas pelo monitor multiparamétrico, que visa auxiliar no atendimento médico.

Caso ocorram desvios de normalidade no monitoramento dos sinais vitais, mensagens de alertas são disparados para a equipe médica, que pode acompanhar o estado de saúde dos pacientes remotamente.

O SIMAp utilizou para a sua validação a base de dados do MIMIC (Multi-parameter Intelligent Monitoring for Intensive Care) para aquisição dos dados fisiológicos (sinais vitais).

Para a realização do pré-diagnóstico o SIMAp correlaciona as medições dos sinais vitais, por exemplo, a pressão arterial média com a saturação de oxigênio.

As inferências obtidas através da lógica nebulosa foram aplicadas para treinamento e validação em uma Rede Neural Artificial (RNA) para realizar a classificação dos resultados encontrados pelo modelo com o intuito de gerar os pré-diagnósticos que auxiliam a equipe médica em suas ações.

Para a classificação dos resultados encontrados pelo SIMAp foram definidas cinco situações que representam as alterações do estado clínico dos pacientes internados na UTI: 1-instabilidade do quadro clínico (todos os sinais alterados), 2 - PAM baixa, 3 - hipoxemia, 4-estável e 5-PAM alta.

## 5.2 Arcabouço Health Support in Aware and Ubiquitous Domestic Environments

O H-SAUDE, sigla para Health Support in Aware and Ubiquitous Domestic Environments, descrito por Copetti et al [12], é um arcabouço que tem o objetivo de realizar o monitoramento inteligente e sensível ao contexto de forma remota a saúde de pessoas. O arcabouço H-SAUDE realiza o monitoramento das variáveis fisiológicas pressão sistólica (PAS), pressão diastólica (PAS) e da frequência cardíaca da pessoa.

O H-SAUDE propõe um monitoramento da saúde de pacientes baseado em lógica nebulosa, que captura os dados de sensores localizados no corpo da pessoa e na residência, e através de regras baseadas no conhecimento de especialistas (médicos) realiza a inferência destas informações, disparando alarmes para um centro de supervisão, caso ocorram desvios da normalidade da pressão arterial do paciente. O modelo utiliza a computação ubíqua para tratar as informações relevantes de contexto para a pessoa, por exemplo, as condições do ambiente como a umidade e temperatura. Os dados obtidos dos sensores são transmitidos por uma rede sem fio para um computador localizado na residência para o processamento do processo de raciocínio sobre os dados.

O H-SAUDE apresenta em sua arquitetura três módulos, a saber: gerenciamento de contexto (Context Management), de raciocínio (Reasoning) e aprendizado (Learning).

O módulo gerenciamento de contexto gerencia os dados dos sensores buscando obter as informações relevantes sobre o contexto do monitoramento. O módulo de raciocínio é o núcleo do arcabouço: este possibilita que uma situação crítica, baseada no pré-processamento dos dados de sensores, possa produzir alertas para uma central. O módulo de aprendizado tem o objetivo de individualizar o parâmetro de referência utilizado para a pessoa. A base de treinamento pode ser formada por exames realizados previamente pela pessoa.

A classificação do estado da pessoa é realizada no módulo de raciocínio e é classificada com os seguintes estados: normal, alerta e emergência.

O estado normal significa que a pessoa está bem, sem necessidade de intervenção dos serviços. O estado alerta significa que o monitoramento deverá ser ajustado para observar melhor o estado da pessoa, por exemplo, diminuir o tempo de monitoramento das variáveis fisiológicas. O estado de emergência significa que alertas deverão ser disparados para serviços de emergência.

## 5.3 TeleCardio

TeleCardio, segundo relatado por Andreão et al [2], é um sistema para monitoramento remoto sensível a contexto de pacientes cardíacos crônicos, que pode utilizar dispositivos móveis para interagir com o sistema. O TeleCardio monitora a atividade cardíaca de pacientes crônicos por meio do eletrocardiograma (ECG), baseando-se na modalidade de telecardiologia em domicílio: há mecanismos que realizam a análise de sinais eletrocardiográficos e o sistema gera alertas automáticos.

A arquitetura de Telecardio é constituída por três camadas conceituais: camada de sensoriamento, camada de suporte e camada de universalização.

Camada de sensoriamento é responsável pela aquisição e processamento dos sinais do ECG. Camada de suporte é a que fornece as ferramentas necessárias para o armazenamento e o processamento das informações de contexto. Camada de universalização é a que fornece a interface de configuração para os diversos domínios da aplicação.

As três camadas são realizadas por duas unidades operacionais: unidade remota e a central de monitoramento. A unidade remota pode ser o domicílio da pessoa ou uma unidade de saúde. A unidade remota é composta por um dispositivo portátil. Este dispositivo portátil é constituído por um Holter (aparelho para medição do ECG), um mecanismo de alarme que é acionado pelo próprio paciente e um dispositivo que realiza a transmissão dos dados por rádio frequência com um computador remoto. É no computador remoto que é realizado o armazenamento e tratamento das informações recebidas, ele possui um mecanismo de análise de sinais do ECG e de geração automática de alarmes.

Através da internet é realizada a comunicação da unidade remota e da central de monitoramento. A central de monitoramento é composta por um servidor de monitoramento e de uma base de dados contextual. O servidor de monitoramento implementa o middleware de suporte e uma aplicação sensível ao contexto que faz gerenciamento das informações e o acompanhamento remoto do estado clínico do paciente. A base de dados contextual armazena as informações de cada paciente utilizado pelo servidor de monitoramento.

O projeto ainda prevê uma plataforma de suporte para o desenvolvimento de aplicações sensíveis ao contexto para atender aos seus diversos cenários. Uma das características da plataforma é o uso de tecnologias da Web Semântica, como ontologias.

O holter envia a cada 30 segundos para o computador remoto as informações dos sinais de ECG pré-processados e eventos associados a situações de risco para o paciente. O módulo de processamento de sinais eletrocardiográficos realiza a análise e dispara alarmes caso ocorra alguma anormalidade. O WrapperHolter é responsável pelo relacionamento entre o modelo de dados específico para o modelo padrão reconhecido pela plataforma de suporte.

Os algoritmos do módulo de processamento de sinal ECG estão organizados em três camadas: camada 0, camada 1 e camada 2.

A camada 0 é a responsável pela realização do processamento mais básico do sinal ECG. Inicialmente, o sinal do ECG é processado para realçar a informação útil através da transformada wavelet Chapéu Mexicano e em seguida é utilizada uma abordagem estatística baseada nos modelos ocultos de Markov (HMM) que associa as amostras do sinal processado ao HMM de cada forma elementar para maximizar a verossimilhança. A camada 1 obtém as informações da camada 0 para realizar a classificação dos batimentos. Esta classificação é obtida por um sistema baseado em regras heurísticas. A classificação implementada é a de batimentos ventriculares prematuros (ESV). Por fim, a camada 2 realiza a identificação de eventos anormais persistentes.

O sistema Telecardio identifica episódios de isquemia cardíaca que normalmente precedem o infarto do miocárdio.

Caso alguma anormalidade seja encontrada na camada 1 e 2, o WrapperHolter é notificado e envia a informação desse evento para a plataforma de suporte.

## 5.4 Projeto Kamer

O projeto Kamer, conforme descrito por Paganelli e Giuli [27], é um projeto regional Italiano no qual estão envolvidas duas Universidades e uma indústria. Um dos objetivos deste projeto é desenhar e desenvolver uma arquitetura de uma plataforma de serviço sensível ao contexto para assistência a pacientes em casa. Um dos trabalhos que faz parte do projeto Kamer está relacionado a um modelo de contexto baseado em ontologia para monitoramento da saúde em domicílio e alerta em redes de cuidados de pacientes crônicos.

Este modelo sensível ao contexto baseado em ontologia estende o modelo de propósito geral do projeto Kamer para representar um domínio relacionado ao cuidado da saúde em domicílio.

Este modelo sensível ao contexto é constituído das seguintes ontologias: ontologia de domínio pessoal do paciente, ontologia de domínio domiciliar, ontologia de gerenciamento de alarme e a ontologia de contexto social.

A ontologia de domínio pessoal do paciente representa os dados de contexto relacionados às variáveis fisiológicas (valores de parâmetros biomédicos, por exemplo, temperatura corpórea), a localização e a atividade do paciente. Estas informações são úteis para o sistema inferir o estado clínico do paciente e detectar possíveis situações de alarmes. Os alarmes são disparados nesta ontologia quando ocorre desvio do valor de medição de alguma variável fisiológica relacionada ao intervalo de parâmetros correspondente à variável fisiológica. Os intervalos de parâmetros são especificados em termos de valores mínimo e máximo e relacionados ao nível do alarme. São especificados quatro níveis de alarme (very low, low, medium e high).

A ontologia de domínio domiciliar representa os dados de contexto relacionados aos parâmetros ambientais, tal como, a temperatura e a umidade. Estes parâmetros são importantes para garantir um ambiente saudável e detectar possíveis situações de alarmes. Os alarmes são disparados nesta ontologia quando ocorre algum fato anormal que está relacionado ao ambiente.

A ontologia de gerenciamento de alarme representa quais os membros da rede de cuidados que podem ser acionados através de envio de SMS e/ou e-mails, conforme a política de notificação que está relacionada ao nível do alarme, para realizar intervenções nos casos em que ocorram situações críticas.

A ontologia de contexto social representa os recursos da rede de atendimento que vêm de diferentes organizações (equipes de saúde, membros da comunidade social, dentre outros).

O modelo adota dois tipos de raciocínio do contexto: uma abordagem baseada em regras de primeira ordem e outra baseada em ontologia. Os dois usam um motor de raciocínio baseado em regras. Dessa forma, é possível realizar inferências sobre as ontologias para determinar se alarmes devem ser disparados quando as medições ficarem fora dos intervalos de parâmetro e inferir o nível do alarme.

Um middleware foi desenvolvido para facilitar a implementação dos serviços de assistência a pacientes crônicos. O middleware do modelo foi baseado no middleware do Projeto Kamer.

