



# **UNIVERSIDADE FEDERAL DO MARANHÃO**

**Programa de Pós-Graduação em Ciência da  
Computação**

**Allan Fábio de Aguiar Barbosa**

***Uma Abordagem Baseada em Linguagens  
Específicas de Domínio e Transformação de  
Modelos para a Geração Semiautomática de  
Aplicações mHealth para Escores Clínicos***

**São Luís  
2019**

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Allan Fábio de Aguiar Barbosa

*Uma Abordagem Baseada em Linguagens Específicas de Domínio  
e Transformação de Modelos para a Geração Semiautomática de  
Aplicações mHealth para Escores Clínicos*

São Luís  
2019

Allan Fábio de Aguiar Barbosa

*Uma Abordagem Baseada em Linguagens Específicas de Domínio  
e Transformação de Modelos para a Geração Semiautomática de  
Aplicações mHealth para Escores Clínicos*

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Maranhão como requisito parcial para a obtenção do grau de MESTRE em Ciência da Computação.

**Orientador: Francisco José da Silva e Silva**

**Doutor em Ciência da Computação – UFMA**

**Coorientador: Luciano Reis Coutinho**

**Doutor em Ciências – UFMA**

São Luís

2019

Barbosa, Allan Fábio de Aguiar

Uma Abordagem Baseada em Linguagens Específicas de Domínio e Transformação de Modelos para a Geração Semiautomática de Aplicações mHealth para Escores Clínicos / Allan Fábio de Aguiar Barbosa. – São Luís, 2019.

134 f.

Orientador: Francisco José da Silva e Silva. Coorientador: Luciano Reis Coutinho.

Impresso por computador (fotocópia).

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Ciência da Computação. São Luís, 2019.

1. Escores Clínicos. 2. Mobile Health. 3. Linguagem de Modelagem Específica de Domínio. 4. Transformação de Modelos. I. da Silva e Silva, Francisco José, orient. II. Título.

CDU 004

Allan Fábio de Aguiar Barbosa

*Uma Abordagem Baseada em Linguagens Específicas de Domínio  
e Transformação de Modelos para a Geração Semiautomática de  
Aplicações mHealth para Escores Clínicos*

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Allan Fábio de Aguiar Barbosa e aprovada pela comissão examinadora.

Aprovada em 14 de junho de 2019.

**BANCA EXAMINADORA**

---

Francisco José da Silva e Silva (orientador)

Doutor em Ciência da Computação – UFMA

---

Luciano Reis Coutinho (coorientador)

Doutor em Ciências – UFMA

---

Markus Endler

Dr.rer.nat. em Informática – PUC-Rio

---

Davi Viana dos Santos

Doutor em Informática – UFMA

*A Deus em primeiro lugar,  
meu pai (em memória),  
minha mãe, meus irmãos e  
amigos, e meus professores.*

## Resumo

Na medicina, escores clínicos são ferramentas criadas para prever resultados clínicos, realizar estratificação de risco, ajudar na tomada de decisão clínica, avaliar a gravidade de uma doença ou auxiliar o diagnóstico. Estas ferramentas geralmente atuam no contexto de doenças crônicas, sendo estruturadas por uma seção de variáveis, uma seção de regras para cálculo da pontuação do escore e uma seção de avaliações. Recentemente, iniciativas surgiram no âmbito de *Mobile Health* para resolver automaticamente problemas de escore clínico. Estas abordagens se limitam a desenvolver um sistema computacional para cada especificidade de escore. Neste cenário, a produtividade é afetada pois toda vez que precisarmos implementar uma aplicação para um novo escore, todas as etapas de um processo de desenvolvimento de software tradicional serão repetidas. Além disso, a complexidade intrínseca ao ambiente tecnológico que envolve tais aplicações requer conhecimento especializado de diversas áreas da Medicina e Ciência da Computação. Esta dissertação apresenta uma nova abordagem para o desenvolvimento de software focada na automatização de escores clínicos em ambientes de computação ubíqua denominada de *MDD4ClinicalScores*. O *MDD4ClinicalScores* provê a geração semiautomática de componentes de software através do uso de metamodelagem e transformação de modelos. Estes componentes são submetidos ao ambiente de execução implementado pela abordagem para a criação de uma aplicação móvel de monitoramento de pacientes baseada no escore clínico previamente especificado. A abordagem proposta foi avaliada sob dois aspectos: a expressividade da linguagem *DSML4ClinicalScore*, por meio da metamodelagem de oito casos concretos, e a viabilidade do *MDD4ClinicalScores*, por meio do desenvolvimento de dois estudos de caso.

**Palavras-chave:** Escores Clínicos, Mobile Health, Linguagem de Modelagem Específica de Domínio, Transformação de Modelos.

## Abstract

In medicine, clinical scores are tools created to predict clinical outcomes, perform risk stratification, aid in clinical decision making, assess disease severity or assist diagnosis. These tools usually act in the context of chronic diseases, and they are structured by a section of variables, by a section of rules for calculating the score, and by a section of evaluations. Recently, initiatives have emerged in the mobile health field to automatically solve clinical score problems. These approaches are limited because they develop a computational system for each particular clinical score. In this scenario, productivity is affected because every time we need to develop an application for a new clinical score, all steps of the traditional software development process will be repeated. Besides, the complexity involving the environment of such applications and its technology demands specialized knowledge of many areas of Medicine and Computer Science. This thesis presents a new approach called *MDD4ClinicalScores* for software development focusing on the automation of clinical scores in the ubiquitous computing environments. *MDD4ClinicalScores* provides automatic generation of software components through the use of metamodeling and model transformation. These components are submitted to the proposed execution environment for the creation of a mobile application for patient monitoring based on the previously specified clinical score. The proposed approach is evaluated through two aspects: the expressiveness of the *DSML4ClinicalScore* language through the metamodeling of eight concrete cases, and the *MDD4ClinicalScores* viability through the development of two case studies..

**Keywords:** Clinical Scores, Mobile Health, Domain-Specific Modeling Language, Model Transformation.

## Agradecimentos

A Deus em primeiro lugar, pela minha vida e por todas as bênçãos que tem derramado sobre mim.

A meu orientador, o Prof. Francisco José da Silva e Silva, por ter me auxiliado nessa jornada de dois anos, mostrando-se sempre presente e compreensivo na orientação deste trabalho.

A meu co-orientador, o Prof. Luciano Reis Coutinho, pelas inúmeras vezes em que se mostrou disposto e paciente comigo, contribuindo decisivamente na proposta principal deste trabalho.

Ao professor Mário Antônio Meireles Teixeira (UFMA), meu antigo orientador da graduação e principal incentivador deste desafio gratificante que foi o Mestrado em Ciência da Computação desta universidade.

Aos professores Markus Endler (PUC-Rio) e Davi Viana (UFMA) por terem aceito fazer parte da banca examinadora.

Aos meus familiares pelo apoio que me deram nesta trajetória. Eles me fazem querer ser alguém melhor todos os dias.

Ao membros do LSDi-UFMA, com os quais compartilhei alegrias, tristezas, e conhecimento.

A UFMA e ao PPGCC pela estrutura disponibilizada para execução deste trabalho.

A Leila Ferreira Moreira dos Santos Barbosa e Antônio Eduardo dos Santos Barbosa, pelo apoio e compreensão durante esses anos de mestrado.

*“O sucesso é ir de fracasso em fracasso sem perder o entusiasmo”.*

*Winston Churchill*

## Lista de Figuras

2.1	Visão geral da MDA [Mateen e Tabassum 2016]. . . . .	27
2.2	Arquitetura típica para serviços mHealth [Silva et al. 2015]. . . . .	28
2.3	Mapa conceitual de processamento de eventos [Fülöp et al. 2010]. . . . .	31
2.4	Arquitetura do SDDL [David et al. 2013]. . . . .	35
2.5	Componentes do M-Hub [Rios et al. 2015]. . . . .	37
2.6	Componentes básicos da OWL [Saha 2007]. . . . .	40
3.1	Aplicativo ACTAPP para avaliar o controle da asma. <b>A</b> , vídeos embutidos e mensagens de texto proporcionam uma experiência amigável para o usuário. <b>B</b> , alertas pró-ativos engajam o usuário no cuidado da asma. <b>C</b> , avaliação do controle da asma pelo ACT [13]. . . . .	45
3.2	Aplicativo móvel RPCRC para estimativa de risco de câncer de próstata [Pereira-Azevedo et al. 2017]. . . . .	48
3.3	Visões da interface de usuário do aplicativo <i>cloudUPDRS</i> [Stamate et al. 2018]. . . . .	51
3.4	Modelo de pontuação do escore IMSS [Aminian et al. 2017c]. . . . .	54
3.5	Telas do aplicativo BSDMC [Aminian et al. 2017b]. . . . .	55
4.1	Arquitetura do <i>MDD4ClinicalScores</i> . . . . .	58
4.2	Mapa Conceitual para uma especificação de escore clínico. . . . .	61
4.3	Metamodelo da <i>DSML4ClinicalScore</i> . . . . .	62
4.4	Exemplo de aplicação escrita conforme a linguagem <i>Deklaer</i> [Pinheiro et al. 2018]. . . . .	68
4.5	Transformação de um objeto <i>Rule</i> da <i>DSML4ClinicalScore</i> para um indivíduo <i>ActRule</i> da <i>Deklaer</i> . . . . .	71

4.6	Ambiente de Execução do <i>MDD4ClinicalScores</i> . . . . .	73
4.7	Exemplo de uma instância da classe <code>SENSORObs</code> . . . . .	73
5.1	Escore MEWS conforme o metamodelo da <i>DSML4ClinicalScore</i> . . . . .	79
5.2	Composição da regra “ <b>id:String = BPB71a80</b> ” do escore MEWS conforme o metamodelo da <i>DSML4ClinicalScore</i> . . . . .	80
5.3	Aplicação mHealth gerada para o escore MEWS. . . . .	84
5.4	Métricas da aplicação mHealth do escore MEWS. . . . .	85
5.5	Execução da aplicação mHealth de monitoramento para o escore MEWS. . . . .	86
5.6	Escore CURB-65 conforme <i>DSML4ClinicalScore</i> . . . . .	88
5.7	Aplicação mHealth gerada para o escore CURB-65. . . . .	89
5.8	Métricas da aplicação mHealth do escore CURB-65. . . . .	89
5.9	Execução da aplicação de monitoramento para o escore CURB-65. . . . .	90

## Lista de Tabelas

2.1	Áreas de aplicação para escores clínicos. . . . .	20
2.2	Especificação do escore MEWS. . . . .	23
2.3	Classificações para DSLs. . . . .	25
3.1	Asthma Control Test versão em português [Roxo et al. 2016]. . . . .	44
4.1	Mapeamento entre os conceitos da <i>DSML4ClinicalScore</i> e <i>Deklaer</i> . . . . .	69
5.1	Relação de escores clínicos usados para validação da <i>DSML4ClinicalScore</i> . . . . .	78
5.2	Especificação do escore CURB-65. . . . .	87

# Sumário

<b>Lista de Figuras</b>	<b>vi</b>
<b>Lista de Tabelas</b>	<b>viii</b>
<b>1 Introdução</b>	<b>15</b>
1.1 Objetivos . . . . .	17
1.2 Estrutura da Dissertação . . . . .	18
<b>2 Fundamentação Teórica</b>	<b>19</b>
2.1 Escores Clínicos . . . . .	19
2.1.1 Metodologia de Desenvolvimento . . . . .	21
2.1.2 Estrutura . . . . .	22
2.2 Model-Driven Development . . . . .	23
2.2.1 Domain-Specific Language . . . . .	24
2.2.2 Transformação de Modelos . . . . .	26
2.3 Mobile Health . . . . .	28
2.3.1 Taxonomia . . . . .	29
2.4 Complex Event Processing . . . . .	31
2.4.1 Aplicabilidade da Tecnologia CEP . . . . .	33
2.5 ContextNet . . . . .	34
2.5.1 Scalable Data Distribution Layer . . . . .	35
2.5.2 Mobile Hub . . . . .	36
2.6 Web Semântica e Ontologias . . . . .	38
2.6.1 Web Ontology Language . . . . .	38

2.7	Considerações Finais . . . . .	40
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>42</b>
3.1	ACTAPP . . . . .	43
3.1.1	Teste de Controle de Asma . . . . .	43
3.1.2	Aplicativo ACTAPP . . . . .	43
3.2	RPCRC . . . . .	45
3.2.1	Escores RPCRC . . . . .	46
3.2.2	Projeto do Aplicativo RPCRC . . . . .	47
3.3	cloudUPDRS . . . . .	49
3.3.1	Escala UPDRS e Questionário PDQ39 . . . . .	50
3.3.2	Sistema e Aplicativo cloudUPDRS . . . . .	50
3.4	BSDMC . . . . .	52
3.4.1	Escores SGRC, RPVTBS e IMSS . . . . .	53
3.4.2	Aplicativo BSDMC . . . . .	55
3.5	Considerações Finais . . . . .	56
<b>4</b>	<b>MDD4ClinicalScores</b>	<b>57</b>
4.1	Visão Geral do MDD4ClinicalScores . . . . .	57
4.2	A Linguagem DSML4ClinicalScore . . . . .	59
4.2.1	Análise do Domínio . . . . .	60
4.2.2	Projeto da Linguagem . . . . .	61
4.3	Geração Automática por Transformação de Modelos . . . . .	65
4.3.1	A Linguagem Deklaer . . . . .	66
4.3.2	Transformação M2T de DSML4ClinicalScore para Deklaer . . . . .	68
4.4	Ambiente de Execução . . . . .	72
4.5	Considerações Finais . . . . .	75

<b>5</b>	<b>Avaliação do MDD4ClinicalScores</b>	<b>77</b>
5.1	Validação da DSML4ClinicalScore . . . . .	77
5.2	Estudos de Caso . . . . .	82
5.2.1	Escore MEWS . . . . .	83
5.2.2	Escore CURB-65 . . . . .	86
5.3	Discussão . . . . .	91
5.4	Limitações . . . . .	93
5.5	Considerações Finais . . . . .	93
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>95</b>
	<b>Referências Bibliográficas</b>	<b>99</b>

# 1 Introdução

Doenças crônicas constituem tema bastante recorrente na área da saúde e um desafio árduo para seus gestores em qualquer nível de governo, seja federal, estadual ou municipal. Essas doenças caracterizam-se pelo prognóstico incerto de longa ou indefinida duração, curso clínico mutável ao longo do tempo com possíveis períodos de agravamento e geração de incapacidades, e processo de cuidado contínuo que nem sempre leva à cura [Saúde 2013]. Segundo a *Organização Mundial da Saúde* (OMS), as doenças crônicas representam a principal causa de morte no mundo e responderão por 73% dos óbitos e 60% da carga global de doenças no ano de 2020 [Machado 2006].

No Brasil, *Doença Crônica Não Transmissível* (DCNT) compõe o problema de saúde de maior magnitude e responde por mais de 70% das causas de morte no país, de acordo com a Pesquisa Nacional de Saúde de 2013. Dentre as principais DCNTs, temos as doenças cardiovasculares, câncer, diabetes, enfermidades respiratórias crônicas e doenças neuropsiquiátricas. Essas doenças geralmente provocam perda de qualidade de vida, incapacidades e alto grau de limitação nas pessoas por elas acometidas. Esse índice elevado de mortes por DCNTs levou o *Ministério da Saúde* (MS) a elaborar o Plano de Ações Estratégicas para o Enfrentamento das Doenças Crônicas Não Transmissíveis, no período de 2011 a 2022 [Cavararo 2014].

Além das ações por parte do poder público, o combate às DCNTs vem ganhando forte apoio da comunidade acadêmico-científica nos últimos anos. Uma dessas iniciativas passa pela criação de ferramentas como os escores clínicos, que buscam auxiliar tanto no rápido diagnóstico clínico quanto no direcionamento do tratamento mais adequado. O suporte dessas ferramentas é importante nesse contexto especialmente para a população de baixa renda, devido principalmente a uma rede pública de saúde deficitária e ao alto custo associado aos serviços privados.

Dessa forma, escores clínicos tem ganhado cada vez mais ênfase como parte da moderna prática clínica. No geral, essas ferramentas são criadas para prever resultados clínicos, realizar estratificação de risco, ajudar na tomada de decisão

clínica, avaliar a gravidade da doença ou auxiliar o diagnóstico [Aakre et al. 2017]. Na literatura médica, existem inúmeros escores clínicos formalmente definidos, cada qual lidando com um tipo específico de doença, principalmente aquelas tidas como crônicas. Por exemplo, temos escores clínicos para doenças cardíacas, doenças infecciosas, doenças neurológicas, e assim por diante.

Outra iniciativa que também busca diminuir gastos com tratamento médico e hospitalar é a *Mobile Health* (mHealth). mHealth consiste no uso e capitalização das principais utilidades de dispositivos móveis, como voz e mensagem de texto, bem como funcionalidades mais complexas, no suporte à prática médica e de saúde pública [Olla e Shimskey 2015]. Para [Silva et al. 2015], mHealth é a inovação em cuidados de saúde, pois objetiva entregar serviços médicos a qualquer hora e em qualquer lugar, ultrapassando barreiras geográficas, temporais e até organizacionais. Com os recentes avanços em tecnologia móvel e *Internet of Things* (IoT), grandes oportunidades podem surgir para melhorar a saúde e bem-estar de pacientes em todo mundo, agregando modelos específicos de escore clínico com aplicações mHealth. Exemplificando, mHealth possibilita o desenvolvimento de aplicações voltadas para o monitoramento remoto de pacientes através de sensores corporais ou espalhados no ambiente, que enviam notificações a profissionais da saúde responsáveis pelo acompanhamento desses pacientes, de modo que alguma ação relacionada ao tratamento possa ser realizada. Essa ação pode ser embasada em regras previamente definidas em cima do cálculo da pontuação de um escore clínico específico.

Todavia, os dados provenientes de escores clínicos são obtidos tradicionalmente de forma manual, o que pode acarretar valor de entrada e resultado incorretos, além de envolver eventualmente cálculos matemáticos complexos. Além disso, as atuais abordagens em mHealth que lidam com esse problema computacionalmente são limitadas, porque um software é desenvolvido para cada especificidade de escore clínico. Portanto, duas questões estão diretamente associadas à resolução de escores clínicos no contexto de ambientes de computação ubíqua:

1. **Produtividade:** as modernas soluções móveis para escores clínicos requerem a construção de um novo software para cada novo escore clínico desenvolvido, mesmo que essas especificações tenham características similares, repetindo assim todas as etapas do processo de desenvolvimento, o que afeta a produtividade.

2. **Complexidade:** a elaboração de software para automatização de escores clínicos é complexo, necessitando de conhecimento especializado da área da qual o escore clínico é derivado, assim como uma equipe de analistas com experiência no desenvolvimento de aplicações para ambientes de computação móvel e redes de sensores pessoais; os dados necessários para avaliar o escore clínico podem ser oriundos de várias fontes, tais como dados de sensores, dados de prontuário eletrônico do paciente, ou dados manuais obtidos por entrevistas com o paciente ou profissional da saúde.

Neste trabalho apresentamos uma nova abordagem para o desenvolvimento de software voltado para a automatização de escores clínicos através do uso de computação ubíqua, denominada de *MDD4ClinicalScores*. Esta abordagem possui como diferencial, com relação às demais descritas na literatura, a aplicação de conceitos de *Model Driven Development* (MDD) que simplifica o processo de desenvolvimento de aplicações mHealth para escores clínicos. Em particular, esta proposta é baseada em uma *Domain-Specific Modeling Language* (DSML) de alto nível que utiliza o metamodelo Ecore para descrever uma especificação de escore clínico, denominada de *DSML4ClinicalScore*. Esta abordagem também fornece regras de transformação que permitem gerar, a partir do modelo concreto especificado, classes de uma ontologia em *Web Ontology Language* (OWL). Por fim, esta ontologia é submetida a um *middleware* que semiautomatiza a geração de uma aplicação móvel para IoT com foco no escore clínico previamente modelado pela *DSML4ClinicalScore*.

## 1.1 Objetivos

A pesquisa à qual se refere esta dissertação de mestrado tem como objetivo geral conceber uma proposta para o desenvolvimento de software capaz de auxiliar desenvolvedores de aplicações e pesquisadores em geral no processo de modelagem e geração semiautomática de software para resolução de escores clínicos no contexto de ambientes de computação ubíqua.

Os objetivos específicos desta pesquisa são:

- Levantar o estado da arte em soluções computacionais para escores clínicos voltadas para ambientes de computação ubíqua;

- Projetar e implementar funcionalidades para a modelagem de especificações de escore clínico de modo geral.
- Projetar e implementar funcionalidades para a criação de regras de transformação que possibilitem a geração de artefatos de código executáveis.
- Avaliar a viabilidade da abordagem proposta através do desenvolvimento de estudos de caso com um escore clínico específico, que permeie todas as etapas do processo de modelagem e geração de uma aplicação mHealth nesse contexto.

## 1.2 Estrutura da Dissertação

Esta dissertação está estruturada como segue: o Capítulo 2 aborda os principais conceitos que formam o arcabouço teórico desta pesquisa, trazendo algumas noções sobre escores clínicos, MDD, mHealth, *Complex Event Processing* (CEP), Web Semântica e ontologias.

O Capítulo 3 apresenta trabalhos relevantes que estão inseridos no mesmo contexto desta pesquisa, pontuando as mais recentes soluções mHealth para automatização de escores clínicos descritas na literatura.

O Capítulo 4 apresenta a abordagem proposta por esta pesquisa, o *MDD4ClinicalScores*, que foca na automatização de escores clínicos em ambientes de computação ubíqua. Este capítulo explica as motivações e os requisitos para o desenvolvimento desta abordagem, enfatizando como ocorre esse processo de automatização: desde a especificação do escore clínico através de uma linguagem de modelagem de domínio específico, até a geração da aplicação mHealth para monitoramento remoto de pacientes que usa a avaliação deste escore.

O Capítulo 5 discute os procedimentos adotados para avaliação da abordagem aqui proposta. Esta avaliação acontece em duas etapas distintas: o processo de validação da linguagem de modelagem de domínio específico e a elaboração de estudos de caso para provar a viabilidade da proposta do *MDD4ClinicalScores* e exemplificar sua utilização.

O Capítulo 6 apresenta as conclusões obtidas por meio desta pesquisa, e trabalhos futuros que podem ser desenvolvidos a partir deste esforço inicial.

## 2 Fundamentação Teórica

Nas últimas décadas, a utilização de escores clínicos por profissionais da saúde tornou-se cada vez mais frequente dentro de ambientes hospitalares. Prova disso é que rotineiramente surgem pesquisas sobre a avaliação do impacto da adoção dessas ferramentas para a prática clínica [1][Aakre et al. 2017]. Em ambientes de computação ubíqua, as soluções computacionais vigentes combinam técnicas de *clouding computing* e mHealth para a resolução de escores clínicos específicos. Diferente dessas abordagens, nossa proposta combina conceitos de DSML e regras de transformação *Model to Text* (M2T) para resolver um número maior de escores clínicos.

Neste capítulo são apresentados conceitos e terminologias que fundamentaram o desenvolvimento desta pesquisa. Aqui são abordados sucintamente aspectos teóricos sobre escores clínicos, MDD, mHealth, CEP, middlewares *ContextNet*, *Scalable Data Distribution Layer* (SDDL) e *Mobile Hub* (M-Hub), Web Semântica e ontologias.

### 2.1 Escores Clínicos

Uma tarefa importante dentro da medicina moderna é traduzir achados empíricos em diagnósticos ou prognósticos válidos para utilidade clínica [21]. Nesse cenário, pesquisadores desenvolvem mecanismos que podem auxiliar os profissionais da saúde no desempenho de suas atividades cotidianas, como os escores clínicos. O termo **escore clínico** ou **sistema de pontuação clínica** vem da terminologia em inglês *Clinical Scoring System* (CSS). Na literatura, existem ainda outros termos como *Clinical Prediction Model* (CPM) ou *Risk Prediction Model* (RPM). No geral, todas essas expressões referem-se ao conceito de escore clínico.

Escores clínicos são modelos estatísticos que combinam um número de características ou preditores (por exemplo, relacionados ao paciente, à doença ou ao tratamento) para prever um diagnóstico ou prognóstico. Tipicamente um número limitado de preditores entre 2 e 20 é considerado [Steyerberg 2009]. Esses modelos

podem ser aplicados em diversos cenários, tais como triagem de indivíduos de alto risco para doença assintomática, previsão de eventos futuros como doença ou morte, assistência na tomada de decisão médica, e educação em saúde [31]. Ademais, os escores clínicos podem aumentar o entendimento sobre a efetividade do tratamento e otimização do uso de recursos hospitalares, ajudando no desenvolvimento de padrões de tratamento [Rapsang e Shyam 2014]. A Tabela 2.1 resume as principais áreas em que escores clínicos podem ser aplicados [Steyerberg 2009]:

**Tabela 2.1:** Áreas de aplicação para escores clínicos.

Área de aplicação	Objetivo	Exemplo
<i>Saúde pública</i>	Intervenções preventivas	Incidência de doença
<i>Prática clínica</i>	Elaboração de diagnóstico	Recomendação de teste
	Tomada de decisão terapêutica	Início do tratamento Tomada de decisão cirúrgica Tratamento intensivo Tratamento prolongado
<i>Pesquisa</i>	Inclusão em ensaio clínico Ajuste de ensaio clínico Ajuste de um escore	

De acordo com [Thompson 2012], escores clínicos são baseados em *Clinical Prediction Rule* (CPR), que é uma ferramenta que utiliza critérios específicos para estabelecer probabilidades de resultados ou auxiliar em decisões de gerenciamento. O aspecto mais relevante das CPRs é que elas quantificam a contribuição de sintomas, sinais clínicos e testes de diagnóstico, e estratificam pacientes conforme a probabilidade de terem um distúrbio alvo. O resultado pode ser diverso dentro do espectro diagnóstico, prognóstico ou terapêutico [Falk e Fahey 2009]. A combinação de um número limitado dessas CPRs forma o esqueleto de uma especificação de escore clínico. Desse modo, existem três tipos de CPRs:

- **CPRs diagnósticas:** focam em fatores relacionados ao alcance do diagnóstico clínico;
- **CPRs prognósticas:** preveem resultados;
- **CPRs prescritivas:** fornecem recomendações para intervenção clínica.

O propósito desta pesquisa consiste em apresentar uma nova abordagem para o desenvolvimento de software dentro do contexto de escores clínicos e computação ubíqua.

### 2.1.1 Metodologia de Desenvolvimento

O processo de formalização de um escore clínico é demasiado complexo e demanda tempo. Segundo [Lee et al. 2016], as etapas necessárias para a criação de um modelo compreendem:

1. **Preparação:** antes de executar qualquer etapa visando o estabelecimento de uma especificação de escore clínico, primeiro se deve responder algumas questões de pesquisa, tais como: (1) qual é o resultado (evento ou doença) que se quer prever (por exemplo, diabetes ou fratura)? (2) quem é o paciente alvo do modelo (por exemplo, população em geral ou população idosa acima dos 65 anos)? (3) quem é o usuário alvo do modelo (por exemplo, pessoas leigas ou doutores)?
2. **Seleção da base de dados:** essa etapa é a mais importante durante o processo de formalização do escore clínico; pesquisadores devem utilizar diferentes tipos de base de dados, dependendo do propósito do modelo (por exemplo, base de dados primária, secundária ou administrativa).
3. **Manipulação de variáveis:** as bases de dados disponíveis possuem mais variáveis do que racionalmente serão aproveitadas na especificação do escore clínico; geralmente a inclusão de mais de 10 variáveis pode diminuir a eficiência, viabilidade e conveniência do modelo; sendo assim, devem ser feitas uma avaliação e seleção de preditores mais sensíveis e preditivos.
4. **Geração do modelo:** apesar de não existir consenso na literatura sobre qual metodologia deve ser adotada, diversas estratégias com métodos estatísticos são amplamente utilizadas para a criação de um escore clínico (por exemplo, modelos lineares ou logísticos, e análise regressiva).
5. **Avaliação e validação (interna e externa):** para a avaliação de uma especificação de escore clínico, podem ser usadas técnicas de calibração (plotagem de eventos observados contra probabilidades previstas para grupos definidos por alcance

de risco individual previsto) e discriminação (habilidade de distinguir eventos de não-eventos); para o processo de validação do modelo, internamente podem ser utilizadas uma amostra aleatória menor ou anos diferentes da base de dados escolhida e, externamente, uma base de dados diferente daquela adotada para a geração do modelo.

### 2.1.2 Estrutura

Salvo algumas exceções, a estrutura básica de uma especificação de escore clínico segue o modelo apresentado na Tabela 2.2. Mais detalhadamente, a Tabela 2.2 mostra os componentes do escore clínico *Modified Early Warning Score* (MEWS) [Subbe et al. 2001], que determina o grau de enfermidade de um paciente.

De modo geral, a especificação de um escore clínico consiste de: (1) uma seção para definição de **variáveis**; (2) uma seção para definição das **regras** para cálculo da pontuação; e (3) uma seção para definição dos modelos de **avaliação**. As variáveis são os preditores clínicos levantados na etapa de manipulação da especificação descrita na Subseção 2.1.1. As regras definem a pontuação de cada variável especificada no escore clínico, segundo a metodologia empregada na etapa de geração do modelo (ver Subseção 2.1.1). As avaliações descrevem a interpretação da especificação de acordo com o resultado obtido pela aplicação das regras para cálculo da pontuação.

Exemplificando, o escore clínico MEWS possui 5 (cinco) variáveis relacionadas a: pressão sanguínea sistólica, batimento cardíaco, frequência respiratória, temperatura, e escore *Alert, Voice, Pain, Unresponsive* (AVPU). Essa especificação conta com 23 (vinte e três) regras para cálculo da sua pontuação, sendo 5 (cinco) regras associadas à pressão sanguínea sistólica, 6 (seis) regras associadas ao batimento cardíaco, 5 (cinco) regras associadas à frequência respiratória, 3 (três) regras associadas à temperatura, e 4 (quatro) regras associadas ao escore AVPU. Finalmente, o escore clínico MEWS define 2 (duas) avaliações possíveis para seu resultado, cuja interpretação determinará a probabilidade de uma doença levar o paciente à morte ou sua admissão em uma *Unidade de Terapia Intensiva* (UTI).

**Tabela 2.2:** Especificação do escore MEWS.

Variável	Valor	Pontuação
<i>Pressão sanguínea sistólica</i>	$\leq 70$ mmHg	+3
	71–80 mmHg	+2
	81–100 mmHg	+1
	101–199 mmHg	0
	$\geq 200$ mmHg	+2
<i>Batimento cardíaco</i>	$< 40$ bpm	+2
	41–50 bpm	+1
	51–100 bpm	0
	101–110 bpm	+1
	111–129 bpm	+2
	$\geq 130$ bpm	+3
<i>Frequência respiratória</i>	$< 9$ bpm	+2
	9–14 bpm	0
	15–20 bpm	+1
	21–29 bpm	+2
	$\geq 30$ bpm	+3
<i>Temperatura</i>	$< 35^{\circ}C$	+2
	$35\text{--}38.4^{\circ}C$	0
	$\geq 38.5^{\circ}C$	+2
<i>Escore AVPU</i>	Alerta	0
	Reage a voz	+1
	Reage a dor	+2
	Sem resposta	+3
<b>Avaliação</b>		<b>Resultado</b>
Sem conexão com morte ou admissão em UTI		$\leq 4$
Estatisticamente associado à morte ou admissão em UTI		$\geq 5$

## 2.2 Model-Driven Development

Para o MDD, o elemento chave em um processo de desenvolvimento de software é a modelagem de sistemas. Na literatura, terminologias como *Model-Driven Engineering* (MDE), *Model-Driven Software Development* (MDSD), *Model-Driven Software Engineering* (MDSE) ou *Model-Driven Architecture* (MDA) também são utilizadas como sinônimos do MDD. Vários especialistas dessa área propuseram suas definições para a mencionada abordagem, sendo que esta seção pontua algumas destas assertivas.

[Voelter et al. 2010] afirmam que usar modelos de uma maneira prescritiva é a essência do MDD. Para o autor, a modelagem de sistemas pode ser entendida sob duas visões: descritiva ou prescritiva. Um modelo descritivo representa um sistema

existente, sendo usado apenas para discussão, comunicação e análise. Por outro lado, um modelo prescritivo é aquele que pode ser utilizado para gerar automaticamente um sistema alvo.

Para [Gronback 2009], o foco principal do MDD é desenvolver e refinar o modelo de um domínio específico, afim de prover um vocabulário padrão para uso no processo de desenvolvimento. O autor enfatiza que o trabalho principal concentra-se na construção de um modelo do domínio ou uma *Domain-Specific Language* (DSL), que então será aproveitado para produzir o software final.

Por sua vez, [Kleppe 2008] preconiza que MDD tem seu alicerce em cima de dois conceitos principais: **plataforma** e **transformação de modelos**. Para o autor, a noção de plataforma é central dentro dessa abordagem, pois os detalhes de engenharia e tecnologia são irrelevantes para a funcionalidade precípua de um componente de software. Apesar de um modelo ser expresso em uma linguagem predefinida, ele possui nível de abstração maior que outros produtos de software, ocultando as particularidades da plataforma que será escolhida como destino final. Já a transformação de modelos é a técnica que permite que um modelo independente de plataforma seja traduzido para outro modelo dependente de plataforma, possibilitando assim que modelos produzam artefatos de software executáveis.

Para resolver computacionalmente o problema de escores clínicos em ambientes de computação ubíqua, a proposta desta pesquisa combina duas técnicas de MDD: DSL e transformação de modelos.

### 2.2.1 Domain-Specific Language

Conforme explicado na Seção 2.2, DSL é parte integrante do MDD. De acordo com [Fowler 2010], as DSLs fundamentam-se em quatro componentes essenciais:

- **Linguagem de programação:** como qualquer linguagem de programação, a estrutura de uma DSL é elaborada para facilitar o entendimento pelas pessoas, mas deve ser executada por um computador.

- **Natureza da linguagem:** a expressividade da linguagem deve partir tanto de suas expressões individuais quanto da forma como elas podem ser combinadas.
- **Expressividade limitada:** diferente das linguagens de programação de propósito geral, uma DSL suporta apenas o mínimo de características necessário para o entendimento de seu domínio; portanto, não se pode construir todo um sistema por meio de uma DSL, mas somente um aspecto particular dele.
- **Foco no domínio:** uma linguagem limitada só é útil se tiver um foco claro em um domínio pequeno; o foco no domínio é o que efetiva uma linguagem limitada.

DSL é baseada em abstrações que são fortemente conectadas com o domínio para o qual a linguagem é destinada. Linguagens especializadas usualmente contam com uma sintaxe única e bem descritiva para expressar essas abstrações de forma concisa, podendo variar de notações textuais a símbolos ou gráficos [Voelter et al. 2010]. A Tabela 2.3 resume os principais tipos de DSL descritos na literatura.

**Tabela 2.3:** Classificações para DSLs.

<b>Classificação segundo [Fowler 2010].</b>	
<b>Tipo</b>	<b>Definição</b>
<i>Externa</i>	Linguagem separada daquela principal da aplicação com a qual trabalha. Exemplo: <i>Structured Query Language</i> (SQL).
<i>Interna</i>	Maneira particular de usar uma linguagem de programação de propósito geral. Exemplo: <i>linguagem Lisp</i> .
<i>Bancada de linguagem</i>	<i>Integrated Development Environment</i> (IDE) especializado em definir e construir DSLs. Exemplo: <i>Eclipse Modeling Framework</i> (EMF).
<b>Classificação segundo [Kleppe 2008].</b>	
<b>Tipo</b>	<b>Definição</b>
<i>Horizontal</i>	Aplicada a um grande grupo de aplicações que compartilham a mesma característica técnica. Exemplo: <i>aplicação Web</i> .
<i>Vertical</i>	Indica que os conceitos aplicados não são provenientes da Ciência da Computação, mas sim do negócio para o qual se destinam. Exemplo: <i>linguagem para descrever restrições de mobilidade</i> .

Ademais, existe um caso especial de DSL, mais conhecida como DSML. DSMLs servem para formalizar a estrutura, o comportamento e os requisitos de uma aplicação por meio de metamodelos, cujas representações definem relacionamentos semânticos e restrições entre conceitos em um domínio [Schmidt 2006]. A definição de uma linguagem de modelagem geralmente começa por essa captura e identificação do domínio. O resultado dessa atividade produz a sintaxe abstrata dessa linguagem, que corresponde a um metamodelo com todos os conceitos identificados em nível de metadomínio. A sintaxe concreta da linguagem refere-se a como ela será entendida e usada por seus usuários, da forma em que os modelos serão escritos e projetados [38][Fowler 2010].

## 2.2.2 Transformação de Modelos

A transformação de modelos é uma atividade que ganhou notoriedade com a promoção da MDA, uma metodologia de desenvolvimento de software cunhada pelo *Object Management Group* (OMG). OMG é uma organização internacional que desenvolve diversos padrões de tecnologia para o mercado de software em geral, como a *Unified Modeling Language* (UML) [Object Management Group 2019].