## 5.5 Análise dos trabalhos

Dentre os trabalhos apresentados, o que possui as mesmas variáveis fisiológicas (sinais vitais) que o modelo ontológico proposto é o sistema SIMAp que representa os sinais vitais de pressão arterial, temperatura corpórea, frequência cardíaca, frequência de pulso e saturação de oxigênio do sangue. O arcabouço H-SAUDE monitora apenas a pressão arterial e a temperatura corpórea. O sistema Telecardio realiza o monitoramento da atividade cardíaca através do eletrocardiograma (ECG). Por fim, a ontologia para

monitoramento da saúde em casa do Projeto Kamer representa a frequência cardíaca e a temperatura corpórea.

A abordagem utilizada pelo sistema SIMAp e a do arcabouço H-SAUDE para inferência dos resultados das medições dos sinais vitais é da forma estatística, representadas através de regras da lógica nebulosa. O sistema Telecardio também possui uma abordagem estatística para a inferência dos resultados das medições dos sinais vitais, porém, é baseada nos modelos ocultos de Markov (HMM) e em regras heurísticas.

Já as abordagens utilizadas pelo modelo ontológico MSVH e pela ontologia do projeto Kamer são determinísticas. O modelo ontológico proposto utiliza regras da lógica de predicados da linguagem SWRL para a inferência dos resultados das medições dos sinais vitais. A ontologia do Projeto Kamer utiliza regras baseadas na lógica de primeira ordem para realizar as inferências dos resultados das medições.

As classificações do estado clínico do paciente servem como parâmetros de disparo de alarmes. As classificações definidas pelo sistema SIMAp, pelo arcabouço H-SAUDE e pela ontologia do Projeto Kamer são apresentadas de forma mais genérica, utilizando valores alto, médio e baixo para os sinais envolvidos na sua classificação de alarmes.

O modelo ontológico desenvolvido detalha a classificação do estado clínico do paciente para cada sinal vital monitorado, no qual são representadas as classificações para a pressão arterial, conforme a Figura 4.6 da VI Diretrizes Brasileiras de Hipertensão Arterial de 2010 descrita pela Sociedade Brasileira de Cardiologia et al [13], para a temperatura corpórea, conforme a Figura 4.2 descrito por Porto e Porto [29] e para a frequência de pulso, frequência respiratória e saturação de oxigênio, conforme as figuras 4.3, 4.4 e 4.5 descritas por Potter e Perry [30].

---

## Conclusões e Trabalhos Futuros

---

Este trabalho propôs um modelo ontológico baseado no dimensionamento de informações de contexto e no conhecimento relativo ao monitoramento de sinais vitais de pacientes em UTIs, com o objetivo de apoiar o desenvolvimento de sistemas para a área de Saúde.

O modelo ontológico denominado Monitoramento de Sinais Vitais Humanos (MSVH) permite inferir informações sobre as medições de sinais vitais humanos, disparando alarmes para alertar a equipe de saúde quando ocorrerem desvios de normalidades dos valores de referência e/ou individualizados dos sinais vitais do paciente. O modelo ontológico MSVH também provê o registro das medições dos sinais vitais de forma automática e não mais de forma manual, como relatado no Capítulo 4. Esses registros poderão ser utilizados pela equipe médica para o acompanhamento das alterações dos valores de medições dos sinais vitais.

A metodologia utilizada para o desenvolvimento da ontologia MSVH seguiu os passos descritos pela metodologia *Ontology Development 101* proposta por Noy e McGuinness [25].

Vale ressaltar o potencial do uso da ontologia MSVH que, além de formalizar o conhecimento modelado, pode facilitar o desenvolvimento de aplicações, ao permitir a separação entre a lógica de programação e o conhecimento que a aplicação precisa tratar.

### 6.1 Contribuições

As principais contribuições deste trabalho são as seguintes:

1. Um modelo ontológico para *Monitoramento de Sinais Vitais Humanos* (MSVH) baseado nas dimensões de informações de contexto. O uso de ontologias permite a formalização de um domínio de interesse, descrevendo a semântica explícita de seus conceitos, o seu compartilhamento e reúso por outros membros que tenham interesse nesse domínio.

2. Extensão da ontologia VSO com adição do sinal vital de saturação de oxigênio e da sua medição, sinal cada vez mais utilizado no dia-a-dia dos profissionais de saúde.
3. Extensão do modelo SeCoM, através do desenvolvimento do modelo ontológico de monitoramento de sinais vitais humanos.
4. Um conjunto de questões de competência para o desenvolvimento do modelo MSVH, como resultado de reuniões, entrevistas, questionários e leitura de bibliografia da área da Saúde.

## 6.2 Limitações do modelo ontológico MSVH

1. A ontologia MSVH tem como parâmetro de classificação dos sinais vitais e de disparo de alarmes os valores de referência populacional das tabelas de classificação dos sinais vitais para pessoas adultas. Os valores de referência para os sinais vitais sofrem alterações conforme a idade da pessoa, por exemplo, lactantes têm um valor de referência de normalidade para o sinal vital de frequência cardíaca de 120 a 160 bpm, enquanto o valor normal de referência para a classificação populacional de adultos está entre 60 a 100 bpm.
2. Os sinais vitais sofrem alterações conforme a atividade realizada por uma pessoa, por exemplo, fazer exercícios físicos aumenta a frequência respiratória e de pulso. O modelo MSVH foi construído para monitorar os sinais vitais de pacientes que se encontram em repouso, o que acontece nas UTIs, onde os pacientes se encontram acomodados em leitos hospitalares.
3. A validação do modelo MSVH, através de consultas na linguagem SPARQL, não foi realizada para todas as questões de competência. Isto se deve ao fato de que na validação realizada com a ferramenta Protégé, os alarmes e as classificações de sinais vitais, que resultam de dados inferidos, não puderam ser reproduzidos por uma limitação dessa ferramenta.

## 6.3 Trabalhos Futuros

1. Validar o modelo ontológico MSVH com o desenvolvimento de uma aplicação que possa utilizar o conhecimento modelado da ontologia MSVH para avaliar o seu raciocínio em relação ao monitoramento de sinais vitais humanos. Dessa forma, a aplicação deverá permitir ao médico conseguir, em um ambiente controlado como uma UTI ou home care, programar o monitoramento de cada sinal vital de um paciente instrumentado com uma rede de sensores, registrar cada sinal monitorado e receber alertas, conforme a ocasião, ou que os alertas possam ser configurados para invocar serviços de emergência em situações extremas.

2. Transferir a complexidade inerente à atividade de monitoramento de sinais vitais humanos da ontologia MSVH para um ambiente de home care. Adicionando novos conceitos que permitam representar possíveis atividades que estariam sendo realizadas pelo paciente quando seus sinais vitais fossem alterados, dessa forma, poderiam ser evitados falsos alarmes, por exemplo, um idoso que fosse subir em uma escada para trocar uma lâmpada do teto, poderia ter seu sinal de frequência de pulso alterado e dispararia um falso alarme para um serviço de emergência ou para o médico que o assiste.
3. Adicionar à ontologia MSVH novas classificações de valores de referência para contemplar os pacientes com idade diferente da classificação de adultos. Assim, permite que o modelo possa ser utilizado para apoiar a equipe médica na assistência à saúde de crianças e adolescentes, podendo ser utilizado no monitoramento dos sinais vitais dos pacientes que se encontram na UTI pediátrica.
4. Adicionar outros sinais vitais a ontologia MSVH, tal como, a glicemia que mede a quantidade de açúcar no sangue para identificar o diabetes e qual o nível de glicose no sangue.

---

## Referências Bibliográficas

---

- [1] ABOWD, G. D.; MYNATT, E. D. **Charting past, present, and future research in ubiquitous computing**. *ACM Trans. Comput.-Hum. Interact.*, 7(1):29–58, Mar. 2000.
- [2] ANDREÃO, R. V.; FILHO, J. G. P.; CALVI, C. Z. **Telecardio – telecardiologia a serviço de pacientes hospitalizados em domicílio**. In: *X Brazilian Conference in Health Informatics*, CBIS '06, Florianópolis, Brazil, 2006. CBIS.
- [3] BAO, J.; KENDALL, E.; MCGUINNESS, D.; PATEL-SCHNEIDER, P. **OWL 2 Web Ontology Language Quick Reference Guide (Second Edition)**. Disponível em <http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/>. Visitado em 26/06/2012.
- [4] BASTIANI, E.; SOARES, K.; LIBRELOTTO, G. R. **Uma abordagem para monitoramento de pacientes com alzheimer em ambientes homecare pervasivos**. In: *Workshop de Informática Médica, WIM '12*, p. 1–10, Curitiba, Paraná, Brasil, 2012.
- [5] BASTOS, A. B.; SENE JR, I. G.; MARQUES, G. S.; SENA, M. V. O.; BULCÃO-NETO, R. F. **Uma abordagem ontológica para representação de conhecimento de monitoramento de sinais vitais humanos**. In: *XIII Workshop de Teses e Dissertações como evento integrante do XIX Simpósio Brasileiro de Sistemas Multimídia e Web*, p. 1–4, Salvador, Bahia, Brasil, 2013.
- [6] BECHHOFFER, S.; VAN HARMELEN, F.; HENDLER, J.; HORROCKS, I.; MCGUINNESS, D. L.; PATEL-SCHNEIDER, P. F.; STEIN, L. A. **OWL Web Ontology Language Reference**. W3C Recommendation 10 February 2004. Disponível em <http://www.w3.org/TR/owl-ref/>. Visitado em 14/03/2013.
- [7] BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. **The semantic web**. *Scientific American*, 284(5):34–43, May 2001.
- [8] BORST, W. N. **Construction of engineering ontologies for knowledge sharing and reuse**. Centre for Telematics and Information Technology, Enschede, 1997.

- [9] BULCÃO-NETO, R. F. **Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto.** Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação – ICMC-USP, 2006.
- [10] BULCÃO-NETO, R. F.; MACEDO, A. A.; CAMACHO-GUERRERO, J. A.; PIMENTEL, M. G. C. **Configurable semantic services leveraging applications context-aware.** In: *Anais do Simpósio Brasileiro de Sistemas Multimídia e Web, WebMedia '05*, p. 1–9, Poços de Caldas-MG, Brasil, 2005. ACM Press.
- [11] BULCÃO-NETO, R. F.; PIMENTEL, M. G. C. **Toward a domain-independent semantic model for context-aware computing.** In: *Proceedings of the 3rd Latin American Web Congress, LA-Web '05*, p. 61–70, Buenos Aires, Argentina, 2005. IEEE CS Press.
- [12] COPETTI, A.; LEITE, J. C. B.; LOQUES, O.; BARBOSA, T. P. C.; NOBREGA, A. C. L. **Monitoramento inteligente e sensível ao contexto na assistência domiciliar telemonitorada.** In: *Anais do XXVIII Congresso da SBC - SEMISH*, p. 166–180, Belém do Pará, PA, 2008.
- [13] DE CARDIOLOGIA, S. B.; DE HIPERTENSÃO, S. B.; DE NEFROLOGIA, S. B. **VI Diretrizes Brasileiras de Hipertensão. Arquivos Brasileiros de Cardiologia 2010.** 5(1 supl. 1), 1-51., 2010.
- [14] DEY, A. K. **Understanding and using context.** *Personal Ubiquitous Comput.*, 5(1):4–7, Jan. 2001.
- [15] FENSEL, D. **Ontologies: a silver bullet for knowledge management and electronic commerce.** Springer, Berlin ; New York, 2001.
- [16] GENNARI, J. H.; MUSEN, M. A.; FERGERSON, R. W.; GROSSO, W. E.; CRUBÉZY, M.; ERIKSSON, H.; NOY, N. F.; TU, S. W. **The evolution of protege: an environment for knowledge-based systems development.** *International Journal of Human-Computer Studies*, 58(1):89–123, Jan. 2003.
- [17] GOLDFAIN, A.; SMITH, B.; ARABANDI, S.; BROCHHAUSEN, M.; HOGAN, W. R. **Vital sign ontology.** In: *Proceedings of the Workshop on Bio-Ontologies, ISMB Intelligent Systems for Molecular Biology*, p. 71–74, Vienna, 2005. ACM Press.
- [18] GRUBER, T. **Ontology.** Encyclopedia of Database Systems: pp. 1963-1965, 2009.
- [19] GRUBER, T. R. **A translation approach to portable ontology specifications.** *Knowledge Acquisition*, 5(2):199–220, June 1993.

- [20] GUARINO, N. **Formal ontology and information systems**. In: *Proceedings of the 1st International Conference*, p. 3–15, Amsterdam, The Netherlands, The Netherlands, 1998. IOS Press.
- [21] HARRIS, S.; SEABORNE, A. **SPARQL 1.1 Query Language**. Disponível em <http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>. Visitado em 18/07/2013.
- [22] HORROCKS, I.; PATEL-SCHNEIDER, P. F.; BOLEY, H.; TABET, S.; GROSOFF, B.; DEAN, M. **SWRL: A Semantic Web rule language combining OWL and RuleML**. W3C member submission, 2004. Disponível em <http://www.w3.org/Submission/SWRL/>. Visitado em 20/06/2013.
- [23] LEITE, C. R. M. **Arquitetura Inteligente Fuzzy para Monitoramento de Sinais Vitais de Pacientes: Um Estudo de Caso em UTI**. Tese de Doutorado. Programa de Pós-Graduação em Engenharia Elétrica e Computação – UFRGN - Centro de Tecnologia, 2011.
- [24] MARTINS, E. E.; SALDIAS, C. P. **Protege**. Programa de Pós-Graduação em Engenharia de Automação e Sistemas e Engenharia Mecânica. UFSC. Florianópolis, SC., 2008.
- [25] NOY, N. F.; MCGUINNESS, D. L. **Ontology development 101: A guide to creating your first ontology**. Technical report, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, 2001.
- [26] OGMS. **Ontology for General Medical Science**. W3C member submission. Disponível em <http://code.google.com/p/ogms/>. Visitado em 06/07/2013.
- [27] PAGANELLI, F.; GIULI, D. **An ontology-based context model for home health monitoring and alerting in chronic patient care networks**. In: *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 02, AINAW '07*, p. 838–845, Washington, DC, USA, 2007. IEEE Computer Society.
- [28] PELLET. **Project Pellet**. Disponível em <http://code.google.com/p/pellet/>. Visitado em 14/07/2013.
- [29] PORTO, C. C.; PORTO, A. L. **Semiologia médica**. Guanabara Koogan, Rio de Janeiro, Brasil, 2009.
- [30] POTTER, P. A.; PERRY, A. G.; MACIEIRA, C. E. S. **Fundamentos de enfermagem**. Elsevier, Rio de Janeiro, 2006.

- [31] PROTEGE. **Protege**. Stanford University School of Medicine. Disponível em <http://protege.stanford.edu/>. Visitado em 12/07/2013.
- [32] RODRIGUES-FILHO, R. V.; BULCÃO-NETO, R. F.; SILVESTRE, B. O.; OLIVEIRA, L. L. G.; OLIVEIRA, R. O.; SENE-JUNIOR, I. G. **An evaluation method of research on wearable wireless body area network in healthcare**. *IJCSIT International Journal of Computer Science e Information Technology*, 5(1):65–78, 2013.
- [33] SCHEUERMANN, R. H.; CEUSTERS, W.; SMITH, B. **Toward an ontological treatment of disease and diagnosis**. In: *Proceedings of the 2009 AMIA Summit on Translational Bioinformatics*, p. 116–120, San Francisco, CA, EUA, 2009.
- [34] SIG-FMA. **Foundational Model of Anatomy**. Disponível em <http://sig.biostr.washington.edu/projects/fm/AboutFM.html>. Visitado em 07/07/2013.
- [35] SIRIN, E.; PARSIA, B.; GRAU, B. C.; KALYANPUR, A.; KATZ, Y. **Pellet: A practical owl-dl reasoner**. *Web Semant.*, 5(2):51–53, June 2007.
- [36] SMITH, M. K.; WELTY, C.; MCGUINNESS, D. L. **OWL Web Ontology Language Guide**. W3C Recommendation 10 February 2004. Disponível em <http://www.w3.org/TR/owl-guide/>. Visitado em 14/03/2013.
- [37] SOUSA, J. P. P.; CARRAPATOSO, E.; FONSECA, B.; PIMENTEL, M. G. C.; BULCÃO-NETO, R. F. **Composition of contextaware mobile services using a semantic context model**. *International Journal on Advances in Software*, Volume 2, pp. 275–287, 2009.
- [38] SPEAR, A. D. **Ontology for the Twenty First Century: An Introduction with Recommendations**. IFOMIS, Saarbrücken, Germany, 2006.
- [39] TRUONG, K. N.; ABOWD, G. D.; MYNATT, E. D. **Who, what, when, where, how: Design issues of capture and access applications**. 2201:209–224, 2001.
- [40] W3C. **SPARQL 1.1 Overview**. Disponível em <http://www.w3.org/TR/sparql11-overview/>. Visitado em 18/07/2013.
- [41] WELTY, C.; GUARINO, N. **Supporting ontological analysis of taxonomic relationships**. *Data Knowledge Engineering*, 39(1):51–74, Oct. 2001.

---

## Questões de Competência

---

A metodologia *Ontology Development 101* de Noy e McGuinness [25] propõe na atividade de delimitação do domínio e do escopo da ontologia a elaboração de uma lista de questões, denominadas questões de competência, que a ontologia a ser desenvolvida deve responder.

Como resultado das atividades de entrevistas, aplicação de questionários, visitas a UTI do HC-UFG e pesquisas à literatura médica, foram produzidas as seguintes questões de competência. Para cada uma delas, são apresentadas as classes, propriedades e relações que permitem respondê-la.

### **1. Quais são as medições de temperatura de um determinado paciente em um período de tempo específico?**

Classes:

MonitoringTemperature, 'respiratory rate measurement datum', 'body temperature', Person e InstantEvent

Relações:

MonitoringTemperature (hasMonitoringTemperature) 'body temperature'

MonitoringTemperature (hasParticipant) Person

MonitoringTemperature (startDateTime) InstantEvent

'body temperature measurement datum' (isMeasurementTemperature) 'body temperature'

Atributos:

'respiratory rate measurement datum'[valueTemperature(float), unitTemperature(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

### **2. Quais são as medições de pressão arterial de um determinado paciente em um período de tempo específico?**

Classes:

MonitoringBloodPressure, 'blood pressure measurement datum', 'blood pressure', Per-

son e InstantEvent

Relações:

MonitoringBloodPressure (hasMonitoringBloodPressure) 'blood pressure'

MonitoringBloodPressure (hasParticipant) Person

MonitoringBloodPressure (startDateTime) InstantEvent

'blood pressure measurement datum' (isMeasurementBloodPressure) 'blood pressure'

Atributos:

'blood pressure measurement datum'[valueDiastolicBloodPressure(nonNegativeInteger), valueSystolicBloodPressure, unitBloodPressure(nonNegativeInteger), unitBloodPressure(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

### **3. Quais são as medições da frequência de pulso de um determinado paciente em um período de tempo específico?**

Classes:

MonitoringPulseRate, 'pulse rate measurement datum', 'pulse rate', Person e InstantEvent

Relações:

MonitoringPulseRate (hasMonitoringPulseRate) 'pulse rate'

MonitoringPulseRate (hasParticipant) Person

MonitoringPulseRate (startDateTime) InstantEvent

'pulse rate measurement datum' (isMeasurementPulseRate) 'pulse rate'

Atributos:

'pulse rate measurement datum'[valuePulseRate(nonNegativeInteger), unitPulseRate(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

### **4. Quais são as medições da frequência respiratória de um determinado paciente em um período de tempo específico?**

Classes:

MonitoringRespiratoryRate, 'respiratory rate measurement datum', 'respiratory rate', Person e InstantEvent

Relações:

MonitoringRespiratoryRate (hasMonitoringRespiratoryRate) 'respiratory rate'

MonitoringRespiratoryRate (hasParticipant) Person

MonitoringRespiratoryRate (startDateTime) InstantEvent

'respiratory rate measurement datum' (isMeasurementRespiratoryRate) 'respiratory rate'  
Atributos:

'respiratory rate measurement datum'[valueRespiratoryRate(nonNegativeInteger), unitRespiratoryRate(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

### **5. Quais são as medições de saturação de oxigênio de um determinado paciente em um período de tempo específico?**

Classes:

MonitoringOxygenSaturation, 'oxygen saturation measurement datum', 'oxygen saturation', Person e InstantEvent

Relações:

MonitoringOxygenSaturation (hasMonitoringOxygenSaturation) 'oxygen saturation'

MonitoringOxygenSaturation (hasParticipant) Person

MonitoringOxygenSaturation (startDateTime) InstantEvent

'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen saturation'

Atributos:

'oxygen saturation measurement datum'[valueOxygenSaturation(nonNegativeInteger), unitOxygenSaturation(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

### **6. Como saber se um certo paciente teve sua medição de temperatura fora da normalidade em um determinado período de tempo específico?**

Classes:

MonitoringTemperature, 'body temperature measurement datum', 'body temperature', Person, InstantEvent e BodyTemperatureAlarm

Relações:

MonitoringTemperature (hasMonitoringTemperature) 'body temperature'

MonitoringTemperature (hasParticipant) Person

MonitoringTemperature (startDateTime) InstantEvent

'body temperature measurement datum' (isMeasurementTemperature) 'body temperature'

'body temperature measurement datum' (triggersAlarm) BodyTemperatureAlarm

Atributos:

'body temperature measurement datum'[valueTemperature(nonNegativeInteger), unit-

Temperature(string)]  
 InstantEvent[instantCalendarClockDataType(dateTime)]  
 Person[hasName(string)]

A classe BodyTemperatureAlarm possui subclasses que indica em qual classificação de alarme de temperatura a medição foi disparada.