A definição de transformação pode ser expressa como a geração automática do modelo de destino a partir do modelo de origem, de acordo com uma **definição de transformação**. Uma definição de transformação é um conjunto de **regras de transformação** que juntas explicam como o modelo descrito na linguagem de origem pode ser transformado em outro modelo descrito na linguagem de destino. Uma regra de transformação é a descrição de como uma ou mais construções da linguagem de origem pode(m) ser transformada(s) em uma ou mais construções da linguagem de destino [30]. Todos esses conceitos estão traduzidos na Figura 2.1.

Logo, com base no número de modelos de entrada e saída presentes na arquitetura mostrada na Figura 2.1, o processo de transformação de modelos pode ser dividida em: (1) **transformação um-para-um**, quando se tratar apenas de um modelo de entrada e um modelo de saída; (2) **transformação um-para-muitos**, quando um modelo de entrada gera vários modelos de saída; (3) **transformação muitos-para-um**, quando vários modelos de entrada geram um modelo de saída; e (4) **transformação**

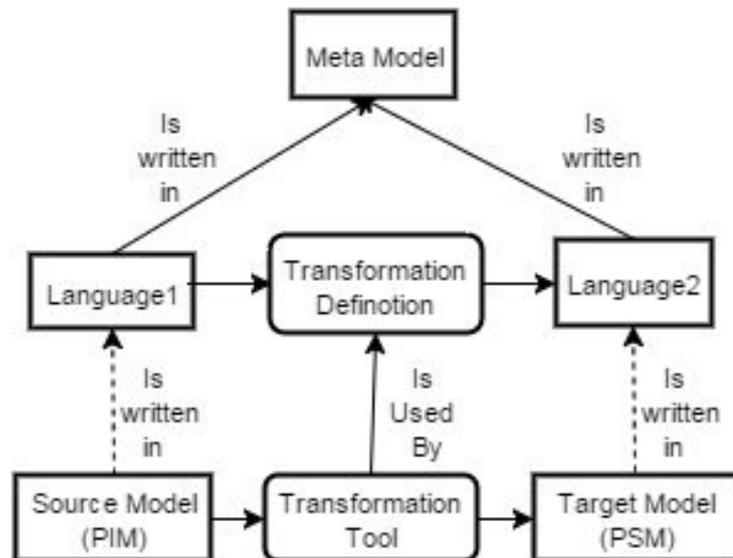


Figura 2.1: Visão geral da MDA [Mateen e Tabassum 2016].

**muitos-para-muitos**, quando vários modelos de entrada geram vários modelos de saída [10].

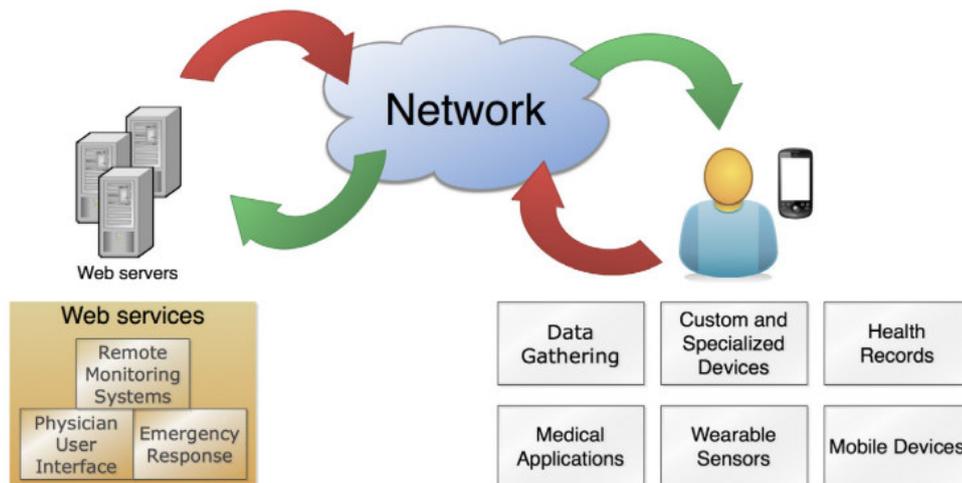
Além das categorias mencionadas acima, [Babau et al. 2010] estabeleceram duas novas taxonomias para transformação de modelos. A primeira taxonomia leva em consideração a linguagem de modelagem empregada. Assim, a transformação pode ser **endógena**, quando os modelos estão descritos na mesma linguagem, ou **exógena**, quando os modelos estão descritos em linguagens diferentes. A segunda taxonomia baseia-se no nível de abstração dos modelos de origem e destino. Desta maneira, a transformação pode ser **horizontal**, quando os modelos de origem e destino residem no mesmo nível de abstração, ou **vertical**, quando os modelos de origem e destino residem em níveis de abstração distintos.

Tradicionalmente, a geração de código dentro do MDD é alcançada pela aplicação de uma técnica de transformação de modelos. Para isso, é necessário percorrer a árvore sintática abstrata (modelo concreto especificado) e gerar código-fonte em uma linguagem de programação (ou outro texto). Como o código gerado é tratado apenas como texto, não há reconhecimento de linguagem para a qual se quer transformar [Voelter et al. 2010]. No âmbito do MDD, este processo chama-se Transformação M2T.

## 2.3 Mobile Health

A introdução de dispositivos móveis na década de 90 possibilitou aos médicos acesso fácil a registros médicos, resultados de exames laboratoriais, exames de imagem e informação sobre remédios. Para os pacientes, essa novidade trouxe a comodidade para acompanhar diagnósticos e monitorar doenças a qualquer hora e em qualquer lugar [Silva et al. 2015]. Estes aspectos caracterizam a tecnologia conhecida como mHealth.

Conceitualmente, o termo mHealth resume o uso de aplicações habilitadas para dispositivos móveis que coletam ou entregam informações e dados sobre cuidados em saúde [Sama et al. 2014]. Essas aplicações são tradicionalmente construídas em cima de dois componentes essenciais: (1) uma plataforma de hardware e sistema operacional e (2) os softwares que fornecem serviços relacionados à saúde física, bem estar, diagnóstico, prognóstico, prescrição médica, entre outros [Terry 2015].



**Figura 2.2:** Arquitetura típica para serviços mHealth [Silva et al. 2015].

Segundo [Silva et al. 2015], uma arquitetura típica para serviços mHealth segue o modelo apresentado na Figura 2.2. mHealth geralmente faz uso da Internet e de serviços Web para prover uma interação pervasiva entre médicos e pacientes. De um lado, a Internet disponibiliza serviços Web direcionados à saúde para consumo, tais como monitoramento remoto, emergência ou interface de usuário, e, de outro, interage com os usuários via sensores, dispositivos móveis, aplicativos médicos e demais interfaces disponíveis para receber requisições desses serviços ou coletar dados para eles.

Esta pesquisa está centrada no desenvolvimento de software na área de mHealth para resolução de escores clínicos, especificamente aplicações móveis voltadas para o monitoramento remoto de pacientes através da avaliação destes modelos de pontuação.

### 2.3.1 Taxonomia

Com muita propriedade, [Olla e Shimskey 2015] estabelecem uma taxonomia para mHealth de acordo com o propósito pretendido pela aplicação. Esta taxonomia é dividida em dimensões e segue as diretrizes a seguir:

1. **Casos de uso médico:** estas aplicações estão diretamente ligadas ao diagnóstico, monitoramento do paciente, bem estar, modificação do comportamento, conformidade, instrução, eficiência e produtividade, e saúde ambiental.
  - Cuidados com diagnóstico: utilizam dispositivos e testes para exames médicos no paciente ou perto dele, podendo ser um acessório móvel, aplicativo de software incorporado, dispositivos conectados, sensores corporais ou dérmicos. Exemplo: aplicativo para medir taxa de glicose no sangue.
  - Monitoramento do paciente: combinam tecnologias voltadas para o monitoramento de pacientes, visando ajudá-los na manutenção de sua independência, identificação de complicações e diminuir custos com o tratamento. Exemplo: aplicativo para indicar o grau da diabetes.
  - Bem estar: auxiliam o usuário na busca de um estilo de vida sadio com a promoção de comportamentos saudáveis. Exemplo: aplicativos para dietas ou exercícios.
  - Modificação de comportamento: objetivam ajudar usuários a modificar e melhorar sua saúde através da promoção, consciência e educação. Exemplo: aplicativo para fumantes.
  - Conformidade: estimam o quão um paciente está dedicado a determinado plano de assistência médica. Exemplo: aplicativo para gerenciamento da diabetes.

- Aplicações instrucionais: estão direcionadas à orientação de pacientes e profissionais da saúde. Exemplo: aplicativo de bulas de medicamentos.
- Eficiência e produtividade: englobam todas as aplicações voltadas para a eficiência e produtividade nos cuidados de saúde. Exemplo: aplicativo de um prontuário eletrônico.
- Saúde ambiental: informam sobre fatores ambientais que podem impactar a saúde de uma comunidade. Exemplo: aplicativo que informa sobre mudanças climáticas em uma cidade.

2. **Modalidades técnicas:** estas aplicações são classificadas consoante o tipo de dispositivo, interface do usuário, sistema operacional e características.

- Tipo de dispositivo: refere-se a que tipo de dispositivo é necessário para executar a aplicação, tais como *smartphones*, *tablets*, *smartwatches*, entre outros.
- Interface de usuário: diz respeito à interface de usuário da aplicação ser amigável ou não, com suporte a gestos, *widgets*, configurações personalizadas e outras facilidades.
- Sistema operacional: refere-se a qual sistema operacional é necessário para dar suporte à aplicação, tais como Windows, iOS ou Android.
- Características: diz respeito a quais características do dispositivo móvel são necessárias para executar a aplicação, como áudio, vídeo, texto, e-mail etc.

3. **Considerações políticas:** estas aplicações seguem regras para gerenciamento de dados, regulação governamental e segurança móvel.

- Gerenciamento de dados: enquadram os requisitos de armazenamento da aplicação para o dispositivo, servidor local e nuvem; aplicativos móveis possuem diferentes capacidades de armazenamento, geralmente variando entre 16, 32 e 64GB; o armazenamento em servidores locais e na nuvem devem ser considerados caso a aplicação faça uso dessas tecnologias.
- Regulação governamental: qualquer aplicação que propõe uma configuração clínica para uso por pacientes ou profissionais da saúde requer aprovação regulatória por agências governamentais.

- Segurança móvel: mecanismos de segurança como criptografia ou redes virtuais privadas devem ser implementados por se tratar de um ambiente em que informações sobre a saúde de pacientes são manipuladas.

## 2.4 Complex Event Processing

CEP pode ser definido como um conjunto de ferramentas e técnicas para análise e controle de eventos complexos interrelacionados que guiam os modernos sistemas de informação distribuídos. Esta tecnologia auxilia no entendimento mais adequado sobre um sistema, na resolução e identificação mais rápida de suas falhas, e na melhoria de sua operação, performance e segurança. CEP pode ser aplicado em uma série de problemas computacionais, tais como automação, controle e escalonamento de processos industriais, monitoramento e predição de desempenho de redes, e detecção de intrusão em sistemas computacionais [Luckham 2008].

Para entender o funcionamento de CEP, existe um conjunto de conceitos que formam sua estrutura metodológica. A Figura 2.3 resume estes conceitos. Para o CEP, entende-se que [Fülöp et al. 2010]:

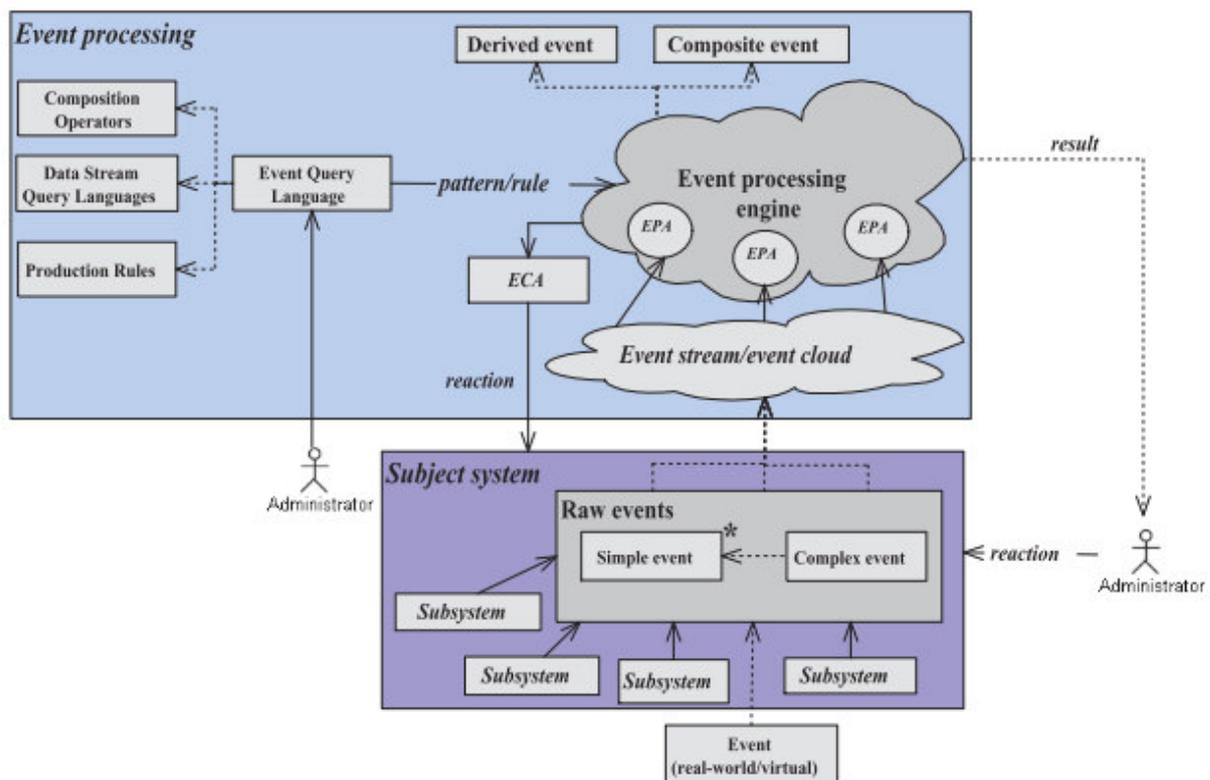


Figura 2.3: Mapa conceitual de processamento de eventos [Fülöp et al. 2010].

- Evento (**event**) é qualquer coisa que acontece ou é contemplada como algo ocorrido; para o processamento computacional, evento é um objeto que representa códigos ou registros em um evento.
- Evento simples (**simple event**) é um evento que não é uma abstração ou composição de outros eventos, registrando um evento no mundo real.
- Evento complexo (**complex event**) é um evento que é uma abstração de outros eventos, ou seja, é derivado da combinação de eventos simples usando um conjunto específico de construtores de eventos (disjunção, conjunção, sequência etc.).
- Tipo de evento (**event type**) é uma classe de objetos de evento.
- Atributo de evento (**event attribute**) é um componente da estrutura de um evento.
- Canal do evento (**event channel**) é um conduto pelo qual eventos são transmitidos de eventos fontes para eventos consumidores.
- Situação (**situation**) é uma sequência específica de eventos.
- Tempo de detecção (**detection time**) é o tempo decorrido para um evento ser detectado.
- Tempo de transação (**transaction time**) é o tempo decorrido para um evento ser processado.
- Motor para Processamento de Eventos (**Event Processing Engine**) é um conjunto de agentes para processamento de eventos e um conjunto de canais de eventos para conectá-los; geralmente os motores CEP utilizam uma linguagem de programação semelhante a SQL para o processamento de eventos.

O Código 2.1 exemplifica uma consulta ou regra CEP em que se procura por eventos do fluxo de dados de entrada **WindIn**, para os quais a velocidade do vento muda em mais de 5 milhas por hora em um segundo; os eventos que correspondem ao padrão procurado são inseridos em um fluxo de dados de saída **WindPatternOut** [Oberoi 2007].

**Código 2.1:** Exemplo de regra CEP [Oberoi 2007].

```
1  INSERT INTO WindPatternOut (Location, Speed1, Speed2)
2  SELECT W1.Location, W1.WindSpeed, W2.WindSpeed FROM WindIn W1, WindIn
   W2
3  MATCHING [2 SECONDS: W1 && W2] ON W1.Location = W2.Location WHERE (W1.
   WindSpeed - W2.WindSpeed) >= 5;
```

A proposta desta pesquisa trata as regras especificadas dentro de um escore clínico como regras CEP, as quais são utilizadas pela aplicação móvel de monitoramento para acompanhar as observações de interesse que devem ser feitas pelos sensores corporais ou espalhados no ambiente. Estas observações são essenciais para o processo de automatização de escores clínicos no contexto deste trabalho.

### 2.4.1 Aplicabilidade da Tecnologia CEP

[Oberoi 2007] descreve algumas situações em que engenheiros de software necessitam considerar a inclusão de CEP na pilha de tecnologia das aplicações. Esses cenários são:

1. Apenas os dados “processados” são úteis: em aplicações nas quais dispositivos localizados nas extremidades do sistema, como sensores ou leitores RFID, se conectam à empresa, todas as amostras de dados brutos não são de igual interesse para o negócio. Eventualmente, os dados precisam ser limpos, validados e enriquecidos antes de serem úteis.
2. Os ciclos de desenvolvimento de software não conseguem acompanhar as alterações nos algoritmos para detecção de padrões: nos aplicativos comerciais, os padrões para detectar eventos de compra e venda podem ser competitivos por alguns meses ou até mesmo algumas semanas. Existem casos em que novos padrões são descobertos, implementados e implantados em um dia. Nesses casos, é necessário parametrizar e abstrair a camada de detecção de padrões do aplicativo. Ter o algoritmo de negociação incorporado no código do aplicativo não é uma boa prática de projeto.
3. O processamento de eventos tem que ser feito em tempo real: nem todo processamento de eventos requer um mecanismo CEP, a exemplo de aplicações

de *data warehouse* que podem analisar dados “off-line”, ou seja, primeiro os dados são persistidos e depois correlacionados. Essa metodologia é muito lenta para aplicativos do tipo bolsa de valores, onde os dados precisam ser analisados em busca de padrões de eventos em tempo real. Logo, os eventos majoritariamente precisam ser inferidos em tempo real e as aplicações não podem esperar que os dados brutos sejam persistidos e analisados.

4. O processamento de eventos tem que ser escalonado: os aplicativos corporativos que se localizam nas extremidades do sistema precisam lidar com um grande número de sensores enviando simultaneamente altas taxas de dados. Os aplicativos tradicionais não estão capacitados para lidar com essa alta taxa e volume de dados. Usando um mecanismo CEP projetado para desempenho, pode-se tomar medidas para gerenciar uma carga deste tipo.