### **7. Como saber se um certo paciente teve sua medição de pressão arterial fora da normalidade em um determinado período de tempo específico?**

Classes:

MonitoringBloodPressure, 'blood pressure measurement datum', 'blood pressure', Person, InstantEvent e BloodPressureAlarm

Relações:

MonitoringBloodPressure (hasMonitoringBloodPressure) 'blood pressure'

MonitoringBloodPressure (hasParticipant) Person

MonitoringBloodPressure (startDateTime) InstantEvent

'blood pressure measurement datum' (isMeasurementBloodPressure) 'blood pressure'

'blood pressure measurement datum' (triggersAlarm) BloodPressureAlarm

Atributos:

'blood pressure measurement datum'[valueDiastolicBloodPressure(nonNegativeInteger), valueSystolicBloodPressure, unitBloodPressure(nonNegativeInteger), unitBloodPressure(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe BloodPressureAlarm possui subclasses que indica em qual classificação de alarme de pressão arterial a medição foi disparada.

### **8. Como saber se um certo paciente teve sua medição de frequência de pulso fora da normalidade em um determinado período de tempo específico?**

Classes:

MonitoringPulseRate, 'pulse rate measurement datum', 'pulse rate', Person, InstantEvent e PulseRateAlarm

Relações:

MonitoringPulseRate (hasMonitoringPulseRate) 'pulse rate'

MonitoringPulseRate (hasParticipant) Person

MonitoringPulseRate (startDateTime) InstantEvent

'pulse rate measurement datum' (isMeasurementPulseRate) 'pulse rate'

'pulse rate measurement datum' (triggersAlarm) PulseRateAlarm

Atributos:

'pulse rate measurement datum'[valuePulseRate(nonNegativeInteger), unitPulseRate(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe PulseRateAlarm possui subclasses que indica em qual classificação de alarme de frequência de pulso a medição foi disparada.

### **9. Como saber se um certo paciente teve sua medição de frequência respiratória fora da normalidade em um determinado período de tempo específico?**

Classes:

MonitoringRespiratoryRate, 'respiratory rate measurement datum', 'respiratory rate', Person, InstantEvent e RespiratoryRateAlarm

Relações:

MonitoringRespiratoryRate (hasMonitoringRespiratoryRate) 'respiratory rate'

MonitoringRespiratoryRate (hasParticipant) Person

MonitoringRespiratoryRate (startDateTime) InstantEvent

'respiratory rate measurement datum' (isMeasurementRespiratoryRate) 'respiratory rate'

'respiratory rate measurement datum' (triggersAlarm) RespiratoryRateAlarm

Atributos:

'respiratory rate measurement datum'[valueRespiratoryRate(nonNegativeInteger), unitRespiratoryRate(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe RespiratoryRateAlarm possui subclasses que indica em qual classificação de alarme de frequência respiratória a medição foi disparada.

### **10. Como saber se um certo paciente teve sua medição de saturação de oxigênio fora da normalidade em um determinado período de tempo específico?**

Classes:

MonitoringOxygenSaturation, 'oxygen saturation measurement datum', 'oxygen saturation', Person, InstantEvent e OxygenSaturationAlarm

Relações:

MonitoringOxygenSaturation (hasMonitoringOxygenSaturation) 'oxygen saturation'

MonitoringOxygenSaturation (hasParticipant) Person

MonitoringOxygenSaturation (startDateTime) InstantEvent

'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen saturation'

'oxygen saturation measurement datum' (triggersAlarm) OxygenSaturationAlarm

Atributos:

'oxygen saturation measurement datum'[valueOxygenSaturation(nonNegativeInteger), unitOxygenSaturation(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe OxygenSaturationAlarm possui subclasses que indica em qual classificação de alarme de saturação de oxigênio a medição foi disparada.

**11. É possível identificar em qual faixa da tabela de classificação da temperatura corpórea, descrita na Figura 4.2, se encontra uma medição de monitoramento do sinal vital de temperatura corpórea fora do intervalo de normalidade?**

Sim. Uma vez que temos os seguintes elementos na ontologia.

Classes:

MonitoringTemperature, 'body temperature measurement datum', 'body temperature', Person, InstantEvent e BodyTemperatureAlarm

Relações:

MonitoringTemperature (hasMonitoringRespiratoryRate) 'body temperature'

MonitoringTemperature (hasParticipant) Person

MonitoringTemperature (startDateTime) InstantEvent

'body temperature measurement datum' (isMeasurementTemperature) 'body temperature'

'body temperature measurement datum' (triggersAlarm) BodyTemperatureAlarm

Atributos:

'body temperature measurement datum'[valueTemperature(nonNegativeInteger), unitTemperature(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe BodyTemperatureAlarm possui subclasses que indica em qual classificação de alarme de temperatura a medição foi disparada.

**12. É possível identificar em qual faixa da tabela de classificação da pressão arterial, descrita na Figura 4.6, se encontra uma medição de monitoramento do sinal vital de**

**pressão arterial fora do intervalo de normalidade?**

Sim. Uma vez que temos os seguintes elementos na ontologia.

Classes:

MonitoringBloodPressure, 'blood pressure measurement datum', 'blood pressure', Person, InstantEvent e BloodPressureAlarm

Relações:

MonitoringBloodPressure (hasMonitoringBloodPressure) 'blood pressure'

MonitoringBloodPressure (hasParticipant) Person

MonitoringBloodPressure (startDateTime) InstantEvent

'blood pressure measurement datum' (isMeasurementBloodPressure) 'blood pressure'

'blood pressure measurement datum' (triggersAlarm) BloodPressureAlarm

Atributos:

'blood pressure measurement datum'[valueDiastolicBloodPressure(nonNegativeInteger), valueSystolicBloodPressure, unitBloodPressure(nonNegativeInteger), unitBloodPressure(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe BloodPressureAlarm possui subclasses que indica em qual classificação de alarme de pressão arterial a medição foi disparada.

**13. É possível identificar em qual faixa da tabela de classificação da frequência de pulso, descrita na Figura 4.3, se encontra uma medição de monitoramento do sinal vital de frequência de pulso fora do intervalo de normalidade?**

Sim. Uma vez que temos os seguintes elementos na ontologia.

Classes:

MonitoringPulseRate, 'pulse rate measurement datum', 'pulse rate', Person, InstantEvent e PulseRateAlarm

Relações:

MonitoringPulseRate (hasMonitoringPulseRate) 'pulse rate'

MonitoringPulseRate (hasParticipant) Person

MonitoringPulseRate (startDateTime) InstantEvent

'pulse rate measurement datum' (isMeasurementPulseRate) 'pulse rate' 'pulse rate measurement datum' (triggersAlarm) PulseRateAlarm

Atributos:

'pulse rate measurement datum'[valuePulseRate(nonNegativeInteger), unitPulseRate(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe PulseRateAlarm possui subclasses que indica em qual classificação de alarme de frequência de pulso a medição foi disparada.

**14. É possível identificar em qual faixa da tabela de classificação da frequência respiratória, descrita na Figura 4.4, se encontra uma medição de monitoramento do sinal vital de frequência respiratória fora do intervalo de normalidade?**

Sim. Uma vez que temos os seguintes elementos na ontologia.

Classes:

MonitoringRespiratoryRate, 'respiratory rate measurement datum', 'respiratory rate', Person, InstantEvent e RespiratoryRateAlarm

Relações:

MonitoringRespiratoryRate (hasMonitoringRespiratoryRate) 'respiratory rate'

MonitoringRespiratoryRate (hasParticipant) Person

MonitoringRespiratoryRate (startDateTime) InstantEvent

'respiratory rate measurement datum' (isMeasurementRespiratoryRate) 'respiratory rate'

'respiratory rate measurement datum' (triggersAlarm) RespiratoryRateAlarm

Atributos:

'respiratory rate measurement datum'[valueRespiratoryRate(nonNegativeInteger), unitRespiratoryRate(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe RespiratoryRateAlarm possui subclasses que indica em qual classificação de alarme de frequência respiratória a medição foi disparada.

**15. É possível identificar em qual faixa da tabela de classificação da saturação de oxigênio, descrita na Figura 4.5, se encontra uma medição de monitoramento do sinal vital de saturação de oxigênio fora do intervalo de normalidade?**

Sim. Uma vez que temos os seguintes elementos na ontologia.

Classes:

MonitoringOxygenSaturation, 'oxygen saturation measurement datum', 'oxygen saturation', Person, InstantEvent e OxygenSaturationAlarm

Relações:

MonitoringOxygenSaturation (hasMonitoringOxygenSaturation) 'oxygen saturation'

MonitoringOxygenSaturation (hasParticipant) Person

MonitoringOxygenSaturation (startDateTime) InstantEvent

'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen saturation'

'oxygen saturation measurement datum' (triggersAlarm) OxygenSaturationAlarm

Atributos:

'oxygen saturation measurement datum'[valueOxygenSaturation(nonNegativeInteger), unitOxygenSaturation(string)]

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

A classe OxygenSaturationAlarm possui subclasses que indica em qual classificação de alarme de saturação de oxigênio a medição foi disparada.