## 2.5 ContextNet

O middleware *ContextNet* objetiva o provisionamento de serviços de contexto para aplicações colaborativas pervasivas em larga e grande escala, assim como o monitoramento e coordenação de entidades móveis, e compartilhamento de informações através de redes sociais. Estas entidades podem ser usuários de dispositivos móveis, veículos ou robôs móveis autônomos [Endler e Silva 2018]. O *ContextNet* é composto basicamente de duas partes: o SDDL e o M-Hub.

O SDDL trata-se do componente do *ContextNet* que executa na nuvem e é responsável pela comunicação entre nós móveis com o objetivo de coletar e distribuir informação entre eles. O M-Hub é o componente do *ContextNet* que habilita dispositivos móveis com sistema operacional Android a funcionarem como *gateways* para a Internet, conectando objetos móveis com interface de comunicação sem fio de curto alcance, como sensores e atuadores [Pinheiro et al. 2018]. Esta pesquisa utiliza o middleware *ContextNet* para compor o Ambiente de Execução do *MDD4ClinicalScores*.

### 2.5.1 Scalable Data Distribution Layer

O SDDL é um middleware de comunicação desenvolvido por [David et al. 2013] que conecta nós DDS estacionários a um núcleo de processamento para nós móveis com conexão sem fio baseada em IP. O *Data Distribution Service for Real-time Systems* (DDS) [OMG 2012] é um padrão OMG que oferece capacidades de comunicação de alto desempenho com distribuição de dados do tipo *Publish/Subscribe* (Pub/Sub), e é usado por muitas aplicações distribuídas no mundo real cujos requisitos ou objetivos são extremamente críticos. A arquitetura básica do SDDL é mostrada na Figura 2.4.

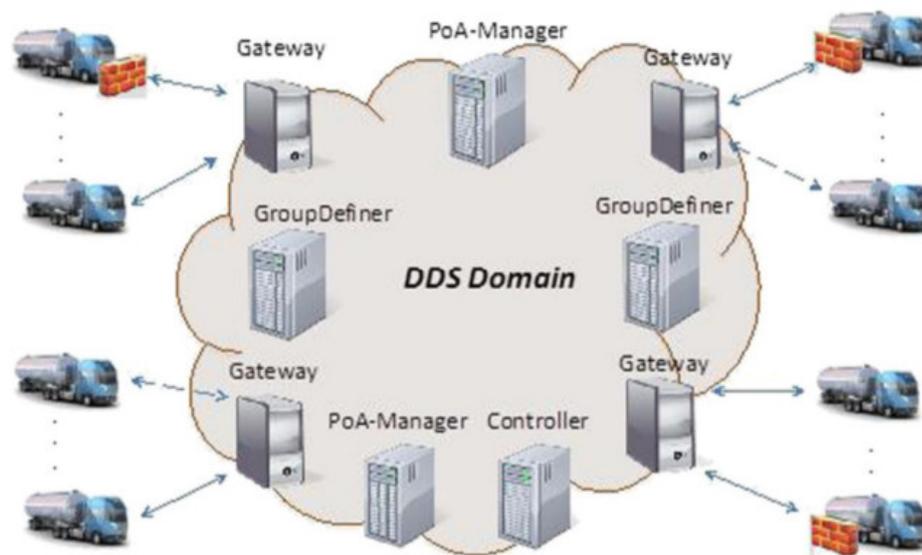


Figura 2.4: Arquitetura do SDDL [David et al. 2013].

De acordo com a Figura 2.4, os nós estacionários podem ser processadores de dados de contexto e informação, *gateways* para comunicação com nós móveis, ou controladores operados pelos gerenciadores do sistema. O *Controller* é responsável por exibir as posições atuais de todos os nós móveis conectados ao núcleo SDDL ou qualquer outra informação de contexto, além de gerenciar grupos de nós e enviar mensagens entre os nós móveis. O *Gateway* (GW) define um ponto único – em inglês, *Point of Attachment* (PoA) – para conexão com os nós móveis. Cabe ao *PoA Manager* distribuir periodicamente uma lista de PoAs para os nós móveis e eventualmente requisitar que alguns deles troquem de GW/PoA. Os *GroupDefiners* são responsáveis por avaliar membros de grupo para todos os nós móveis, o que pode ser determinado pela sua identificação, posição atual ou qualquer outro atributo de sua informação de contexto (por exemplo, nível de bateria).

A arquitetura do SDDL goza das vantagens da poderosa abordagem DDS centrada em dados, proporcionada pela sua arquitetura *Peer-to-Peer* (P2P) totalmente descentralizada, escalável, otimizada e rica em suporte de *Quality of Service* (QoS).

### 2.5.2 Mobile Hub

O M-Hub foi implementado por [Rios et al. 2015] como uma extensão natural do middleware SDDL, funcionando como um intermediário entre os objetos móveis (sensores, atuadores, *smart objects*, qualquer dispositivo com interface de comunicação sem fio) e a Internet. Dentre suas principais funcionalidades, o M-Hub oferece:

1. **Descoberta, monitoramento e registro de objetos móveis nas proximidades:** através da varredura periódica do ambiente em busca de objetos móveis, com o armazenamento da sua identificação e capacidades.
2. **Conexão com objetos móveis:** mediante o estabelecimento de uma comunicação sem fio estável com os objetos móveis.
3. **Tradução de protocolos:** transcodificação das mensagens trocadas entre os objetos móveis.
4. **Armazenamento de informações recentes sobre os objetos móveis:** com o objetivo de otimizar a comunicação dentro da rede móvel.
5. **Envio de requisições para objetos móveis:** envio periódico de requisições para os recursos ou serviços ofertados pelos objetos móveis.
6. **Gerenciamento do mecanismo de handover de objetos móveis de/para outros M-Hubs:** com o objetivo de manter a comunicação com os objetos móveis durante sua movimentação pelo ambiente.

A Figura 2.5 ilustra como se dá a comunicação entre os componentes centrais do M-Hub e dois objetos móveis. A arquitetura do M-Hub consiste de três serviços locais e dois mecanismos de gerenciamento. *Location Service* é o serviço responsável por mostrar a posição corrente do M-Hub e anexá-la a qualquer mensagem trocada entre ele e o GW. O GW é o componente do SDDL responsável por manter

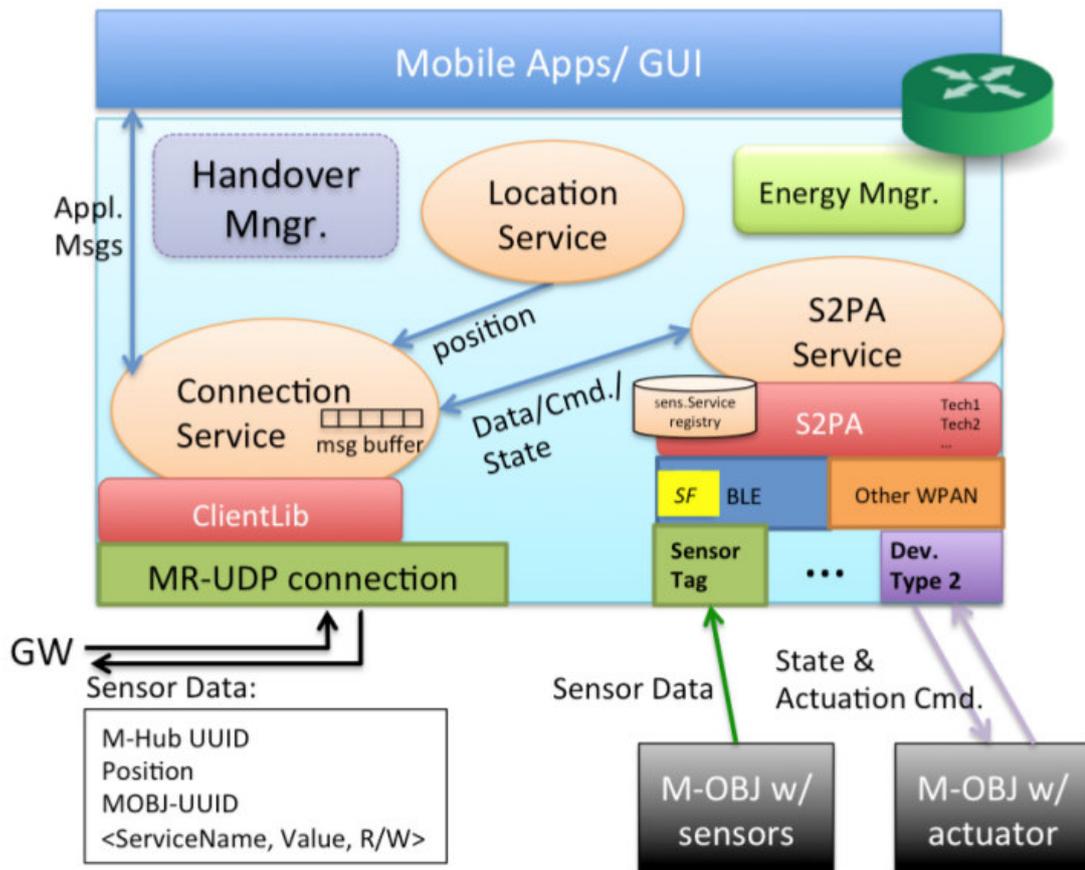


Figura 2.5: Componentes do M-Hub [Rios et al. 2015].

uma conexão individual com cada nó móvel dentro da nuvem. *S2PA Service* usa a API *Short-Range Sensor, Presence and Actuation* (S2PA) para interagir com os objetos móveis próximos. S2PA é um protocolo de comunicação de curto alcance projetado para a conexão entre o M-Hub e os objetos móveis. *Connection Service* é responsável pela troca de mensagens dentro da nuvem. *Energy Manager* gerencia periodicamente o nível de bateria dos dispositivos conectados. *Handover Manager* consulta o estado da conectividade com os objetos móveis, afim de promover a troca de M-Hubs dentro do ambiente e manter a comunicação com o SDDL.

Essas funcionalidades proporcionam ao M-Hub poder de processamento, memória suficiente, e interfaces de rede para tecnologias móveis e sem fio, para manipular as informações provenientes de objetos móveis do ambiente.

## 2.6 Web Semântica e Ontologias

Historicamente, o termo **ontologia** é uma ideia que nasceu na Filosofia com o objetivo de estudar principalmente dois aspectos: a natureza da existência, e o ramo de metafísicas relacionadas a como identificar os tipos de coisas que atualmente existem e como descrevê-las [Saha 2007]. Na Ciência da Computação, ontologia geralmente está associada ao conceito de **Web Semântica**.

Web Semântica trata-se basicamente da Web de segunda geração. [Saha 2007] explica que essa tecnologia possibilita a coleção automática e correlação de diferentes partes de informação sobre um objeto, que estão disponíveis em diferentes recursos Web. Para o autor, a Web Semântica economiza tempo valioso na navegação de um recurso Web para outro quando se procura uma informação relevante sobre um objeto particular. Desse modo, torna-se mais fácil para máquinas processar e integrar as informações presentes na Web.

Além disso, há uma estrita relação entre Web Semântica e aplicações em IoT. Uma questão de pesquisa amplamente discutida nesse contexto é a heterogeneidade de dados, já que é imperativo lidar com uma grande quantidade de sensores provenientes de diferentes fornecedores no mercado, e que produzem um alto volume de dados. Abordagens semânticas são usadas aqui com a finalidade de resolver essas questões de heterogeneidade em grande escala [Pinheiro et al. 2018].

Nesse cenário, as ontologias auxiliam na busca de uma representação genérica para um domínio de conhecimento, que seja acordada de forma consensual entre as aplicações participantes deste negócio. Com essas ferramentas, os objetos do mundo real podem ser representados e agrupados em conjuntos com características similares [Pinheiro et al. 2018]. A seguir será discutida brevemente a representação da OWL para ontologias na Web.

### 2.6.1 Web Ontology Language

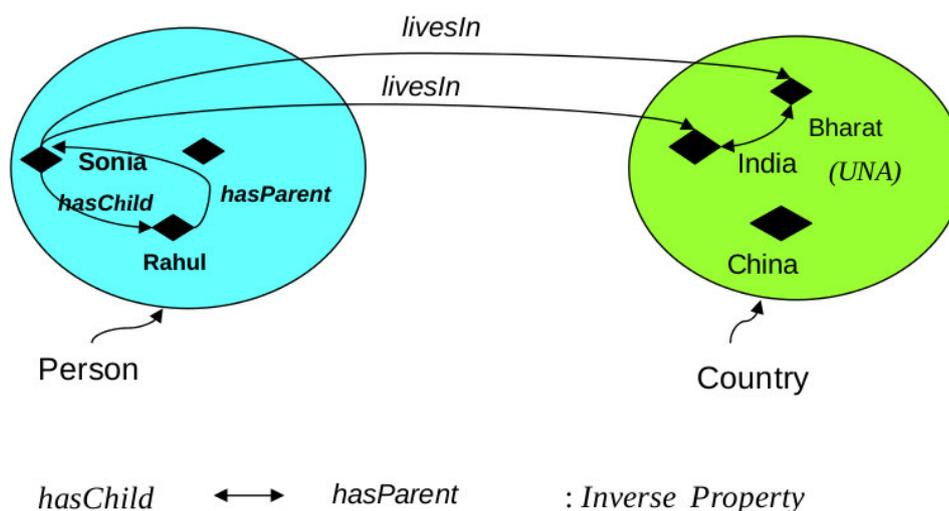
Linguagens de ontologias fornecem aos usuários uma maneira de escrever conceitos explícitos e formais para modelos de domínio. Para isso, essas linguagens precisam atender alguns requisitos, tais como: (1) deve ter uma sintaxe e semântica bem definidas; e (2) deve ter suporte ao raciocínio eficiente e forte

expressividade. O item n° 1 é bastante intuitivo e são tópicos que constituem requisitos essenciais para qualquer linguagem, não somente àquelas processadas por máquinas. O suporte ao raciocínio deriva da semântica e pode ser provida manual ou automaticamente, permitindo, dentre outros fatores, verificar a consistência da ontologia, o relacionamento entre classes, e a integração de ontologias provenientes de várias fontes. O requisito da expressividade decorre da habilidade da linguagem em representar todo o conhecimento ontológico existente [Antoniou e Harmelen 2004].

Com base nessas premissas, o *World Wide Web Consortium* (W3C) definiu a OWL e sua família de linguagens. A ontologia OWL foi concebida para combater a expressividade limitada do *Resource Description Framework Schema* (RDFS), disponibilizando três espécies de linguagem com diferentes expressividades [Saha 2007]:

1. **OWL Lite**: é a menos expressiva e ideal para hierarquia de classes simples, assim como restrições igualmente simples.
2. **OWL Description Logics (DL)**: é mais expressiva que a OWL Lite, e, como o próprio nome suscita, foi definida em cima de lógica descritiva; ela possui completude computacional, ou seja, todas as computações são finalizadas em um tempo finito.
3. **OWL Full**: é a mais expressiva das ontologias OWL, sendo sintaticamente livre do RDFS; apesar de não fornecer garantias computacionais nem raciocínio automático, ela permite que uma ontologia aumente o significado de um vocabulário predefinido (RDF ou OWL).

Com relação aos componentes da linguagem, uma ontologia OWL permite definir os seguintes: indivíduos, instâncias ou objetos; relacionamentos ou propriedades; classes ou conceitos. A Figura 2.6 traz um melhor entendimento sobre estas terminologias. Na Figura 2.6, tanto **Sonia** quanto **Rahul** são indivíduos da Classe **Person**, e **India** e **Bharat** são instâncias idênticas da Classe **Country**; Sonia tem um filho (propriedade **hasChild**) chamado Rahul, e Rahul tem uma mãe chamada Sonia (propriedade **hasParent**); e os relacionamentos **hasChild** e **hasParent** são inversos um com o outro [Saha 2007].



**Figura 2.6:** Componentes básicos da OWL [Saha 2007].

A proposta desta pesquisa engloba o uso da linguagem definida por [Pinheiro et al. 2018] baseada em OWL para descrever os indivíduos que são monitorados no contexto dos escores clínicos e os sensores que são utilizados neste monitoramento.

## 2.7 Considerações Finais

Este capítulo resumiu os aspectos teóricos mais relevantes que serviram de base para a elaboração desta pesquisa. Inicialmente, apresentamos os conceitos sobre a problemática de escores clínicos, bem como sua estrutura e metodologia de desenvolvimento, cujo conteúdo constitui o principal objeto de investigação deste trabalho.

Em seguida, este capítulo discutiu tópicos referentes à abordagem do MDD, incluindo as características das DSLs e transformações de modelo. A sistemática empregada pelo MDD compõe a proposta desta pesquisa para a resolução de escores clínicos no contexto da computação ubíqua.

Depois, a temática da mHealth e suas taxonomias foram abordadas com o intuito de contextualizar a aplicabilidade da nossa proposta. Logo após, este capítulo explicou os aspectos tecnológicos sobre CEP e os middlewares *ContextNet*, *SDDL* e *M-Hub*, cujos componentes integram o ambiente de execução da nossa abordagem.

---

Por fim, este capítulo desenvolveu um resumo teórico sobre Web Semântica e ontologias, destacando suas relações com a IoT. A forma de representação de ontologia OWL foi detalhada, pois consiste no tipo aplicado por esta pesquisa. Todos os conceitos aqui apresentados permeiam a solução desenvolvida pela nossa proposta.

### 3 Trabalhos Relacionados

Antes de iniciar o desenvolvimento da proposta desta pesquisa, foi feita uma investigação preliminar, nas bibliotecas digitais mais conhecidas da área da Ciência da Computação (Elsevier, IEEE, ACM, entre outros), de trabalhos científicos cujos temas abordassem a automatização de escores clínicos. Esta investigação envolveu a inclusão de fontes bibliográficas de outras áreas do conhecimento, principalmente da Medicina.

Na literatura, existem inúmeros trabalhos na área de mHealth para o monitoramento remoto de pacientes. Por exemplo, a própria OMG desenvolveu 12 (doze) padrões internacionais para sistemas computacionais da área da saúde, conforme descrito na URL <<https://www.omg.org/spec/category/healthcare/>>. Especificamente no âmbito da mHealth, [Olla e Tan 2009] trazem em sua coletânea uma série de 15 (quize) soluções para aplicações biomédicas. Estes autores dividiram os trabalhos conforme critérios aplicados em tecnologia, sistema de monitoramento e ciência de contexto. Por sua vez, [Istepanian et al. 2006] englobam em sua coletânea sistemas mHealth das mais diversas categorias. Dentre elas, destacam-se: sistemas inteligentes para profissionais da saúde; aplicações para compressão de sinal de áudio e vídeo; sistemas voltados para emergência médica; serviços ecográficos; e aplicações para monitoramento local e remoto [25]. Ademais, [Adams et al. 2017] descrevem trabalhos que focam em soluções mHealth para aquisição de biomarcadores comportamentais e psicológicos voltada para a pesquisa clínica psiquiátrica. Estas pesquisas abordam problemas clínicos que envolvem a esfera da psiquiatria, como o tabagismo, alcoolismo, drogas, estresse, ansiedade, autismo e transtornos do humor.