## **16. Como saber a data do último monitoramento do sinal vital temperatura de um determinado paciente?**

Classes:

MonitoringTemperature, 'body temperature', 'body temperature measurement datum' Person e InstantEvent

Relações:

MonitoringTemperature (hasParticipant) Person

MonitoringTemperature (startDateTime) InstantEvent

'body temperature measurement datum' (isMeasurementTemperature) 'body temperature'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'body temperature measurement datum'[valueTemperature(nonNegativeInteger), unitTemperature(string)]

## **17. Como saber a data do último monitoramento do sinal vital pressão arterial de um determinado paciente?**

Classes:

MonitoringBloodPressure, 'blood pressure', 'blood pressure measurement datum', Person e InstantEvent

Relações:

MonitoringBloodPressure (hasParticipant) Person

MonitoringBloodPressure (startDateTime) InstantEvent

'blood pressure measurement datum' (isMeasurementBloodPressure) 'blood pressure'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'blood pressure measurement datum'[valueDiastolicBloodPressure(nonNegativeInteger), valueSystolicBloodPressure, unitBloodPressure(nonNegativeInteger), unitBloodPressure(string)]

### **18. Como saber a data do último monitoramento do sinal vital frequência de pulso de um determinado paciente?**

Classes:

MonitoringPulseRate, 'pulse rate', 'pulse rate measurement datum', Person e InstantEvent

Relações:

MonitoringPulseRate (hasParticipant) Person

MonitoringPulseRate (startDateTime) InstantEvent

'pulse rate measurement datum' (isMeasurementPulseRate) 'pulse rate'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'pulse rate measurement datum'[valuePulseRate(nonNegativeInteger), unitPulseRate(string)]

### **19. Como saber a data do último monitoramento do sinal vital frequência respiratória de um determinado paciente?**

Classes:

MonitoringRespiratoryRate, 'respiratory rate', 'respiratory rate measurement datum', Person e InstantEvent

Relações:

MonitoringRespiratoryRate (hasParticipant) Person

MonitoringRespiratoryRate (startDateTime) InstantEvent

'respiratory rate measurement datum' (isMeasurementRespiratoryRate) 'respiratory rate'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'respiratory rate measurement datum'[valueRespiratoryRate(nonNegativeInteger), unitRespiratoryRate(string)]

### **20. Como saber a data do último monitoramento do sinal vital saturação de oxigênio de um determinado paciente?**

Classes:

MonitoringOxygenSaturation, 'oxygen saturation', 'oxygen saturation measurement datum', Person e InstantEvent

Relações:

MonitoringOxygenSaturation (hasParticipant) Person

MonitoringOxygenSaturation (startDateTime) InstantEvent

'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen saturation'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'oxygen saturation measurement datum'[valueOxygenSaturation(nonNegativeInteger), unitOxygenSaturation(string)]

## **21. Quais são os sinais vitais que a ontologia de monitoramento de sinais vitais do paciente irá cobrir?**

As subclasses da classe 'vital sign' representam os sinais vitais que a ontologia cobre.

## **22. É possível parametrizar os sinais vitais de um paciente, ou seja, permitir atribuir valores que expressem a sua condição de normalidade?**

Na ontologia existe a classe ParameterizedVitalSign que possui as subclasses e atributos abaixo descritos que parametrizam os valores de normalidade adequado ao perfil do paciente.

Classes:

BloodPressure Diastolic [valueMaximumBloodPressureDiastolic (nonNegativeInteger), valueMinimumBloodPressureDiastolic (nonNegativeInteger), unitParameterizedValueBloodPressureDiastolic(string)]

Systolic [valueMaximumBloodPressureSystolic (nonNegativeInteger), valueMinimumBloodPressureSystolic (nonNegativeInteger), unitParameterizedValueBloodPressureSystolic(string)]

'Oxygen Saturation' [valueMaximumOxygenSaturation (nonNegativeInteger), valueMinimumOxygenSaturation (nonNegativeInteger), unitParameterizedValueOxygenSaturation(string)]

PulseRate [valueMaximumPulseRate (nonNegativeInteger), valueMinimumPulseRate (nonNegativeInteger), unitParameterizedValuePulseRate(string)]

RespiratoryRate [valueMaximumRespiratoryRate (nonNegativeInteger), valueMinimumRespiratoryRate (nonNegativeInteger), unitParameterizedValueRespiratoryRate(string)]

**23. Quais são os atores que participam de um determinado monitoramento de sinal vital de algum paciente?**

Classes:

MonitoringOxygenSaturation, Person e RoleOfMember

Relações:

MonitoringOxygenSaturation (hasParticipant) Person

Person (hasRole) RoleOfMember

**24. É possível obter um histórico do quadro clínico do paciente para cada sinal vital?**

Sim, através dos relacionamentos das subclasses:

'body temperature measurement datum' (isMeasurementTemperature) 'body temperature'

MonitoringTemperature (hasMonitoringTemperature) 'body temperature'

'blood pressure measurement datum' (isMeasurementBloodPressure) 'blood pressure' MonitoringBloodPressure (hasMonitoringBloodPressure) 'blood pressure'

'pulse rate measurement datum' (isMeasurementPulseRate) 'pulse rate' MonitoringPulseRate (hasMonitoringPulseRate) 'pulse rate'

'respiratory rate measurement datum' (isMeasurementRespiratoryRate) 'respiratory rate'

MonitoringRespiratoryRate (hasMonitoringRespiratoryRate) 'respiratory rate'

'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen saturation'

MonitoringOxygenSaturation (hasMonitoringOxygenSaturation) 'oxygen saturation'

**25. Quais sinais vitais foram parametrizados individualmente para um determinado paciente?**

Foi criado na classe 'vital sign' um atributo booleano chamado isParameterizedValue que tem o status atribuído como true se o sinal vital deve ser parametrizado para o paciente, caso não tenha a necessidade de ser parametrizado então o status do atributo isParameterizedValue tem o valor false. Existem os seguintes relacionamentos entre o sinal vital e os valores parametrizados:

'blood pressure' (hasParameterizedValueBloodPressure) BloodPressure  
 'pulse rate' (hasParameterizedValuePulseRate) PulseRate  
 'respiratory rate' (hasParameterizedValueRespiratoryRate) RespiratoryRate  
 'oxygen saturation' (hasParameterizedValueOxygenSaturation) 'Oxygen Saturation'

**26. É possível obter quando foi realizado cada monitoramento do sinal vital temperatura corpórea do paciente?**

Classes:

MonitoringTemperature, 'body temperature', 'body temperature measurement datum' Person e InstantEvent

Relações:

MonitoringTemperature (hasParticipant) Person

MonitoringTemperature (startDateTime) InstantEvent

'body temperature measurement datum' (isMeasurementTemperature) 'body temperature'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'body temperature measurement datum'[valueTemperature(nonNegativeInteger), unitTemperature(string)]

**27. É possível obter quando foi realizado cada monitoramento do sinal vital pressão arterial do paciente?**

Classes:

MonitoringBloodPressure, 'blood pressure', 'blood pressure measurement datum', Person e InstantEvent

Relações:

MonitoringBloodPressure (hasParticipant) Person

MonitoringBloodPressure (startDateTime) InstantEvent

'blood pressure measurement datum' (isMeasurementBloodPressure) 'blood pressure'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'blood pressure measurement datum'[valueDiastolicBloodPressure(nonNegativeInteger), valueSystolicBloodPressure, unitBloodPressure(nonNegativeInteger), unitBloodPressure(string)]

**28. É possível obter quando foi realizado cada monitoramento do sinal vital frequên-**

**cia de pulso do paciente?**

Classes:

MonitoringPulseRate, 'pulse rate', 'pulse rate measurement datum', Person e InstantEvent

Relações:

MonitoringPulseRate (hasParticipant) Person

MonitoringPulseRate (startDateTime) InstantEvent

'pulse rate measurement datum' (isMeasurementPulseRate) 'pulse rate'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'pulse rate measurement datum'[valuePulseRate(nonNegativeInteger), unitPulseRate(string)]

**29. É possível obter quando foi realizado cada monitoramento do sinal vital frequência respiratória do paciente?**

Classes:

MonitoringRespiratoryRate, 'respiratory rate', 'respiratory rate measurement datum', Person e InstantEvent

Relações:

MonitoringRespiratoryRate (hasParticipant) Person

MonitoringRespiratoryRate (startDateTime) InstantEvent

'respiratory rate measurement datum' (isMeasurementRespiratoryRate) 'respiratory rate'

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

'respiratory rate measurement datum'[valueRespiratoryRate(nonNegativeInteger), unitRespiratoryRate(string)]

**30. É possível obter quando foi realizado cada monitoramento do sinal vital saturação de oxigênio do paciente?**

Classes:

MonitoringOxygenSaturation, 'oxygen saturation', 'oxygen saturation measurement datum', Person e InstantEvent

Relações:

MonitoringOxygenSaturation (hasParticipant) Person

MonitoringOxygenSaturation (startDateTime) InstantEvent

'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen

saturation’

Atributos:

InstantEvent[instantCalendarClockDataType(dateTime)]

Person[hasName(string)]

’oxygen saturation measurement datum’[valueOxygenSaturation(nonNegativeInteger), unitOxygenSaturation(string)]

**31. É possível saber em qual região do corpo de um paciente ocorreu a medição do sinal vital temperatura corpórea?**

Classes:

MonitoringTemperature, ’body temperature measurement datum’, ’body temperature’, e ’Anatomical entity’

Relações:

MonitoringTemperature (hasMonitoringTemperature) ’body temperature’ ’body temperature measurement datum’ (isMeasurementTemperature) ’body temperature’ ’body temperature’ (inheres\_in) ’Anatomical entity’

**32. É possível saber em qual região do corpo de um paciente ocorreu a medição do sinal vital pressão arterial?**

Classes:

MonitoringBloodPressure, ’blood pressure measurement datum’, ’blood pressure’, e ’Anatomical entity’

Relações:

MonitoringBloodPressure (hasMonitoringBloodPressure) ’blood pressure’ ’blood pressure measurement datum’ (isMeasurementBloodPressure) ’blood pressure’ ’blood pressure’ (inheres\_in) ’Anatomical entity’

**33. É possível saber em qual região do corpo de um paciente ocorreu a medição do sinal vital frequência de pulso?**

Classes:

MonitoringPulseRate, ’pulse rate measurement datum’, ’pulse rate’, e ’Anatomical entity’

Relações:

MonitoringPulseRate (hasMonitoringPulseRate) ’pulse rate’ ’pulse rate measurement datum’ (isMeasurementPulseRate) ’pulse rate’ ’pulse rate’ (inheres\_in) ’Anatomical entity’

**34. É possível saber em qual região do corpo de um paciente ocorreu a medição do**

**sinal vital saturação de oxigênio?**

Classes:

MonitoringOxygenSaturation, 'oxygen saturation measurement datum', 'oxygen saturation', e 'Anatomical entity'

Relações:

MonitoringOxygenSaturation (hasMonitoringOxygenSaturation) 'oxygen saturation' 'oxygen saturation measurement datum' (isMeasurementOxygenSaturation) 'oxygen saturation'

'oxygen saturation' (inheres\_in) 'Anatomical entity'

**35. É possível saber a localização física de um paciente?**

Sim. Uma vez que a classe MonitoringVitalSign é uma subclasse da classe Activity, que descreve atividades como eventos espaço-temporais, ele pode relacionar com as subclasses da classe PhysicalLocation para descobrir a localização do paciente.

**36. É possível definir quais sinais vitais do paciente serão monitorados?**

Sim. Através do relacionamento entre as subclasses de MonitoringVitalSign e de 'vital sign'.

**37. Um período de tempo específico pode ser determinado em termos de hora, dia, semana, mês, etc?**

A ontologia de Monitoramento de Sinais Vitais Humanos (MSVH) importa a ontologia Activity do modelo SeCoM. A ontologia Activity por sua vez importa outras ontologias, sendo que uma delas se refere a ontologia de tempo. A classe MonitoringVitalSign da ontologia MSVH estende a classe Activity da ontologia Activity que descreve atividades como eventos espaço-temporais, herdando assim, atributos e relações temporais que podem ser descritos na forma de segundo, minuto, hora, dia, semana, mês e ano.

## Alarmes Expressos como Regras SWRL

O modelo ontológico proposto para monitoramento de sinais vitais humanos foi estendido com regras da lógica de predicados da linguagem SWRL. O objetivo é promover a inferência dos resultados das medições dos sinais vitais para representar os alarmes associados a essas medições.

A seguir são apresentadas as regras SWRL associadas a cada sinal vital descrito na ontologia e a classificação sugerida para valores normais e fora da normalidade.

### B.1 Alarmes para Pressão Arterial

#### B.1.1 Pressão Arterial utilizando a tabela populacional

Os valores de pressão arterial seguem a Classificação da Pressão Arterial (Maiores de 18 anos), conforme a VI Diretrizes Brasileiras de Hipertensão Arterial [13].

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Ótima	< 120	< 80	'blood pressure measurement datum'(?bpd), nonNegativeInteger[> 90 , < 120](?ts), nonNegativeInteger[< 80](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> OptimumBloodPressure(?bpd)
Normal	< 130	< 85	'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 120 , <= 129](?ts), nonNegativeInteger[>= 80 , <= 84](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> NormalBloodPressure(?bpd)
Limítrofe	130 – 139	85 - 89	'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 130 , <= 139](?ts), nonNegativeInteger[>= 85 , <= 89](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> LimitropheBloodPressure(?bpd)
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 140 , <= 159](?ts), nonNegativeInteger[>= 90 , <= 99](?td), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false), nonNegativeInteger[>= 160, <= 179](?ts), nonNegativeInteger[>= 100, <= 109](?td) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)

Hipertensão Estágio 3	$\geq 180$	$\geq 110$	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 110$ ](?td), nonNegativeInteger[ $\geq 180$ ](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Hipertensão Sistólica Isolada	$\geq 140$	$< 90$	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 140$ ](?ts), nonNegativeInteger[ $< 90$ ](?td), isMeasurementBloodPressure(?bpd, ?bp), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> IsolatedSystolicHypertension(?bpd), IsolatedSystolicHypertensionAlarm(?bpd)
Hipotensão Arterial	$< 90$		'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 80$ , $\leq 90$ ](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> HypotensionBloodPressure(?bpd), HypotensionBloodPressureAlarm(?bpd)

**Tabela B.1:** *Pressão Arterial - valor populacional*

**Pressão Diastólica em categoria diferente, ou seja, alteração da classificação em razão da pressão diastólica ser a de maior valor na classificação**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Ótima	< 120	< 80	'blood pressure measurement datum'(?bpd), nonNegativeInteger[> 90 , < 120](?ts), nonNegativeInteger[< 80](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> OptimumBloodPressure(?bpd)
Normal	< 120	< 80 - 84	'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 80 , <= 84](?td), nonNegativeInteger[>= 90 , < 120](?ts), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> NormalBloodPressure(?bpd)
Limítrofe	< 120	85 - 89	'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 85 , <= 89](?td), nonNegativeInteger[>= 90 , < 120](?ts), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> LimitropheBloodPressure(?bpd)
Normal	120 – 129	80 - 84	
Limítrofe	120 – 129	85 – 89	'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 120 , <= 129](?ts), nonNegativeInteger[>= 85 , <= 89](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> LimitropheBloodPressure(?bpd)
Hipertensão Estágio 1	120 – 129	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false), nonNegativeInteger[>=90, <= 99](?td), nonNegativeInteger[>=120, <= 129](?ts) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	120 – 129	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 100 , <= 109](?td), nonNegativeInteger[>= 120 , <= 129](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	120 – 129	>= 110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 110](?td), nonNegativeInteger[>= 120 , <= 129](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Limítrofe	130 – 139	85 - 89	
Hipertensão Estágio 1	130 – 139	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 130 , <= 139](?ts), nonNegativeInteger[>= 90 , <= 99](?td), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	130 – 139	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 100 , <= 109](?td), nonNegativeInteger[>= 130 , <= 139](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	130 – 139	>=110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 110](?td), nonNegativeInteger[>= 130 , <= 139](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Hipertensão Estágio 1	140 – 159	90 - 99	
Hipertensão Estágio 2	140 – 159	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[>= 100 , <= 109](?td), nonNegativeInteger[>= 140 , <= 159](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)

Hipertensão Estágio 3	140 – 159	$\geq 110$	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 110$ ](?td), nonNegativeInteger[ $\geq 140$ , $\leq 159$ ](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	
Hipertensão Estágio 3	160 – 179	$\geq 110$	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 110$ ](?td), nonNegativeInteger[ $\geq 160$ , $\leq 179$ ](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.2:** *Pressão Arterial – Pressão Diastólica em categoria diferente*

**Pressão Sistólica em categoria diferente, ou seja, alteração da classificação em razão da pressão sistólica ser a de maior valor na classificação**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Ótima	< 120	< 80	
Normal	120 – 129	< 80	'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 120$ , $\leq 129$ ](?ts), nonNegativeInteger[ $< 80$ ](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> NormalBloodPressure(?bpd)
Limítrofe	130 – 139	< 80	'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 130$ , $\leq 139$ ](?ts), nonNegativeInteger[ $< 80$ ](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> LimitropheBloodPressure(?bpd)
Normal	120 – 129	80 - 84	
Limítrofe	130 – 139	80 - 84	'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 130$ , $\leq 139$ ](?ts), nonNegativeInteger[ $\geq 80$ , $\leq 84$ ](?td), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts) -> LimitropheBloodPressure(?bpd)
Hipertensão Estágio 1	140 – 159	90 - 99	
Hipertensão Estágio 2	160 – 179	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 160$ , $\leq 179$ ](?ts), nonNegativeInteger[ $\geq 90$ , $\leq 99$ ](?td), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	$\geq 180$	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 180$ ](?ts), nonNegativeInteger[ $\geq 90$ , $\leq 99$ ](?td), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	
Hipertensão Estágio 3	$\geq 180$	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), nonNegativeInteger[ $\geq 100$ , $\leq 109$ ](?td), nonNegativeInteger[ $\geq 180$ ](?ts), isMeasurementBloodPressure(?bpd, ?bp), isParameterizedValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, false) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.3:** *Pressão Arterial – Pressão Sistólica em categoria diferente*

### B.1.2 Pressão Arterial utilizando o valor individualizado

Classificação da pressão arterial de pacientes que possuem valores parametrizados individualmente. Esta classificação é realizada apenas para informar a faixa de classificação que é considerada aceitável para o paciente. Os valores parametrizados de pressão arterial seguem a Classificação da Pressão Arterial (Maiores de 18 anos), conforme a VI Diretrizes Brasileiras de Hipertensão Arterial [13].