Recentemente, surgiram algumas iniciativas na área de mHealth com foco na automatização de escores clínicos. Todavia, o que todos esses trabalhos têm em comum é que abordam doenças ou comorbidades específicas. Este capítulo aborda soluções mHealth modernas da literatura que resolvem escores clínicos específicos. Aqui serão enfatizados os aplicativos ACTAPP, *Rotterdam Prostate Cancer Risk Calculator* (RPCRC), *cloudUPDRS* e *Bariatric Surgery Decision-Making Calculator* (BSDMC).

## 3.1 ACTAPP

A asma é uma das doenças respiratórias crônicas mais comuns, caracterizando-se pela dificuldade de respirar, chiado e aperto no peito, respiração curta e rápida. Os sintomas tendem a piorar à noite e nas primeiras horas da manhã ou em resposta à prática de exercícios físicos, à exposição a alérgenos, à poluição ambiental e a mudanças climáticas [Ministério da Saude 2019]. Na literatura, existem modelos de tratamento e gerenciamento de decisão focados especificamente nessa doença, como o *Asthma Control Test* (ACT). Estas ferramentas auxiliam o paciente com asma a ter um melhor controle sobre os sintomas da doença e melhorar sua qualidade de vida.

Diante disso, [Cook et al. 2016] desenvolveram e avaliaram um aplicativo móvel para melhorar o controle da asma por meio de ações pró-ativas e sem a necessidade de insumos regulares. Este aplicativo foi criado com base nas diretrizes definidas pelo ACT.

### 3.1.1 Teste de Controle de Asma

O teste ACT é uma marca comercial originalmente concebida pela *QualityMetric Incorporated*<sup>1</sup> e fornece uma visão geral de quão a asma do paciente tem sido controlada nas últimas quatro semanas. A pontuação deste teste varia entre 5 e 25 pontos. Ele é composto por cinco questionamentos com cinco opções de resposta cada, e cada resposta possui uma pontuação correspondente. Salvo pequenas variações dependendo da localidade, o teste básico ACT segue o modelo descrito na Tabela 3.1 (versão em português).

### 3.1.2 Aplicativo ACTAPP

O aplicativo ACTAPP foi concebido por [Cook et al. 2016] usando a plataforma *URXmobile System*. Este aplicativo interage com os usuários através de alertas e ações iniciadas pelo paciente, como consultas de informações e entrada de dados. Os alertas incluem pedidos de autoavaliação do controle da asma, avaliação do

---

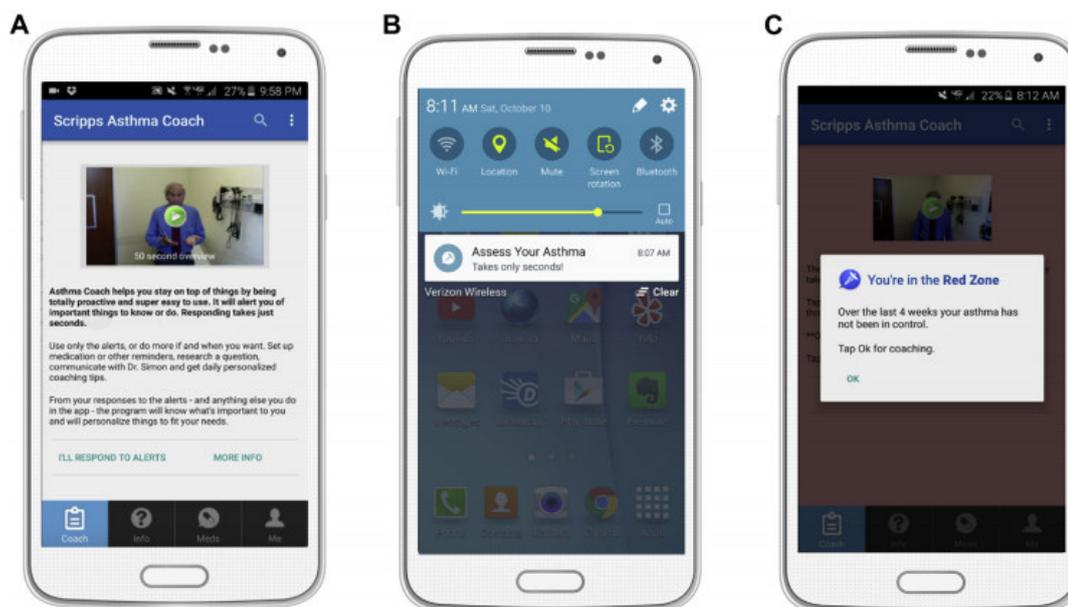
<sup>1</sup><[www.qualitymetric.com](http://www.qualitymetric.com)>

**Tabela 3.1:** Asthma Control Test versão em português [Roxo et al. 2016].

Questão	Resposta	Pontuação
<i>Nas últimas 4 semanas, a asma prejudicou suas atividades no trabalho, na escola ou em casa?</i>	Todo o tempo	1
	Maioria das vezes	2
	Algumas vezes	3
	Poucas vezes	4
	Nenhuma vez	5
<i>Nas últimas 4 semanas, como está o controle da sua asma?</i>	Totalmente descontrolada	1
	Pobrementemente controlada	2
	Um pouco controlada	3
	Bem controlada	4
	Completamente controlada	5
<i>Nas últimas 4 semanas, quantas vezes você teve falta de ar?</i>	Mais de 1 vez por dia	1
	1 vez por dia	2
	3 ou 6 vezes por semana	3
	1 ou 2 vezes por semana	4
	De jeito nenhum	5
<i>Nas últimas 4 semanas, a asma acordou você à noite ou mais cedo que de costume?</i>	4 ou + noites por semana	1
	2 ou 3 noites por semana	2
	1 vez por semana	3
	1 ou 2 vezes	4
	De jeito nenhum	5
<i>Nas últimas 4 semanas, quantas vezes você usou o remédio por inalação para alívio?</i>	3 ou + vezes por dia	1
	1 ou 2 vezes por dia	2
	Poucas vezes por semana	3
	1 vez ou menos por semana	4
	De jeito nenhum	5
<b>Avaliação</b>		<b>Resultado</b>
Quanto maior o escore, mais controlada é a asma		5-25

conhecimento do paciente sobre o autocuidado da asma e o conseqüente treinamento individualizado com base em suas anotações anteriores. A autoavaliação do controle da asma utiliza o teste ACT. Os autores definiram e empregaram uma classificação para avaliar o controle da asma do paciente de acordo com o resultado obtido após aplicação do ACT, conforme a seguir: **zona verde** (ACT > 19) indica controle de asma adequado; **zona amarela** (ACT = [16-19]) indica controle inadequado; e **zona vermelha** (ACT < 16) indica controle ruim. A Figura 3.1 demonstra funcionalidades do ACTAPP.

O ACTAPP contém uma coleção extensa de materiais educacionais baseada em evidências, que estão disponíveis nos formatos de texto e vídeo. Este material é facilmente customizável, para que seja futuramente melhorado com o acréscimo



**Figura 3.1:** Aplicativo ACTAPP para avaliar o controle da asma. **A**, vídeos embutidos e mensagens de texto proporcionam uma experiência amigável para o usuário. **B**, alertas pró-ativos engajam o usuário no cuidado da asma. **C**, avaliação do controle da asma pelo ACT [13].

de outros tratamentos e modelos de asma. Um total de 48 vídeos exclusivos estão disponíveis para os pacientes, com conteúdo incluindo discussão de alertas, importância da adesão à medicação, o papel de diferentes medicações, técnica e solução de problemas do inalador e tratamento de sintomas agudos.

Para avaliar a eficácia do ACTAPP, os autores submeteram o uso do aplicativo a um total de 60 pacientes, com média de idade de 50 anos. O uso controlado do aplicativo durou quatro meses. Com o uso do ACTAPP, o ranking geral desses pacientes quanto ao controle da asma melhorou significativamente (média do ACT subiu de 16.6 para 20.5), minimizando a necessidade de insumos pesados e provendo suporte de ensino e tratamento individualizado.

## 3.2 RPCRC

Câncer de próstata é uma doença grave que responde por 14% de novos casos de câncer e 6% do total de mortes de homens por câncer no mundo [Jemal et al. 2011]. Na tentativa de melhorar os procedimentos de diagnóstico desta doença, foi publicado recentemente o Estudo Europeu Randomizado de Diagnóstico do Câncer de Próstata [Schröder et al. 2014]. Atualmente, a decisão de realizar

uma biópsia da próstata baseia-se principalmente no resultado do teste de antígeno específico da próstata, ou *Prostate-Specific Antigen* (PSA) em inglês. Todavia, os resultados do PSA podem aumentar em várias situações, incluindo condições benignas e inflamatórias. Assim, recomenda-se o uso do PSA combinado com o uso de calculadoras de risco, que evidenciam outras características relevantes do paciente [Mottet et al. 2015]. Nesse contexto, calculadoras de risco cientificamente válidas para o risco de câncer de próstata são os escores RPCRC [Roobol et al. 2012]. Os escores RPCRC usam informações sobre o nível de PSA, biópsia de próstata negativa anterior, resultado do exame de toque retal, medição do volume da próstata e ultrassonografia transretal [Pereira-Azevedo et al. 2017].

Nessa linha, [Pereira-Azevedo et al. 2017] projetaram e desenvolveram um aplicativo móvel para estratificar o risco de câncer de próstata baseado nos escores RPCRC. O processo de desenvolvimento envolveu análise, projeto, implementação e avaliação.

### 3.2.1 Escores RPCRC

Os escores RPCRC<sup>2</sup> foram desenvolvidos por [Roobol et al. 2012], fornecendo um total de oito especificações relacionadas ao rastreamento do câncer de próstata. Os dois primeiros escores foram projetados para uso individual sem qualquer conhecimento especializado. Os demais escores são mais indicados para uso por especialistas da saúde, pois apresentam modelos de pontuação mais complexos. Os escores RPCRC são:

1. O 1º RPCRC é conhecido como *Escore de Saúde Geral* e se trata de um ponto inicial de investigação, procurando pelo histórico familiar, idade e alguns problemas envolvendo a urina do paciente.
2. O 2º RPCRC é conhecido como *Escore de Risco com PSA* e examina os níveis do PSA no sangue do paciente para prever se uma investigação adicional será necessária.
3. Os 3º e 4º RPCRC são conhecidos como *Escore de Risco com Ultrassonografia Transretal ou Exame de Toque Retal* e são combinados em uma árvore de decisão

---

<sup>2</sup><<http://www.prostatecancer-riskcalculator.com/seven-prostate-cancer-risk-calculators>>

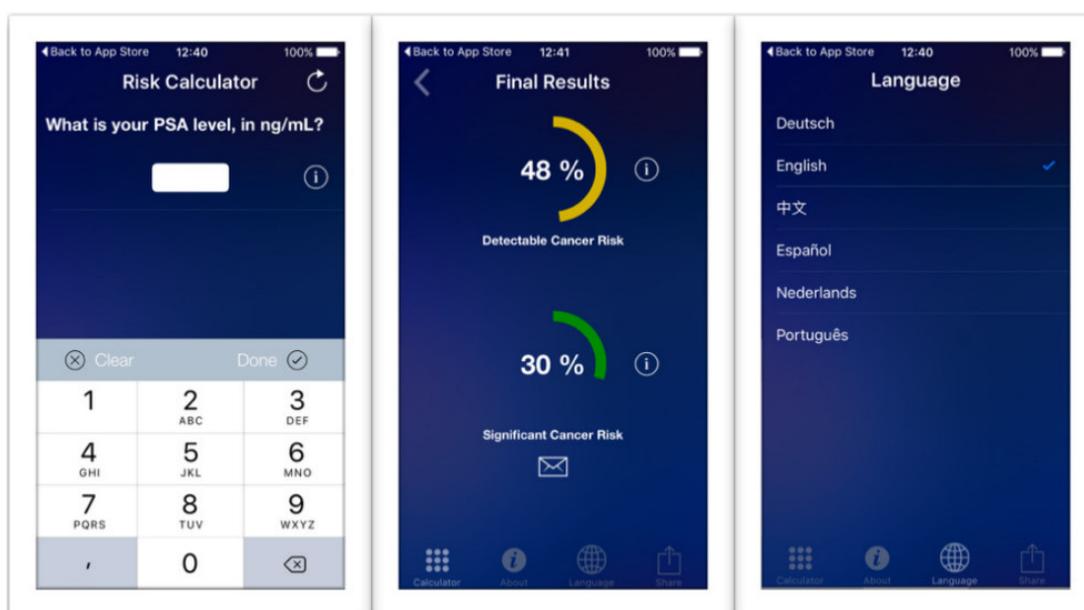
para uso em homens que não foram previamente biopsiados ou que foram examinados anteriormente e tiveram uma biópsia de próstata com resultado benigno. O volume da próstata (25, 40 ou 60 cc) pode ser incluído de duas formas diferentes - ou durante a ultrassonografia transretal ou estimado com base no exame de toque retal. A inclusão do volume da próstata no cálculo do risco fornece uma estimativa mais precisa do risco de ter um câncer de próstata detectável por biópsia (potencialmente agressivo).

4. Os 5º e 6º RPCRC são conhecidos como *Escore de Risco com Ultrassonografia Transretal ou Exame de Toque Retal e Resultado do Teste Phi* e oferecem a possibilidade de incluir o resultado do exame de sangue do *Prostate Health Index* (Phi) que aumenta a capacidade preditiva dos escores RPCRC.
5. O 7º RPCRC calcula a chance de ter câncer de próstata indolor que não requer tratamento imediato.
6. O 8º RPCRC é o mais recente da coleção e calcula o risco futuro de um homem nos próximos quatro anos, levando em consideração a idade, o PSA, o exame de toque retal, o histórico familiar, o volume da próstata e o status da biópsia anterior.

### 3.2.2 Projeto do Aplicativo RPCRC

Na etapa de análise do aplicativo RPCRC, [Pereira-Azevedo et al. 2017] estabeleceram requisitos funcionais e de conhecimento. Os requisitos de conhecimento usados no aplicativo foram desenvolvidos com base no Estudo de [Schröder et al. 2014] descrito na Seção 3.2, com o uso de dados clínicos e resultados de biópsia de próstata de 3.624 homens não rastreados previamente e 2.896 homens com biópsia de próstata negativa anterior. A partir daí, os seguintes modelos foram construídos, com informações clínicas cumulativas: (1) PSA sozinho; (2) PSA e exame de toque retal padrão; (3) PSA, exame de toque retal padrão e de volume da próstata; (4) e RPCRCs descritos na Subseção 3.2.1. Para melhorar a usabilidade do aplicativo, foi criada uma árvore de decisão única, com uma abordagem de várias etapas, para reunir informações clínicas disponíveis: biópsia de próstata negativa anterior, valor do PSA, avaliação do exame de toque retal e avaliação da ultrassonografia transretal.

A interface de usuário do aplicativo RPCRC foi desenvolvida com base nas diretrizes de projeto da Apple. Fatores como clareza e facilidade de uso guiaram o projeto da interface do aplicativo, que usou a ferramenta *GNU Image Manipulation Program*<sup>3</sup> (GIMP). O aplicativo foi implementado para o sistema operacional iOS usando o Apple's Xcode. A Figura 3.2 apresenta algumas telas presentes no aplicativo RPCRC, que evidenciam suas funcionalidades e interface. O projeto pode ser dividido dentro de 6 interfaces: **aviso legal**, **questionamento**, **explicação**, **linguagem**, **resultados**, e **sobre o aplicativo**. O aviso legal deve ser aceito pelo usuário antes de começar a utilizar o aplicativo. Um total de 11 questionamentos foram inseridos dentro do aplicativo e, dependendo das respostas, um dos quatro modelos de pontuação pode ser aplicado para prever o risco (significativo ou não) de câncer de próstata. Todos os questionamentos apresentam a mesma interface e, para cada questionamento, existe uma tela com sua respectiva explicação. Os resultados são exibidos pelo aplicativo nas formas gráfica e numérica (porcentagem). O usuário tem a opção de escolher dentre seis línguas estrangeiras: Chinês, Holandês, Inglês, Alemão, Português e Espanhol. Por fim, a tela sobre o aplicativo detalha as contribuições científicas do sistema.



**Figura 3.2:** Aplicativo móvel RPCRC para estimativa de risco de câncer de próstata [Pereira-Azevedo et al. 2017].

Tanto os requisitos funcionais quanto de usabilidade do aplicativo RPCRC foram avaliados pelos autores. A comprovação dos requisitos funcionais do aplicativo advém do seu projeto, que foi construído em cima de informações de saúde de alta

<sup>3</sup><https://www.gimp.org/downloads/>

qualidade, extraídas de diversos trabalhos da literatura que validam o resultado das calculadoras de risco de Roterdã em múltiplos estudos. A usabilidade do aplicativo foi calculada quantitativamente por meio do questionário *IBM Computer Usability Satisfaction Questionnaire*, cuja ferramenta oferece mensurações fortes de usabilidade. Como exemplo, na aplicação deste questionário para cinco usuários, o resultado obtido foi uma cobertura de 85% de todas as questões de usabilidade do aplicativo RPCRC.

### 3.3 cloudUPDRS

A *Doença de Parkinson* (DP) é uma condição neurológica degenerativa associada a um conjunto de sintomas motores e cognitivos, incluindo tremor, lentidão de movimento e congelamento, rigidez muscular, instabilidade postural, dificuldades relacionadas ao sono, depressão e psicose [Jankovic 2008]. O atendimento a pacientes com DP envolve o manejo de sintomas motores e não motores, bem como cuidados paliativos. Como não há cura, o manejo destes sintomas é um processo que afeta a vida toda, englobando tratamento farmacológico correspondente ao estágio da doença, fisioterapia e cirurgia [NIHCE 2006]. Além disso, os sintomas da DP progridem independentemente do tratamento e de forma individual, o que exige um acompanhamento clínico particular. Tais fatores indicam um método terapêutico laborioso e dispendioso, pois requer o envolvimento direto de um profissional de saúde especializado [Stamate et al. 2018].