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 90), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 140), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?td, 99), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 159) -> Stage1Hypertension(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 100), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 160), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 110), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 180), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?ts, ?vmaxs) -> Stage3Hypertension(?bpd)
Hipertensão Sistólica Isolada	> =140	< 90	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 140), lessThan(?td, 90), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?ts, ?vmaxs) -> SystolicHypertension(?bpd)
Hipotensão Arterial	< 90		'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 80), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 90) -> HypotensionBloodPressure(?bpd)

**Tabela B.4:** Pressão Arterial – valor individualizado

**Utilizando valores de medidas das pressões sistólicas e diastólica maiores que os valores máximos individualizados para as pressões sistólica e diastólica**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThan(?td, ?vmaxd), greaterThanOrEqual(?td, 90), greaterThanOrEqual(?ts, 140), lessThanOrEqual(?td, 99), greaterThan(?ts, ?vmaxs), lessThanOrEqual(?ts, 159) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThan(?td, ?vmaxd), greaterThan(?ts, ?vmaxs), greaterThanOrEqual(?td, 100), greaterThanOrEqual(?ts, 160), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThan(?td, ?vmaxd), greaterThan(?ts, ?vmaxs), greaterThanOrEqual(?td, 110), greaterThanOrEqual(?ts, 180) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.5:** *Pressão Arterial – valores de medidas das pressões sistólicas e diastólica maiores que os valores máximos individualizados para as pressões sistólica e diastólica*

**Utilizando valores de medidas das pressões sistólicas e diastólica menores que os valores mínimos individualizados para as pressões sistólica e diastólica – Sem disparar alarme, somente classificado-os**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqualTo(?td, 90), greaterThanOrEqualTo(?ts, 140), lessThan(?td, ?vmind), lessThan(?ts, ?vmins), lessThanOrEqualTo(?td, 99), lessThanOrEqualTo(?ts, 159) -> Stage1Hypertension(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqualTo(?td, 100), greaterThanOrEqualTo(?ts, 160), lessThan(?td, ?vmind), lessThan(?ts, ?vmins), lessThanOrEqualTo(?td, 109), lessThanOrEqualTo(?ts, 179) -> Stage2Hypertension(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqualTo(?td, 110), greaterThanOrEqualTo(?ts, 180), lessThan(?td, ?vmind), lessThan(?ts, ?vmins) -> Stage3Hypertension(?bpd)

**Tabela B.6:** *Pressão Arterial – valores de medidas das pressões sistólicas e diastólica menores que os valores mínimos individualizados para as pressões sistólica e diastólica*

**Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThan(?td, ?vmaxd), greaterThanOrEqual(?td, 90), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 140), lessThanOrEqual(?td, 99), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 159) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThan(?td, ?vmaxd), greaterThanOrEqual(?td, 100), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 160), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThan(?td, ?vmaxd), greaterThanOrEqual(?td, 110), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 180), lessThanOrEqual(?ts, ?vmaxs) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.7:** *Pressão Arterial – valor da medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica*

**Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 90), greaterThan(?ts, ?vmaxs), greaterThanOrEqual(?ts, 140), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?td, 99), lessThanOrEqual(?ts, 159) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 100), greaterThan(?ts, ?vmaxs), greaterThanOrEqual(?ts, 160), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 110), greaterThan(?ts, ?vmaxs), greaterThanOrEqual(?ts, 180), lessThanOrEqual(?td, ?vmaxd) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.8:** *Pressão Arterial – valor da medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica*

**Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica menor que o valor individualizado mínimo da diastólica**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 90), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 140), lessThan(?td, ?vmind), lessThanOrEqual(?td, 99), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 159) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 100), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 160), lessThan(?td, ?vmind), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 110), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 180), lessThan(?td, ?vmind), lessThanOrEqual(?ts, ?vmaxs) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.9:** *Pressão Arterial – valor da medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica menor que o valor individualizado mínimo da diastólica*

**Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica menor que o valor individualizado mínimo da sistólica**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 1	140 – 159	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 90), lessThan(?ts, ?vmins), greaterThanOrEqual(?ts, 140), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?td, 99), lessThanOrEqual(?ts, 159) -> Stage1Hypertension(?bpd), Stage1HypertensionAlarm(?bpd)
Hipertensão Estágio 2	160 – 179	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 100), lessThan(?ts, ?vmins), greaterThanOrEqual(?ts, 160), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	> =110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?td, 110), lessThan(?ts, ?vmins), greaterThanOrEqual(?ts, 180), lessThanOrEqual(?td, ?vmaxd) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.10:** *Pressão Arterial – valor da medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica menor que o valor individualizado mínimo da sistólica*

**Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica e de maior valor na classificação**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 2	140 – 159	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 100), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 140), greaterThan(?td, ?vmaxd), lessThanOrEqual(?td, 109), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 159) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	140 – 159	>=110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 110), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 140), greaterThan(?td, ?vmaxd), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 159) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Hipertensão Estágio 3	160 – 179	>=110	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 110), greaterThanOrEqual(?ts, ?vmins), greaterThanOrEqual(?ts, 160), greaterThan(?td, ?vmaxd), lessThanOrEqual(?ts, ?vmaxs), lessThanOrEqual(?ts, 179) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.11:** *Pressão Arterial – valor da medida da pressão sistólica entre os valores individualizados mínimo e máximo da sistólica e a medida da pressão diastólica maior que o valor individualizado máximo da diastólica e de maior valor na classificação*

**Utilizando valores de medidas das pressões sistólicas e diastólica, mantendo a medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica e de maior valor na classificação**

Classificação	Pressão Sistólica (mmHg)	Pressão Diastólica (mmHg)	Regra
Hipertensão Estágio 2	160 - 179	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 90), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?ts, 160), greaterThan(?ts, ?vmaxs), lessThanOrEqual(?td, 99), lessThanOrEqual(?td, ?vmaxd), lessThanOrEqual(?ts, 179) -> Stage2Hypertension(?bpd), Stage2HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	90 - 99	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 90), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?ts, 180), greaterThan(?ts, ?vmaxs), lessThanOrEqual(?td, 99), lessThanOrEqual(?td, ?vmaxd) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)
Hipertensão Estágio 3	>= 180	100 - 109	'blood pressure'(?bp), 'blood pressure measurement datum'(?bpd), Diastolic(?bdr), Systolic(?bsr), isMeasurement(?bpd, ?bp), hasReferenceValue(?bp, ?bdr), hasReferenceValue(?bp, ?bsr), useReferenceValue(?bp, ?rv), valueDiastolicBloodPressure(?bpd, ?td), valueMaximum(?bdr, ?vmaxd), valueMaximum(?bsr, ?vmaxs), valueMinimum(?bdr, ?vmind), valueMinimum(?bsr, ?vmins), valueSystolicBloodPressure(?bpd, ?ts), equal(?rv, true), greaterThanOrEqual(?td, 100), greaterThanOrEqual(?td, ?vmind), greaterThanOrEqual(?ts, 180), greaterThan(?ts, ?vmaxs), lessThanOrEqual(?td, 109), lessThanOrEqual(?td, ?vmaxd) -> Stage3Hypertension(?bpd), Stage3HypertensionAlarm(?bpd)

**Tabela B.12:** *Pressão Arterial – valor da medida da pressão diastólica entre os valores individualizados mínimo e máximo da diastólica e a medida da pressão sistólica maior que o valor individualizado máximo da sistólica e de maior valor na classificação*

## B.2 Alarmes para Temperatura Corpórea

Os valores da temperatura corpórea axilar seguem a Classificação da Semiologia Médica (Maiores de 18 anos), conforme Porto e Porto [29].

Classificação	Temperatura (°C)	Regra
Normal	35,5 – 37	'body temperature measurement datum'(?td), float[>= 35.5f , <= 37.0f](?t), valueTemperature(?td, ?t) -> NormalTemperature(?td)
Febre leve ou febrícula	37,1 – 37,5	'body temperature'(?bt), 'body temperature measurement datum'(?td), float[>= 37.1f , <= 37.5f](?t), isMeasurementTemperature(?td, ?bt), isParameterizedValue(?bt, ?rv), valueTemperature(?td, ?t), equal(?rv, false) -> LowFever(?td), LowFeverAlarm(?td)
Febre moderada	37,6 – 38,5	'body temperature'(?bt), 'body temperature measurement datum'(?td), float[> 37.5f , <= 38.5f](?t), isMeasurementTemperature(?td, ?bt), isParameterizedValue(?bt, ?rv), valueTemperature(?td, ?t), equal(?rv, false) -> ModerateFever(?td), ModerateFeverAlarm(?td)
Febre alta	> 38,5	'body temperature'(?bt), 'body temperature measurement datum'(?td), float[> 38.5f](?t), isMeasurementTemperature(?td, ?bt), isParameterizedValue(?bt, ?rv), valueTemperature(?td, ?t), equal(?rv, false) -> HighFever(?td), HighFeverAlarm(?td)
Hipotermia leve	32 – 35	'body temperature'(?bt), 'body temperature measurement datum'(?td), float[>= 32.0f , <= 35.0f](?t), isMeasurementTemperature(?td, ?bt), isParameterizedValue(?bt, ?rv), valueTemperature(?td, ?t), equal(?rv, false) -> MildHypothermia(?td), MildHypothermiaAlarm(?td)
Hipotermia moderada	30 – 31,9	'body temperature'(?bt), 'body temperature measurement datum'(?td), float[>= 30.0f , < 32.0f](?t), isMeasurementTemperature(?td, ?bt), isParameterizedValue(?bt, ?rv), valueTemperature(?td, ?t), equal(?rv, false) -> ModerateHypothermia(?td), ModerateHypothermiaAlarm(?td)
Hipotermia grave	< 30	'body temperature'(?bt), 'body temperature measurement datum'(?td), float[< 30.0f](?t), isMeasurementTemperature(?td, ?bt), isParameterizedValue(?bt, ?rv), valueTemperature(?td, ?t), equal(?rv, false) -> SevereHypothermia(?td), SevereHypothermiaAlarm(?td)

**Tabela B.13:** *Temperatura Corpórea Axilar*

## B.3 Alarmes para Frequência de Pulso

Os valores para frequência de pulso (Maiores de 18 anos) seguem a classificação apresentada por Potter e Perry [30].

### B.3.1 Frequência de Pulso utilizando a tabela populacional

Classificação	Batimentos por minuto (bpm)	Regra
Normal	60 - 100	'pulse rate measurement datum'(?pr), nonNegativeInteger[>= 60 , <= 100](?t), valuePulseRate(?pr, ?t) -> NormalPulseRate(?pr)
Bradycardia	< 60	'pulse rate'(?rp), 'pulse rate measurement datum'(?pr), isMeasurementPulseRate(?pr, ?rp), isParameterizedValue(?rp, ?rv), valuePulseRate(?pr, ?t), equal(?rv, false), nonNegativeInteger[< 60](?t) -> Bradycardia(?pr), BradycardiaAlarm(?pr)
Taquicardia	> 100	'pulse rate'(?rp), 'pulse rate measurement datum'(?pr), nonNegativeInteger[> 100](?t), isMeasurementPulseRate(?pr, ?rp), isParameterizedValue(?rp, ?rv), valuePulseRate(?pr, ?t), equal(?rv, false) -> Tachycardia(?pr), TachycardiaAlarm(?pr)

**Tabela B.14:** *Frequência de Pulso - valor populacional*

### B.3.2 Frequência de Pulso utilizando o valor individualizado

Classificação do valor da frequência de pulso conforme estabelecido pelo seu valor individualizado. Sem disparar alarme.

Classificação	Batimentos por minuto (bpm)	Regra
Bradycardia	< 60	'pulse rate'(?pr), 'pulse rate measurement datum'(?prd), PulseRate(?ipr), isMeasurement(?prd, ?pr), hasReferenceValue(?pr, ?ipr), useReferenceValue(?pr, ?rv), valuePulseRate(?prd, ?tpr), valueMaximum(?ipr, ?vmaxpr), valueMinimum(?ipr, ?vminpr), equal(?rv, true), greaterThanOrEqual(?tpr, ?vminpr), lessThanOrEqual(?tpr, ?vmaxpr), lessThan(?tpr, 60) -> Bradycardia(?prd)
Taquicardia	> 100	'pulse rate'(?pr), 'pulse rate measurement datum'(?prd), PulseRate(?ipr), isMeasurement(?prd, ?pr), hasReferenceValue(?pr, ?ipr), useReferenceValue(?pr, ?rv), valuePulseRate(?prd, ?tpr), valueMaximum(?ipr, ?vmaxpr), valueMinimum(?ipr, ?vminpr), equal(?rv, true), greaterThanOrEqual(?tpr, ?vminpr), lessThanOrEqual(?tpr, ?vmaxpr), greaterThan(?tpr, 100) -> Tachycardia(?prd)

**Tabela B.15:** *Frequência de Pulso - valor individualizado sem disparo de alarme*

**Utilizando o valor dentro de cada classificação. Disparando alarme, caso ultrapassem os valores parametrizados para a medição.**

Classificação	Batimentos por minuto (bpm)	Regra
Bradycardia	< 60	'pulse rate'(?pr), 'pulse rate measurement datum'(?prd), PulseRate(?ipr), isMeasurement(?prd, ?pr), hasReferenceValue(?pr, ?ipr), useReferenceValue(?pr, ?rv), valuePulseRate(?prd, ?tpr), valueMaximum(?ipr, ?vmaxpr), valueMinimum(?ipr, ?vminpr), equal(?rv, true), lessThan(?tpr, ?vminpr), lessThan(?tpr, 60) -> Bradycardia(?prd), BradycardiaAlarm(?prd)
Taquicardia	> 100	'pulse rate'(?pr), 'pulse rate measurement datum'(?prd), PulseRate(?ipr), isMeasurement(?prd, ?pr), hasReferenceValue(?pr, ?ipr), useReferenceValue(?pr, ?rv), valuePulseRate(?prd, ?tpr), valueMaximum(?ipr, ?vmaxpr), valueMinimum(?ipr, ?vminpr), equal(?rv, true), greaterThan(?tpr, ?vmaxpr), greaterThan(?tpr, 100) -> Tachycardia(?prd), TachycardiaAlarm(?prd)

**Tabela B.16:** *Frequência de Pulso - valor individualizado com disparo de alarme*

## B.4 Alarmes para Frequência Respiratória

Os valores para frequência respiratória (Maiores de 18 anos) seguem a classificação apresentada por Potter e Perry [30].

### B.4.1 Frequência Respiratória utilizando a tabela populacional

Classificação	Movimento de respiração por minuto (mrpm)	Regra
Eupnéia (Normal)	12 - 20	respiratory rate measurement datum'(?rd), nonNegativeInteger[>= 12 , <= 20](?t), valueRespiratoryRate(?rd, ?t) -> Eupnea(?rd)
Bradipnéia	< 12	'respiratory rate'(?rr), 'respiratory rate measurement datum'(?rd), nonNegativeInteger[< 12](?t), isMeasurementRespiratoryRate(?rd, ?rr), isParameterizedValue(?rr, ?rv), valueRespiratoryRate(?rd, ?t), equal(?rv, false) -> Bradypnea(?rd), BradypneaAlarm(?rd)
Taquipnéia	> 20	'respiratory rate'(?rr), 'respiratory rate measurement datum'(?rd), isMeasurementRespiratoryRate(?rd, ?rr), isParameterizedValue(?rr, ?rv), valueRespiratoryRate(?rd, ?t), equal(?rv, false), nonNegativeInteger[> 20](?t) -> Tachypnea(?rd), TachypneaAlarm(?rd)
Apnéia	0	'respiratory rate measurement datum'(?rd), valueRespiratoryRate(?rd, ?t), equal(?t, 0) -> Apnea(?rd), ApneaAlarm(?rd)

**Tabela B.17:** *Frequência respiratória – valor populacional*

### B.4.2 Frequência Respiratória utilizando o valor individualizado

Classificação do valor da frequência respiratória conforme estabelecido pelo seu valor individualizado. Sem disparar alarme.

Classificação	Movimento de respiração por minuto (mrpm)	Regra
Bradipnéia	< 12	'respiratory rate'(?rr), 'respiratory rate measurement datum'(?rd), RespiratoryRate(?irr), isMeasurement(?rd, ?rr), hasReferenceValue(?rr, ?irr), useReferenceValue(?rr, ?rv), valueRespiratoryRate(?rd, ?t), equal(?rv, true), valueMaximum(?irr, ?vmaxrr), valueMinimum(?irr, ?vminrr), greaterThanOrEqual(?t, ?vminrr), lessThanOrEqual(?t, ?vmaxrr), lessThan(?t, 12) -> Bradypnea(?rd)
Taquipnéia	> 20	'respiratory rate'(?rr), 'respiratory rate measurement datum'(?rd), RespiratoryRate(?irr), hasReferenceValue(?rr, ?irr), isMeasurement(?rd, ?rr), useReferenceValue(?rr, ?rv), valueMaximum(?irr, ?vmaxrr), valueMinimum(?irr, ?vminrr), valueRespiratoryRate(?rd, ?t), equal(?rv, true), greaterThan(?t, 20), greaterThanOrEqual(?t, ?vminrr), lessThanOrEqual(?t, ?vmaxrr) -> Tachypnea(?rd)

**Tabela B.18:** *Frequência respiratória – valor individualizado sem disparar alarme*

**Utilizando valor individualizado dentro de cada classificação. Disparando alarme, caso ultrapassem os valores parametrizados para a medição.**

Classificação	Movimento de respiração por minuto (mrpm)	Regra
Bradipnéia	< 12	'respiratory rate'(?rr), 'respiratory rate measurement datum'(?rd), RespiratoryRate(?irr), hasReferenceValue(?rr, ?irr), isMeasurement(?rd, ?rr), useReferenceValue(?rr, ?rv), valueMaximum(?irr, ?vmaxrr), valueMinimum(?irr, ?vminrr), valueRespiratoryRate(?rd, ?t), equal(?rv, true), lessThan(?t, ?vminrr), lessThan(?t, 12) -> Bradypnea(?rd), BradypneaAlarm(?rd)
Taquipnéia	> 20	'respiratory rate'(?rr), 'respiratory rate measurement datum'(?rd), RespiratoryRate(?irr), hasReferenceValue(?rr, ?irr), isMeasurement(?rd, ?rr), useReferenceValue(?rr, ?rv), valueMaximum(?irr, ?vmaxrr), valueMinimum(?irr, ?vminrr), valueRespiratoryRate(?rd, ?t), equal(?rv, true), greaterThan(?t, ?vmaxrr), greaterThan(?t, 20) -> Tachypnea(?rd), TachypneaAlarm(?rd)

**Tabela B.19:** *Frequência respiratória – valor individualizado com disparo de alarme*

## B.5 Alarmes para Saturação de Oxigênio

Os valores para saturação de oxigênio (Maiores de 18 anos) seguem a classificação apresentada por Potter e Perry [30].

### B.5.1 Saturação de Oxigênio utilizando a tabela populacional

Classificação	Porcentagem da Saturação de Oxigênio (%)	Regra
Normal	95 - 100	'oxygen saturation measurement datum'(?osd), nonNegativeInteger[>= 95](?t), valueOxygenSaturation(?osd, ?t) -> NormalOxygenSaturation(?osd)
Hipoxemia	< 95	'oxygen saturation'(?os), 'oxygen saturation measurement datum'(?osd), nonNegativeInteger[<95](?t), isMeasurementOxygenSaturation(?osd, ?os), isParameterizedValue(?os, ?vb), valueOxygenSaturation(?osd, ?t), equal(?vb, false) -> Hypoxemia(?osd), HypoxemiaAlarm(?osd)

**Tabela B.20:** *Saturação de Oxigênio (SpO2) – valor populacional*

### B.5.2 Saturação de Oxigênio utilizando o valor individualizado

Classificação do valor da saturação de oxigênio conforme estabelecido pelo seu valor individualizado. Sem disparar alarme.

Classificação	Porcentagem da Saturação de Oxigênio (%)	Regra
Hipoxemia	< 95	'oxygen saturation'(?os), 'oxygen saturation measurement datum'(?osd), 'Oxygen Saturation'(?ios), isMeasurement(?osd, ?os), hasReferenceValue(?os, ?ios), useReferenceValue(?os, ?vb), valueMinimum(?ios, ?vminos), valueMaximum(?ios, ?vmaxos), valueOxygenSaturation(?osd, ?t), greaterThanOrEqual(?t, ?vminos), lessThanOrEqual(?t, ?vmaxos), equal(?vb, true), lessThan(?t, 95) -> Hypoxemia(?osd)

**Tabela B.21:** *Saturação de Oxigênio (SpO2) – valor individualizado sem disparo de alarme*

**Utilizando valor individualizado dentro de cada classificação. Disparando alarme, caso ultrapassem os valores parametrizados para a medição.**

Classificação	Porcentagem da Saturação de Oxigênio (%)	Regra
Hipoxemia	< 95	'oxygen saturation'(?os), 'oxygen saturation measurement datum'(?osd), 'Oxygen Saturation'(?ios), isMeasurement(?osd, ?os), hasReferenceValue(?os, ?ios), useReferenceValue(?os, ?vb), valueMinimum(?ios, ?vminos), valueMaximum(?ios, ?vmaxos), valueOxygenSaturation(?osd, ?t), lessThan(?t, ?vminos), equal(?vb, true), lessThan(?t, 95) -> Hypoxemia(?osd), HypoxemiaAlarm(?osd)

**Tabela B.22:** *Saturação de Oxigênio (SpO2) – valor individualizado com disparo de alarme*