Para enfrentar esses desafios, [Stamate et al. 2018] desenvolveram o aplicativo **cloudUPDRS** para avaliação clínica dos sintomas motores da DP. Segundo os autores, o *cloudUPDRS* é o primeiro aplicativo para smartphone certificado como dispositivo médico Classe I pela Agência Reguladora de Medicamentos e Produtos de Saúde no Reino Unido. Este aplicativo é baseado na escala *Unified Parkinson's Disease Rating Scale* (UPDRS) desenvolvida pela *Movement Disorder Society* (MDS) [Goetz et al. 2008] e no questionário *Parkinson's Disease Questionnaire* (PDQ39) [Jenkinson et al. 1997]. Os pacientes usam o aplicativo em casa para registrar as medidas de sensores comuns do *smartphone* (por exemplo, acelerômetro e GPS) enquanto executam uma série de ações simples. Os dados capturados pelo *smartphone* são então usados para calcular a pontuação clínica da escala UPDRS.

### 3.3.1 Escala UPDRS e Questionário PDQ39

A escala UPDRS foi originalmente desenvolvida na década de 80 e posteriormente revisada em 2008 pela MDS. Trata-se da escala de avaliação clínica para DP mais utilizada no mundo. A revisão proposta pela MDS manteve os pontos fortes da escala original, mas incorporou uma série de problemas clínicos da doença que eram mal capturados na versão original. Cada pergunta dessa escala é ponderada por 5 respostas que estão relacionadas a termos clínicos comumente aceitos: **0 = normal**, **1 = leve**, **2 = suave**, **3 = moderado**, e **4 = grave**. A escala MDS-UPDRS avalia um total de 65 itens associados a DP e é composta da seguinte forma [Goetz et al. 2008]:

- Parte 1: refere-se às experiências não motoras da vida diária, com 13 pontuações (por exemplo, problemas com depressão e ansiedade);
- Parte 2: refere-se às experiências motoras da vida diária, com 13 pontuações (por exemplo, atividades no lazer e higiene pessoal);
- Parte 3: refere-se ao exame dos sintomas motores, com 33 pontuações (baseadas em 18 itens, a maioria com pontuações distribuídas pelo lado direito, esquerdo ou outro do corpo; por exemplo, movimento das mãos e agilidade das pernas);
- Parte 4: refere-se às complicações do sistema motor, com 6 pontuações (por exemplo, discinesia).

O questionário PDQ39 foi desenvolvido por [Jenkinson et al. 1997] para avaliar com que frequência as pessoas afetadas pela DP enfrentam dificuldades em 8 dimensões da sua vida diária. Estas dimensões incluem relacionamentos, situações sociais e comunicação. O PDQ39 é composto por 39 itens que ajudam a medir o estado de saúde e qualidade de vida do paciente com DP. Os itens são avaliados em uma escala da seguinte forma: **nunca**, **ocasionalmente**, **algumas vezes** e **sempre**. “Teve dificuldades de fazer atividades de lazer que gostaria?” ou “Teve dificuldades de carregar sacolas de compras?” são exemplos de perguntas que constam do PDQ39.

### 3.3.2 Sistema e Aplicativo cloudUPDRS

O aplicativo *cloudUPDRS* de [Stamate et al. 2018] implementa um fluxo de trabalho abrangente que fornece mídia de áudio, vídeo e texto (ver Figura 3.3) para

orientação de pacientes e seus cuidadores na realização dos testes em casa, sem a necessidade de supervisão de um profissional de saúde especializado. O sistema *cloudUPDRS* é composto de:

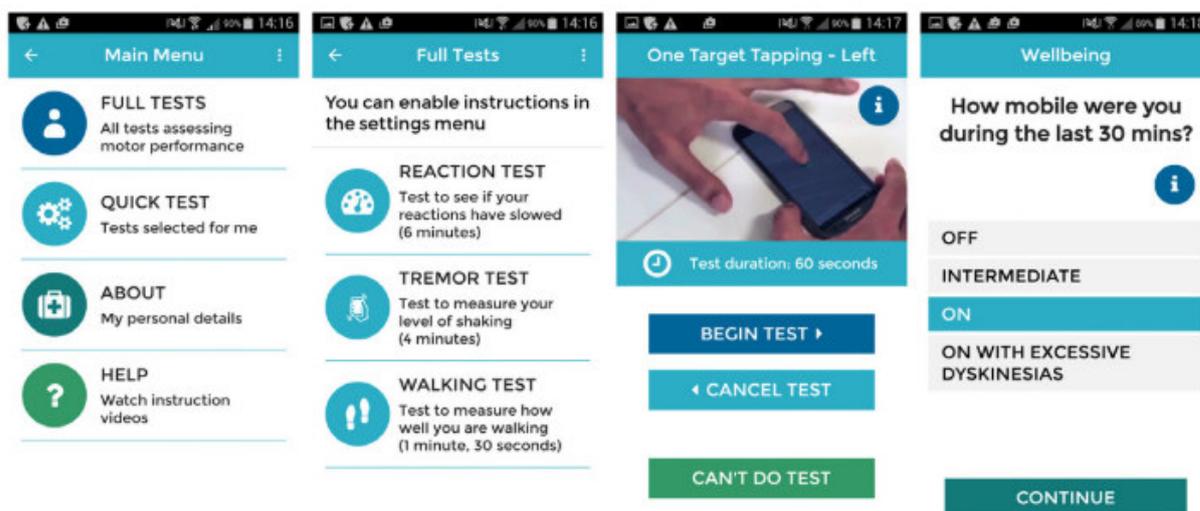


Figura 3.3: Visões da interface de usuário do aplicativo *cloudUPDRS* [Stamate et al. 2018].

- Um aplicativo Android que permite ao paciente realizar testes motores e concluir uma autoavaliação de bem-estar, além de gerenciar sessões e enviar dados para o serviço *cloudUPDRS* com segurança;
- Serviço escalável para coleta de dados baseado em nuvem que recebe dados dos smartphones de pacientes, gerencia dados de forma segura e aplica processamento de sinal;
- Kit de ferramentas para mineração de dados visando inteligência médica, incorporação de dados quantitativos semiestruturados, análises longitudinais, agrupamento e classificação, bem como uma interface de usuário que incorpora dados clínicos.

O *cloudUPDRS* guia os pacientes por uma sequência de ações cuidadosamente orquestrada, enquanto o aplicativo registra as medições dos sensores do *smartphone*. Ao exigir a execução de sequências de ação específicas, o aplicativo restringe o grau de liberdade do movimento individual e, portanto, impõe estrutura e isola o contexto do usuário, limitando o intervalo de comportamentos observados. Como resultado, o sinal gravado pode ser interpretado com precisão usando um pequeno conjunto de heurísticas. Por fim, o aplicativo se adapta automaticamente

para atender às especificações do dispositivo hospedeiro e incorpora um serviço de tolerância a atrasos para gerenciar o upload de dados.

O teste completo administrado pelo *cloudUPDRS* é formado por 17 observações, que são: tremor cinético, postural e em repouso para a mão direita e esquerda; agilidade da perna esquerda e direita e tremor em repouso; dedo alvo único e duplo tocando em ambos os lados; e o caminhar. Cada observação dura 60 segundos e o paciente deve assumir uma posição específica e depois realizar o movimento conforme descrito anteriormente. Após o registro destas observações, o aplicativo apresenta ao paciente questões do PDQ39 e registra o tempo da ingestão mais recente de medicamentos. Para o processamento de biossinais, o aplicativo implementa um serviço de análise de dados baseado em nuvem que avalia três características do paciente: **tremor, bradicinesia e caminhar**.

Para avaliar a progressão da DP, o *cloudUPDRS* combina dois métodos: o primeiro mecanismo fornece suporte para execução de ações com precisão por parte do usuário, além de orientá-lo durante todo o processo; o segundo mecanismo usa *Redes Neurais Convolucionais Recorrentes* [Schmidhuber 2015] para confirmar que as gravações enviadas foram capturadas durante a realização dos testes pelo paciente. A primeira abordagem carece de comprovação de que o cumprimento total das ações prescritas foi efetivamente alcançado, enquanto que a segunda permite que o aplicativo aprenda características de movimento associadas a um sinal de alta qualidade e alerte o usuário quando uma observação não foi capturada sob condições satisfatórias.

## 3.4 BSDMC

Nos últimos cinco anos, houve um tremendo crescimento no desenvolvimento de aplicativos móveis voltados para a área médica. A maioria desses aplicativos é voltada para o paciente (por exemplo, aplicativos de condicionamento físico), sendo poucos aqueles que foram desenvolvidos e efetivamente utilizados na prática clínica. O uso de um aplicativo mHealth é ideal no campo da cirurgia bariátrica para garantir que o paciente e o médico entendam o risco relativo e o benefício da cirurgia [Aminian et al. 2017b].

Nesse contexto, [Aminian et al. 2017b] desenvolveram um aplicativo para iPhone, denominado de BSDMC móvel, que oferece acesso rápido e conveniente a calculadoras baseadas em evidências para tomada de decisões com relação a procedimentos cirúrgicos em nível bariátrico.

### 3.4.1 Escores SGRC, RPVTBS e IMSS

O escore *Sleeve Gastrectomy Risk Calculator* (SGRC) foi desenvolvido por [Aminian et al. 2015] para prever o risco de mortalidade pós-operatória precoce após uma gastrectomia vertical laparoscópica. Os fatores de risco deste escore são: **histórico de insuficiência cardíaca congestiva (I)**, **uso de esteróides para condições crônicas (II)**, **sexo masculino (III)**, **diabetes (IV)**, **nível de bilirrubina total no soro pré-operatório (V)**, **Índice de Massa Corporal (IMC) (VI)** e **nível de hematócrito pré-operatório (VII)**. A probabilidade estimada para ocorrer algum evento adverso para um dado paciente é calculada através das fórmulas 3.1 e 3.2:

$$L = -3.497 + (0.487 * IV) + (0.0307 * VI) + (0.517 * III) + (1.83 * I) + (1.61 * II) + (0.451 * V) + (0.054 * VII) \quad (3.1)$$

$$P(E) = e^x(L)/(1 + e^x(L)) \quad (3.2)$$

, onde  $e^x=2.718$  é a base do logaritmo natural.

O escore *Risk of Post-discharge Venous Thromboembolism after Bariatric Surgery* (RPVTBS) foi criado por [Aminian et al. 2017a] para determinar os fatores de risco para tromboembolismo venoso pós-alta de 30 dias após cirurgia bariátrica e identificar possíveis indicações para farmacoprofilaxia prolongada. Os fatores de risco deste escore são: **histórico de insuficiência cardíaca congestiva (I)**, **paraplegia (II)**, **retorno para a sala de cirurgia (III)**, **dispneia em repouso (IV)**, **cirurgia de banda não-gástrica (V)**, **idade  $\geq 60$  anos (VI)**, **sexo masculino (VII)**, **IMC  $\geq 50$  kg/m<sup>2</sup> (VIII)**, **duração da estadia  $\geq 3$  dias (IX)** e **tempo operativo  $\geq 3$ h (X)**. A probabilidade estimada para ocorrer algum evento adverso para um dado paciente é calculada através das fórmulas

3.3 e 3.4:

$$L = -7.337 + (I * 1.883) + (II * 1.743) + (III * 1.631) + (IV * 1.375) \\ + (V * 0.893) + (VI * 0.67) + (VII * 0.655) + (VIII * 0.516) + (IX * 0.455) \\ + (X * 0.451) \quad (3.3)$$

$$P(E) = e^x(L)/(1 + e^x(L)) \quad (3.4)$$

, onde  $e^x=2.718$  é a base do logaritmo natural.

O escore *Individualized Metabolic Surgery Score* (IMSS) foi concebido por [Aminian et al. 2017c] para validar um sistema de pontuação para seleção de procedimentos cirúrgicos bariátricos e metabólicos baseada em evidências conforme a gravidade da diabetes tipo 2. Os fatores de risco deste escore são: **número de medicamentos para diabetes pré-operatório, uso de insulina, duração da diabetes tipo 2 em anos (pré-operatório) e controle glicêmico**. A Figura 3.4 demonstra o modelo de pontuação de acordo com o valor de cada variável. A avaliação do IMSS segue a seguinte classificação:

1.  $\leq 25$ : paciente é classificado como diabetes tipo 2 suave;
2.  $= [26-95]$ : paciente é classificado como diabetes tipo 2 moderada;
3.  $> 95$ : paciente é classificado como diabetes tipo 2 severa.

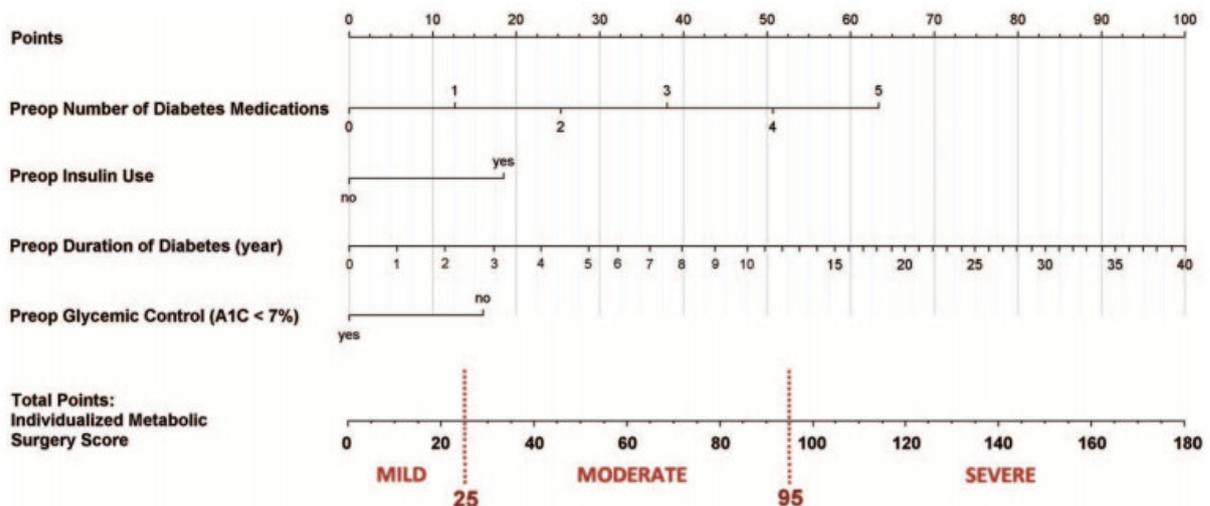


Figura 3.4: Modelo de pontuação do escore IMSS [Aminian et al. 2017c].

### 3.4.2 Aplicativo BSDMC

O aplicativo BSDMC foi desenvolvido por [Aminian et al. 2017b] com base nos escores clínicos descritos na Subseção 3.4.1. Estes escores respondem importantes questões clínicas no campo da cirurgia bariátrica. A Figura 3.5 ilustra as telas do aplicativo que estão associados a cada escore anteriormente mencionado.

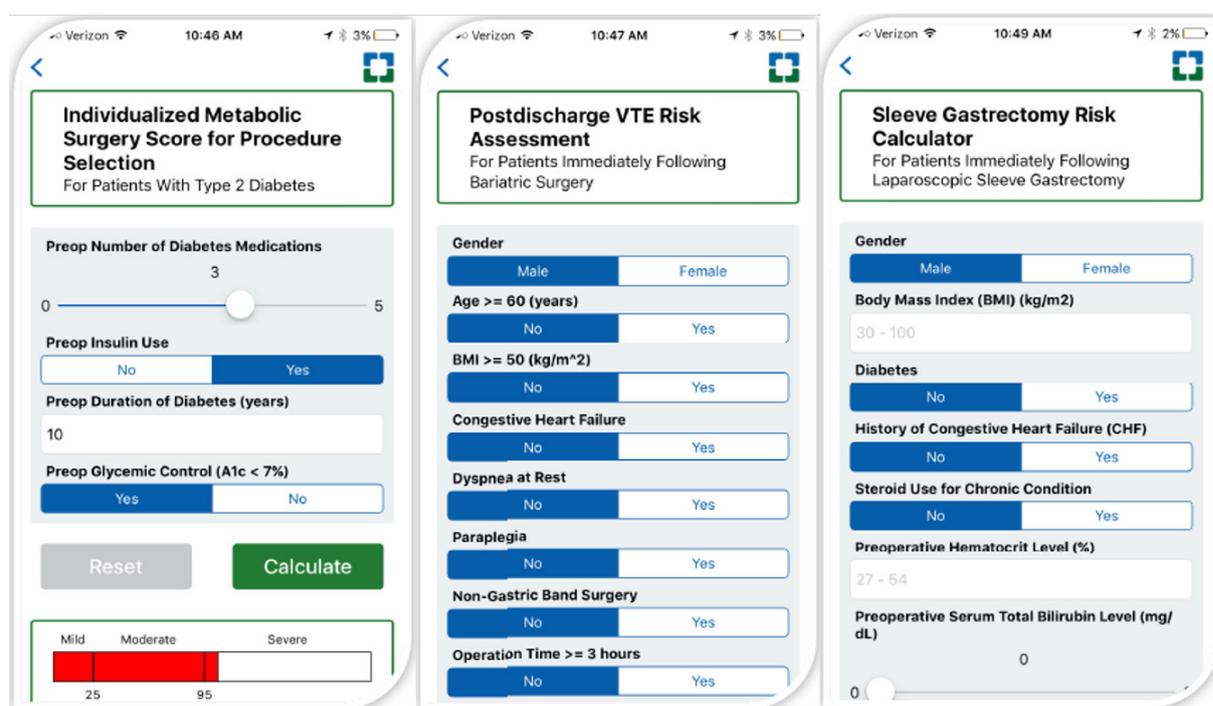


Figura 3.5: Telas do aplicativo BSDMC [Aminian et al. 2017b].

A funcionalidade associada ao IMSS classifica os pacientes com diabetes em três estágios de gravidade validados para a seleção de procedimentos metabólicos baseados em evidências. Já a funcionalidade do RPVTBS utiliza 10 fatores de risco independentes e identifica pacientes de alto risco que poderiam se beneficiar da tromboprolifaxia prolongada por tromboembolismo venoso após a alta. Por sua vez, a funcionalidade do SGRC, que estima especificamente eventos adversos graves no pós-operatório de 30 dias após gastrectomia vertical laparoscópica com base em 7 variáveis independentes, pode contribuir para a tomada de decisão cirúrgica e consentimento informado para os pacientes.

O aplicativo BSDMC trata-se de um sistema promissor para estratificação de risco e seleção de procedimentos cirúrgicos e metabólicos baseado em evidências. No futuro, os autores desejam expandir a ferramenta com a incorporação de outros modelos válidos para auxiliar profissionais na prática clínica.

## 3.5 Considerações Finais

No presente capítulo foram exploradas as características das mais recentes abordagens mHealth para automatização de escores clínicos apresentadas na literatura. Foi possível observar que todas as soluções aqui discutidas tem uma característica comum: estão focadas no desenvolvimento de um sistema computacional para tratar um escore clínico específico. Isso representa uma limitação na forma como o problema é resolvido computacionalmente hoje.

Dentre as soluções computacionais em mHealth que abordam escores clínicos, este capítulo discutiu as seguintes: (1) ACTAPP, um aplicativo móvel baseado no teste ACT para auxiliar pacientes com asma no controle da doença; (2) RPCRC, um aplicativo móvel baseado nos escores RPCRC para estratificar o risco de câncer de próstata nos homens; (3) *cloudUPDRS*, um sistema baseado em nuvem que utiliza a escala UPDRS para avaliação clínica de pacientes com Doença de Parkinson; e (4) BSDMC, uma solução móvel baseada em três sistemas de pontuação clínica para tomada de decisão sobre cirurgia bariátrica.

O *MDD4ClinicalScores*, diferentemente das abordagens citadas anteriormente, permite tratar um conjunto maior de escores clínicos e gerar componentes de software de forma automática, por meio do uso de técnicas de metamodelagem e transformação de modelos. Os componentes de software gerados possibilitam o monitoramento remoto em tempo real de pacientes através de escores clínicos previamente especificados.

## 4 MDD4ClinicalScores

O presente capítulo apresenta uma nova abordagem para desenvolvimento de software focada na automatização de escores clínicos que foi desenvolvida nesta pesquisa, o *MDD4ClinicalScores*. São descritos os principais componentes dessa abordagem, destacando a linguagem *DSML4ClinicalScore*. No que tange ao escopo dessa linguagem de especificação, são explicadas as motivações e os requisitos para seu desenvolvimento, bem como seus conceitos fundamentais.

Depois, este capítulo discute o processo de geração automática de código por meio de transformação de modelos, o que viabiliza a produção de componentes de software a partir de um modelo especificado conforme a *DSML4ClinicalScore*. Para isso, a transformação M2T foi a técnica escolhida para traduzir os conceitos entre a linguagem proposta e a *Deklaer*, uma linguagem baseada em ontologias preconizada no trabalho de [Pinheiro et al. 2018].

Para finalizar, este capítulo examina a arquitetura do ambiente de execução que foi concebido para gerar semiautomaticamente uma aplicação móvel para IoT, utilizando os componentes de software produzidos pela abordagem proposta.

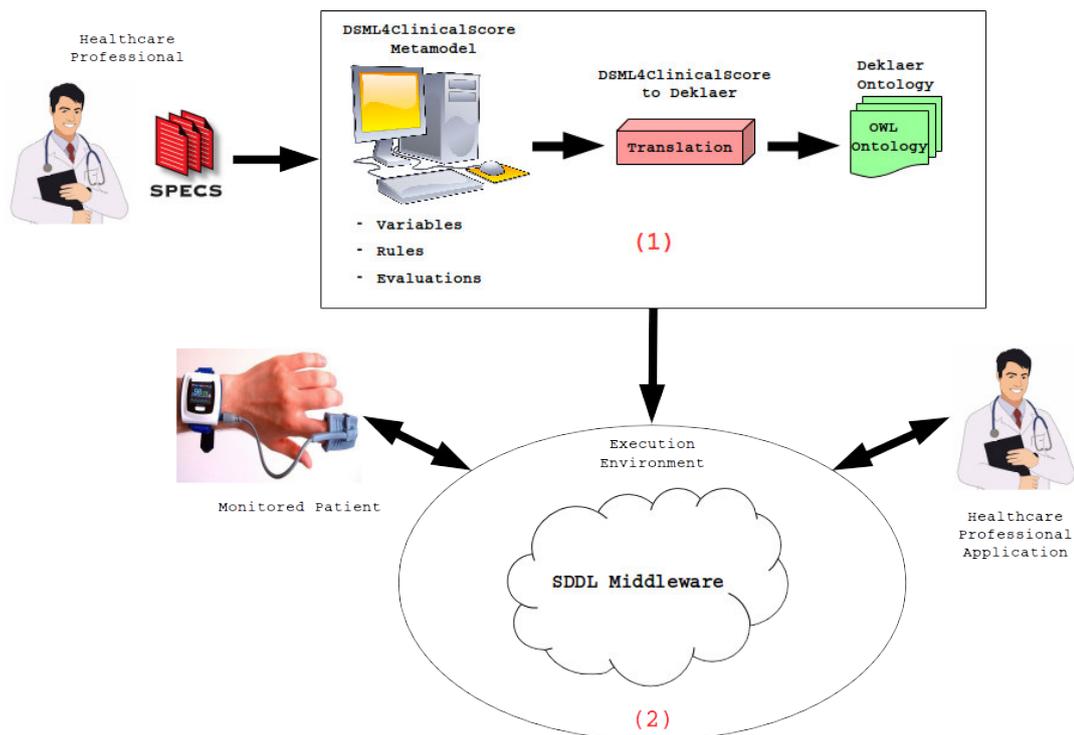
### 4.1 Visão Geral do MDD4ClinicalScores

O *MDD4ClinicalScores* é uma abordagem para desenvolvimento de software que dá suporte à modelagem e execução de escores clínicos em ambientes de computação ubíqua. O principal objetivo dessa abordagem é auxiliar desenvolvedores de aplicações mHealth e demais pesquisadores no processo de modelagem e resolução de escores clínicos usando computação ubíqua.

As motivações para o desenvolvimento do *MDD4ClinicalScores* surgiram no atual cenário de soluções computacionais limitadas para resolução de escores clínicos dentro do contexto da computação ubíqua (ver Capítulo 1). O objetivo principal desta pesquisa era responder os seguintes questionamentos: (1) é viável a criação de uma solução que cubra várias especialidades de escore clínico?; e (2) é possível

responder questões de escores clínicos de forma automática ou semiautomática através de sensores e/ou dados eletrônicos? Logo, os conceitos metodológicos aplicados pelo MDD compuseram a abordagem escolhida para resolver tais questionamentos.

O *MDD4ClinicalScores* é uma abordagem baseada na *DSML4ClinicalScore* (ver Figura 4.1), uma linguagem de modelagem de domínio específico voltada para a especificação de escores clínicos em geral. A *DSML4ClinicalScore* oferece suporte à modelagem dos componentes existentes em um escore clínico padrão: variáveis, regras para cálculo da pontuação e avaliações. O *MDD4ClinicalScores* implementa regras de transformação que, a partir da especificação (modelo concreto) criada pela *DSML4ClinicalScore*, geram classes de uma ontologia em OWL. Por sua vez, essa ontologia resultante é submetida a um framework orientado por ontologias, que cria e executa semiautomaticamente uma aplicação mHealth para monitoramento remoto de pacientes, com foco na avaliação do escore clínico previamente especificado. Os componentes mostrados na Figura 4.1 são:



**Figura 4.1:** Arquitetura do *MDD4ClinicalScores*.

- 1. Geração Automática de Código:** representada pela parte número 1 na Figura 4.1, envolve a modelagem da especificação de escore clínico por meio do metamodelo da *DSML4ClinicalScore* e a tradução do modelo concreto criado para uma

ontologia OWL através de uma transformação M2T; essa ontologia é descrita conforme a linguagem *Deklaer* [Pinheiro et al. 2018].

2. **Ambiente de Execução:** representado pela parte número 2 na Figura 4.1, provê o suporte para execução da aplicação móvel para IoT e é centrado no middleware SDDL, responsável por configurar o sistema com os componentes de software produzidos na etapa da geração automática de código, que descrevem a aplicação mHealth para monitoramento remoto do paciente com base no escore clínico previamente especificado, assim como por processar as regras CEP correspondentes às observações de interesse presentes nessa aplicação.
3. **Aplicação Móvel de Monitoramento:** conecta-se aos sensores biomédicos presentes no ambiente, coleta os dados mensurados durante o monitoramento do paciente, e analisa quais regras predefinidas do escore clínico foram verificadas, associando um valor específico a cada uma delas.
4. **Aplicação de Gerenciamento:** aplicação desktop ou móvel que recebe as notificações de monitoramento do paciente e efetua o cálculo da pontuação do escore clínico, apresentando a respectiva avaliação de acordo com o resultado obtido.

## 4.2 A Linguagem DSML4ClinicalScore

A principal contribuição do *MDD4ClinicalScores* é prover uma linguagem de domínio específico para modelar diferentes especificações de escore clínico, a *DSML4ClinicalScore*. A linguagem *DSML4ClinicalScore* foi projetada para incorporar todos os aspectos pertinentes a um escore clínico padrão, o que envolve a definição das variáveis, das regras para o cálculo da pontuação e dos modelos de avaliação.

Esta seção aborda o metamodelo proposto pela *DSML4ClinicalScore*, discutindo a metodologia usada para seu desenvolvimento e os conceitos por ela empregada.

### 4.2.1 Análise do Domínio

Para propor um metamodelo que permita definir uma DSML para modelar diferentes especificações de escore clínico, majoritariamente é necessário identificar quais conceitos devem ser incluídos como artefatos nesse metamodelo. Consequentemente, o primeiro passo é determinar os dados e requisitos funcionais que estão presentes em um escore clínico. Para isso, é imperativo definir o espaço amostral do qual esses conceitos serão extraídos.

Por conseguinte, para alcançar o objetivo acima mencionado, adotou-se o trabalho de [Aakre et al. 2017] como uma referência inicial, levando em consideração sua relevância e contemporaneidade para com a proposta deste trabalho. Neste artigo, os autores avaliaram empiricamente 110 escores clínicos e os submeteram à avaliação de especialistas médicos dos Estados Unidos da América. Essa avaliação foi embasada na opinião subjetiva desses profissionais em relação à importância de ter as pontuações desses escores clínicos inseridas automaticamente nos prontuários eletrônicos dos pacientes. Os critérios escolhidos para qualificar determinado escore clínico foram: **MUITO IMPORTANTE**, **BOM TER** ou **NÃO É IMPORTANTE** [1].

Desse modo, para facilitar a interpretação do domínio sobre escores clínicos, elaborou-se um Mapa Conceitual (ver Figura 4.2) que evidenciasse os principais conceitos de uma especificação de escore clínico que deveriam constar do metamodelo da *DSML4ClinicalScore*. Este mapa conceitual foi concebido a partir de 89 dos 110 escores clínicos previamente analisados, uma vez que foram considerados somente aquelas especificações avaliadas como **MUITO IMPORTANTE** ou **BOM TER** no trabalho de [Aakre et al. 2017]. Esses modelos foram analisados sob dois pontos de vista: (1) classificá-los com relação a como as variáveis da especificação são obtidas computacionalmente; e (2) extrair os conceitos relevantes sobre escores clínicos de diferentes especialidades médicas.

O Mapa Conceitual da Figura 4.2 mostra todos os conceitos identificados durante a análise do domínio sobre escores clínicos. Assim, os conceitos encontrados foram:

- **Variables:** este conceito identifica e representa os preditores clínicos que influenciam a pontuação final do escore clínico; englobam todas as variáveis

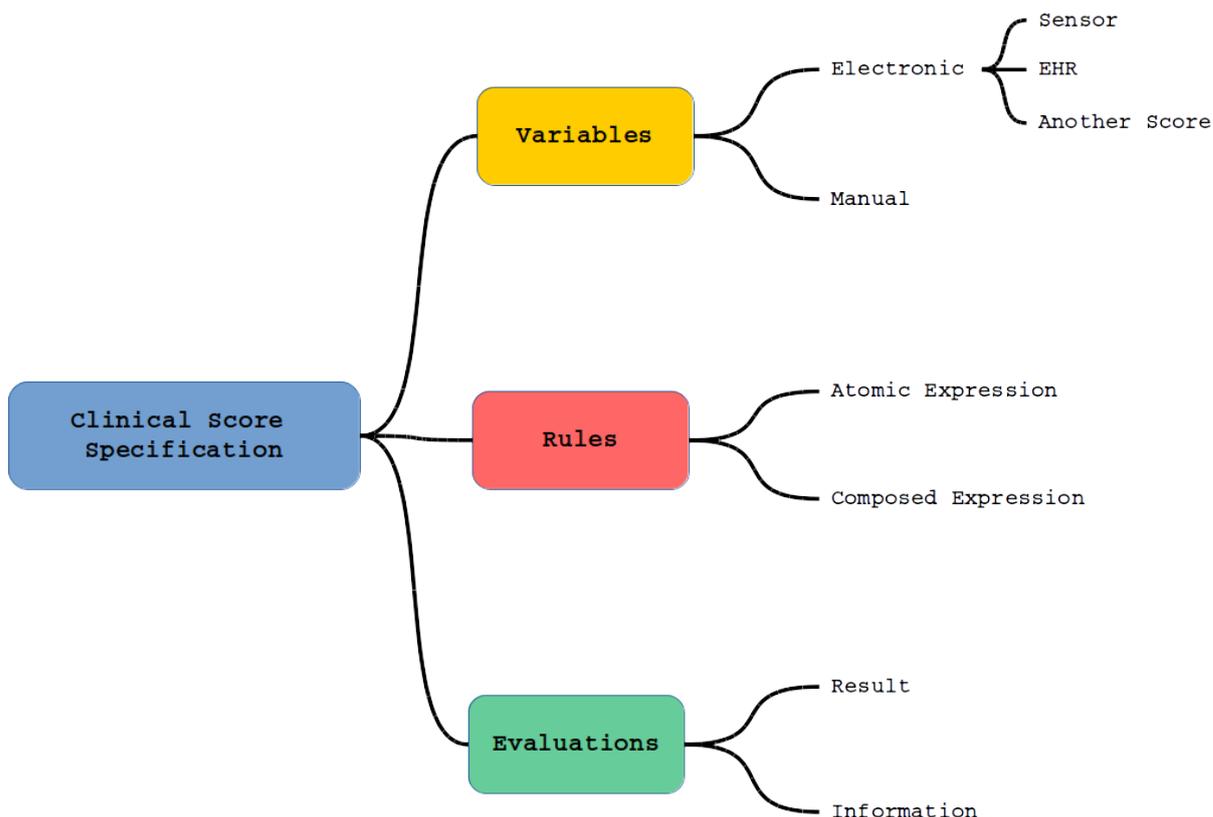


Figura 4.2: Mapa Conceitual para uma especificação de escore clínico.

manuais ou eletrônicas possíveis que podem ser obtidas computacionalmente e classificar um preditor clínico.

- **Rules:** este conceito identifica e representa as regras necessárias para o cálculo da pontuação de cada preditor clínico previamente definido na especificação; incluem todas as expressões atômicas ou compostas possíveis que podem caracterizar uma regra específica.
- **Evaluations:** este conceito identifica e representa as avaliações necessárias para a interpretação do escore clínico; abrangem um resultado final e todas as informações que podem ser descritas para definir uma avaliação específica.

### 4.2.2 Projeto da Linguagem

O metamodelo da *DSML4ClinicalScore* foi implementado com base no metamodelo Ecore [Eclipse Community 2019] com auxílio do IDE do EMF<sup>1</sup>, um ambiente de desenvolvimento integrado desenvolvido pela Comunidade do Eclipse.

<sup>1</sup><<https://www.eclipse.org/sirius/overview.html>>



classe objetiva identificar de forma única e exclusiva uma especificação de escore clínico, além de prever sua finalidade de uso. Uma classe do tipo `Specification` é composta por uma única classe do tipo `VariableSection`, uma única classe do tipo `RuleSection`, e uma ou mais classes do tipo `EvaluationSection`.

A classe `VariableSection` da Figura 4.3 define todas as variáveis presentes em uma especificação de escore clínico. Essas variáveis podem ser obtidas computacionalmente por meio de sensores, informações do prontuário eletrônico do paciente, entrevistas com profissionais de saúde ou pacientes, e informações de outros escores clínicos. Uma classe do tipo `VariableSection` é composta por uma ou mais classes do tipo `Variable`, e está associada a uma única classe do tipo `RuleSection`.

A classe `Variable` da Figura 4.3 assina um identificador, uma unidade de medida, e um ou mais valores de entrada para cada variável criada dentro da especificação de escore clínico, podendo ser do tipo `Electronic` ou `Manual`. A classe `Electronic` engloba as variáveis da especificação obtidas eletronicamente com seu respectivo tipo primitivo de dado, podendo ser do tipo `Sensor`, `EHR` ou `AnotherScore`. A classe `Manual` inclui as variáveis da especificação obtidas manualmente com seu respectivo tipo primitivo de dado. A classe `Sensor` identifica qual tipo de dado de sensor está alimentando a especificação, conforme descrito na classe `Sensors` (`BloodPressure`, no caso de Pressão Sanguínea; `HeartRate`, no caso de Batimento Cardíaco; `Oxygenation`, no caso de Oxigenação Sanguínea; `RespiratoryRate`, no caso de Frequência Respiratória; `Temperature`, no caso de Temperatura; ou `Other`, para dar suporte à expansão de conceitos no futuro). A classe `EHR` identifica qual tipo de informação do prontuário eletrônico do paciente está alimentando a especificação, conforme descrito na classe `EHRInformation` (`Age`, no caso de Idade; `Event`, no caso de Eventos diversos; `Exam`, no caso de Resultados de Exames diversos; `Gender`, no caso de Gênero; `PersonalMeasures`, no caso de Medidas Pessoais; ou `Other`, para dar suporte à expansão de conceitos no futuro). A classe `AnotherScore` engloba resultados de escores clínicos que funcionam como variável de entrada para outra especificação.

A classe `RuleSection` da Figura 4.3 estabelece todas as regras para o cálculo da pontuação do escore clínico, associadas a cada variável previamente definida na classe `VariableSection`. Essas regras podem ser representadas por qualquer expressão lógica corretamente descrita. Uma classe do tipo `RuleSection`

é composta por uma ou mais classes do tipo `Rule`, e está associada a uma única classe do tipo `VariableSection` e a uma ou mais classes do tipo `EvaluationSection`.

A classe `Rule` da Figura 4.3 descreve como as regras para cálculo da pontuação do escore clínico são definidas, atribuindo a cada uma delas um identificador, uma pontuação e seu respectivo tipo primitivo de dado, e sendo composta por uma única classe do tipo `Expression`. A classe `Expression` cria uma expressão atômica ou composta para uma regra específica dentro da especificação, podendo ser do tipo `Atomic` ou `Composed`. A classe `Atomic` cria uma expressão atômica dentro da classe `Rule`, podendo ser do tipo `RefVariable` ou `Constant`. A classe `RefVariable` representa uma expressão atômica associada a uma única variável do tipo `Variable` e previamente definida na classe `VariableSection`. A classe `Constant` representa uma expressão atômica associada a uma constante qualquer, seja numérica, textual ou booleana, e assume obrigatoriamente um valor de entrada e seu respectivo tipo primitivo de dado.

Por sua vez, a classe `Composed` da Figura 4.3 cria uma expressão composta dentro da classe `Rule`, podendo ser do tipo `LogicalOperator`, `RelationalOperator` ou `MathematicalOperator`. A classe `LogicalOperator` identifica qual operador lógico foi usado em uma expressão composta, conforme descrito na classe `LogicalOperators` (`And`, no caso de E; `Or`, no caso de Ou; `Not`, no caso de Não; `NotAnd`, no caso de Não E; `NotOr`, no caso de Não Ou; `OrExclusive`, no caso de Ou Exclusivo; ou `NotOrExclusive`, no caso de Não Ou Exclusivo), sendo composta por até duas classes do tipo `Atomic`, ou duas classes do tipo `RelationalOperator`, ou duas classes do tipo `LogicalOperator`. A classe `RelationalOperator` identifica qual operador relacional foi usado em uma expressão composta, conforme descrito na classe `RelationalOperators` (`EqualTo`, no caso de Igual A; `DifferentFrom`, no caso de Diferente De; `GreaterThan`, no caso de Maior Que; `LessThan`, no caso de Menor Que; `GreaterOrEqualThan`, no caso de Maior Ou Igual Que; ou `LessOrEqualThan`, no caso de Menor Ou Igual Que), sendo composta por até duas classes do tipo `Atomic` ou duas classes do tipo `MathematicalOperator`. A classe `MathematicalOperator` identifica qual operador matemático foi usado em uma expressão composta, conforme descrito na classe `MathematicalOperators` (`Addition`, no caso de Adição; `Subtraction`, no caso de Subtração; `Multiplication`, no caso de Multiplicação; `Division`, no caso

de Divisão; *Exponentiation*, no caso Exponenciação; *SquareRoot*, no caso de Raiz Quadrada; *Factorial*, no caso de Fatorial; *Logarithm*, no caso de Logaritmo; ou *Other*, para dar suporte à expansão de conceitos no futuro), sendo composta por até duas classes do tipo *Atomic* ou mais classes do tipo *MathematicalOperator*.

A classe *EvaluationSection* da Figura 4.3 descreve todos os modelos possíveis de avaliação para uma especificação de escore clínico, identificando cada modelo criado. Uma classe do tipo *EvaluationSection* é composta por uma ou mais classes do tipo *Evaluation* e uma única classe do tipo *Result*, e está associada a uma única classe do tipo *RuleSection*. A classe *Evaluation* denota uma avaliação do escore clínico de acordo com o respectivo intervalo de valores alcançado para seu resultado, assinando um identificador, uma notificação e os valores mínimo e máximo do intervalo da pontuação, e pode estar associada ou não a uma classe do tipo *Result*. A classe *Result* obtém o resultado final da especificação com seu respectivo tipo primitivo de dado, e está associada a uma única classe do tipo *Evaluation*.

Por último, o metamodelo da *DSML4ClinicalScore* estabelece a classe de enumeração *PrimitiveTypes* da Figura 4.3 para descrever quais os tipos primitivos de dado que podem ser utilizados pelas suas classes. A classe *PrimitiveTypes* define quatro tipos básicos: *Int*, no caso de Inteiro; *Double*, no caso de Ponto Flutuante; *String*, no caso de Texto; e *Boolean*, no caso de Booleano.

### 4.3 Geração Automática por Transformação de Modelos

O procedimento de geração automática de código definido pelo *MDD4ClinicalScores* baseia-se em uma técnica de transformação M2T. Esta técnica permite que um produto no formato textual seja concebido a partir de um modelo concreto descrito em *Ecore*, por exemplo. O *MDD4ClinicalScores* implementa regras de transformação que mapeiam os conceitos do metamodelo proposto pela *DSML4ClinicalScore* para uma ontologia OWL baseada na linguagem *Deklaer*, projetada para descrever aplicações IoT que interagem com sensores do ambiente.

Esta seção aborda inicialmente os conceitos definidos pela linguagem *Deklaer*, seguindo com a explicação sobre o mapeamento existente entre as linguagens da *DSML4ClinicalScore* e *Deklaer*.

### 4.3.1 A Linguagem Deklaer

A linguagem de modelagem declarativa *Deklaer* foi proposta por [Pinheiro et al. 2018] para descrever uma aplicação IoT com presença de sensores e atuadores no ambiente. Esta linguagem é definida por meio de um conjunto de ontologias, visando proporcionar a ela capacidades de raciocínio no futuro. Para isso, os autores empregaram séries de ontologias que posteriormente são alinhadas em uma ontologia final. Tal característica busca obter modularidade e capacidade de particionamento do conhecimento, para que o raciocínio seja executado apenas sobre parte deste conhecimento, e não sobre toda a base, técnica esta conhecida como **ontologias contextualizadas** [Cafezeiro et al. 2014]. Essa técnica faz uso de lógica descritiva que, junto com um mecanismo de raciocínio, minimiza os erros que porventura possam ser inseridos pelo desenvolvedor durante a criação da aplicação.

As ontologias que criam a linguagem *Deklaer* são *ObservationActuation*, *Environment*, *DomainSpecific* e *Entities*. *ObservationActuation* é a ontologia principal onde concentra toda a lógica da aplicação, descrevendo sensores e atuadores que devem ser observados e executados, respectivamente. *Environment* descreve as características dos sensores localizados no ambiente, assim como que tipo(s) de valor(es) eles fornecem e com que frequência. *DomainSpecific* contém uma classe representativa para cada dado produzido pelos sensores definidos na ontologia *Environment*. Enfim, *Entities* possui informações específicas (nome, descrição e identificador) sobre cada entidade que deve ser observada pela aplicação, podendo ser pessoas ou objetos do mundo real.

Todavia, toda a complexidade acima mencionada é escondida do desenvolvedor da aplicação pela linguagem *Deklaer*. Para o desenvolvedor, é necessário descrever sua aplicação unicamente com o vocabulário provido pela ontologia final, devido ao alinhamento de todas as ontologias previamente definidas. Sendo assim, o vocabulário dessa linguagem é composto pelas seguintes classes:

- *Entity*: representa instâncias de diferentes entidades do sistema, cada qual com sua própria identificação; neste caso, entidades podem ser pessoas, animais ou objetos físicos.

- `ActRule`: representa instâncias de uma classe que relata uma atuação no ambiente; no momento, a única atuação implementada pelo *Deklaer* é o envio de uma mensagem de texto para uma ou mais entidades.
- `ActuationEffect`: agrupa diferentes formas de atuação.
- `Notification`: é subclasse de `ActuationEffect` e descreve uma mensagem de texto e para quais entidades ela deve ser enviada.
- `Actuator`: representa instâncias de dispositivos que podem produzir alguma ação no mundo real.
- `Sensor`: representa instâncias de sensores do mundo real, podendo ter um ou mais tipos de informação, isto é, se um dispositivo é capaz de fornecer dados de mais de um sensor. Cada sensor é representado por uma instância de `Sensor` dentro da *Deklaer*.
- `SensorObs`: especifica o que deve ser observado por um sensor específico, sendo aplicada uma condição lógica associada a um valor para mensuração. Por exemplo, com uma `SensorObs`, pode-se definir uma observação do tipo **“temperatura maior que 37°C”**.
- `ObsRule`: combina uma ou mais instâncias de `SensorObs` com uma instância de `ActRule`.
- `LogicalCondition`: representa tipos de condições lógicas que podem ser aplicadas em uma `SensorObs`. *Deklaer* define as seguintes condições lógicas: **Igual A; Maior Que; Menor Que; Maior Ou Igual Que; e Menor Ou Igual Que**.
- `ObsActRule`: associa um conjunto de `ObsRule` a um conjunto de `ActRule`, ou seja, se todos os `ObsRule` forem observados, um grupo de `ActRule` é executado.

No que se refere à implementação da linguagem *Deklaer*, [Pinheiro et al. 2018] usaram o *Protégé*<sup>2</sup> para editar e criar exemplos de aplicações para IoT. O *Protégé* é uma ferramenta de código aberto criada no âmbito da Universidade de Stanford (EUA) para manipulação de ontologias, com suporte a raciocínio lógico sobre modelos por ela criados. Assim, essa ferramenta possibilita a validação da aplicação

---

<sup>2</sup><<https://protege.stanford.edu/products.php>>

descrita pela *Deklaer*. Um exemplo de aplicação para IoT descrita por esta linguagem é mostrada na Figura 4.4. Este exemplo inclui três sensores, quatro atuadores e quatro observações que devem ser verificadas por estes sensores.

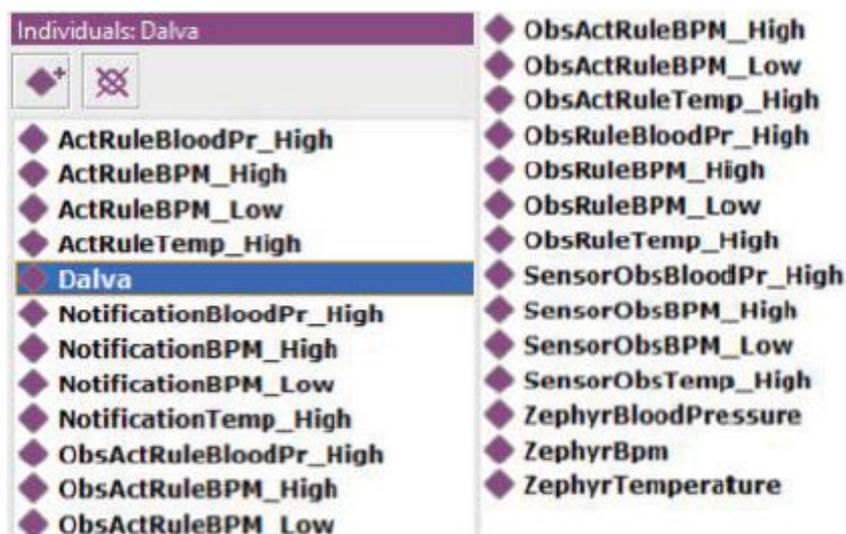


Figura 4.4: Exemplo de aplicação escrita conforme a linguagem *Deklaer* [Pinheiro et al. 2018].

### 4.3.2 Transformação M2T de DSML4ClinicalScore para Deklaer

Um aspecto fundamental definido pelo *MDD4ClinicalScores* é a possibilidade de gerar código automaticamente através da aplicação de uma transformação M2T. O processo de transformação recebe como entrada o modelo concreto criado pelo usuário, ou seja, a especificação gerada conforme o metamodelo da linguagem *DSML4ClinicalScore* e composta de todas as seções de um escore clínico específico. A partir deste modelo concreto, são geradas classes de uma ontologia OWL descrita conforme a linguagem *Deklaer*.

O processo de transformação é executado a partir de um conjunto de regras previamente definidas. Essas regras estão estruturadas na forma de *templates* e *queries*, com cada *template* responsável por um estágio da transformação e cada *query* usada para obter valores ou coleção de valores do modelo concreto especificado. A transformação M2T do *MDD4ClinicalScores* foi implementada usando a ferramenta *Acceleo*<sup>3</sup>. O *Acceleo* fornece uma forma fácil de manipular metamodelos descritos em uma linguagem de modelagem e gerar texto a partir desses metamodelos, além de estar disponível como um plug-in do EMF.

<sup>3</sup><<http://www.eclipse.org/acceleo/>>

Para entender o processo de transformação, primeiro é imperativo informar quais conceitos da *Deklaer* podem ser gerados a partir de quais conceitos da *DSML4ClinicalScore*. A Tabela 4.1 mostra o mapeamento entre os conceitos dessas duas linguagens. Pelas informações contidas na Tabela 4.1, conclui-se que este mapeamento gera somente as *data properties* da ontologia da *Deklaer*. Essa característica indica que as *object properties* da referida ontologia são geradas a partir da criação pelas regras de transformação dos indivíduos associados às classes. Ademais, isso também indica que nem todos os conceitos definidos pela *Deklaer* podem ser concebidos a partir do modelo concreto criado pela *DSML4ClinicalScore*. Consequentemente, algumas informações devem ser inseridas manualmente dentro da ontologia produzida pelo *MDD4ClinicalScores* para que seu ambiente de execução consiga criar corretamente uma aplicação mHealth para o escore clínico especificado. Isso é melhor detalhado no Capítulo 5, Seção 5.2.

**Tabela 4.1:** Mapeamento entre os conceitos da *DSML4ClinicalScore* e *Deklaer*.

Conceitos da <i>DSML4ClinicalScore</i>	Conceitos da <i>Deklaer</i>
Rule:id	ActRule:has_id ObsActRule:has_id ObsRule:has_id
Rule:score	Notification:has_notificationText
Rule:Expression	SensorObs:has_valueOneRestriction SensorObs:has_name SensorObs:isAbout SensorObs:has_numberOfValues SensorObs:has_logicalCondition SensorObs:has_unityOfMeasurement

Como pode ser visto na Tabela 4.1, as classes *ActRule*, *ObsRule* e *ObsActRule* da *Deklaer* requerem um identificador. Para facilitar a criação de relacionamentos entre estes conceitos durante o mapeamento, para cada objeto do tipo *Rule* presente no modelo concreto da *DSML4ClinicalScore*, o *MDD4ClinicalScores* utiliza seu identificador para criar cada um daqueles respectivos conceitos na ontologia final. Aquilo que deve ser monitorado do paciente ou as instâncias de *SensorObs* da *Deklaer* que devem constar da ontologia final, é considerado pelo *MDD4ClinicalScores* objetos do tipo *Expression* definidos no modelo concreto da *DSML4ClinicalScore*. Diferentemente da concepção original da linguagem *Deklaer*, na qual a classe *Notification* carrega uma informação referente ao que foi observado

por uma `SensorObs`, optou-se por modificar esta característica para que aquela classe carregue consigo a pontuação correspondente a sua respectiva regra/observação.

A transformação implementada pelo *MDD4ClinicalScores* começa pela instanciação da classe `Specification` do modelo concreto criado pela *DSML4ClinicalScore* (ver Subseção 4.2.2). A partir desta classe, todos os objetos especificados do tipo `Variable` e `Rule` dentro do modelo concreto são investigados pelas regras de transformação, com o objetivo de encontrar objetos especificados do tipo `Sensor` e suas respectivas regras. Em outras palavras, as rotinas desenvolvidas buscam por todas as regras especificadas que estão associadas a um sensor dentro do modelo concreto.

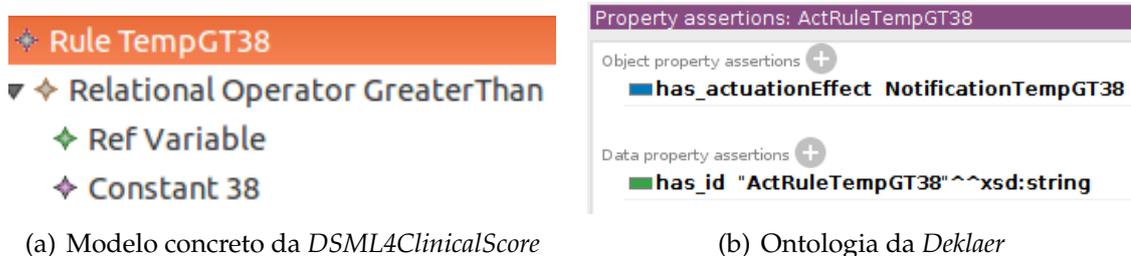
Como mencionado anteriormente, as regras de transformação consistem de *templates* e *queries*. Um *template* dentro do *Acceleo* gera o código em texto que será usado pela aplicação final. Dessa forma, para cada classe da *Deklaer* que pode ser criada a partir do modelo concreto da *DSML4ClinicalScore*, desenvolveu-se um *template* específico que gera os indivíduos dessa classe e suas respectivas *data properties* e *object properties*. Esses *templates* podem ser divididos em duas categorias básicas segundo sua finalidade: **genéricos** e **específicos**. Nos *templates* genéricos, tem-se o código comum para toda ontologia elaborada a partir da *DSML4ClinicalScore*, isto é, independente do modelo concreto especificado, as informações aqui produzidas estão presentes em todas as ontologias descritas pela *Deklaer*. Por sua vez, os *templates* específicos produzem o código específico relacionado ao modelo concreto fornecido, com as informações inerentes aos sensores encontrados e suas respectivas regras.

Com base nisso, os *templates* genéricos definidos pelo *MDD4ClinicalScores* são: `preamble`, `classes`, `subclasses`, `generateAnnotations` e `endOntology`. `preamble` insere as informações de preâmbulo de uma ontologia OWL. `classes` cria todas as classes definidas pela *Deklaer*. `subclasses` cria todas as subclasses definidas pela *Deklaer*. `generateAnnotations` insere informações adicionais que o desenvolvedor queira fornecer para a ontologia final, geralmente associadas a *data properties*. Por fim, `endOntology` finaliza a ontologia criada com as *tags* de encerramento.

Por outro lado, os *templates* específicos definidos pelo *MDD4ClinicalScores* são: `actrule`, `notification`, `obsactrule`, `obsrule`, `sensor` e `sensorobs`.

`actrule` gera os indivíduos, *data properties* e *object properties* da classe `ActRule` dentro da ontologia final descrita conforme a *Deklaer*, a partir dos objetos especificados no modelo concreto descrito conforme a *DSML4ClinicalScore*, que serviu de entrada para a transformação. O mesmo raciocínio é aplicado para os demais *templates*, cuja responsabilidade é a criação das informações associadas com as correspondentes classes da *Deklaer* (ver Subsecção 4.3.1), ou seja, `notification` para `Notification`, `obsactrule` para `ObsActRule`, e assim por diante.

Exemplificando, a partir da regra representada através do modelo concreto da *DSML4ClinicalScore* e mostrada na Figura 4.5(a), que descreve a regra de id: “TempGT38”, é gerado o indivíduo correspondente da classe `ActRule` (Figura 4.5(b)) com mesmo identificador (nome da classe + id da regra), descrito conforme *Deklaer*.



**Figura 4.5:** Transformação de um objeto `Rule` da *DSML4ClinicalScore* para um indivíduo `ActRule` da *Deklaer*.

Dentro das regras de transformação implementadas pelo *MDD4ClinicalScores*, uma *query* do *Acceleo* é responsável por extrair informação do modelo concreto da *DSML4ClinicalScore*, podendo retornar valores ou coleções de valores. Esta funcionalidade é geralmente usada no código dos *templates* para compor expressões condicionais e validá-las. Logo, as *queries* criadas pelo *MDD4ClinicalScores* são as seguintes:

- `getExpressionName`: obtém o nome da classe `Expression` criada dentro do modelo concreto da *DSML4ClinicalScore*.
- `getExpressions`: obtém uma coleção de objetos do tipo `Expression` criados dentro do modelo concreto da *DSML4ClinicalScore*.
- `getNumberOfValues`: obtém a quantidade de instâncias do atributo `value` especificadas para um objeto do tipo `Variable` dentro do modelo concreto da *DSML4ClinicalScore*.

- `getSensorQuantity`: obtém a quantidade de objetos do tipo `Sensor` especificados dentro do modelo concreto da *DSML4ClinicalScore*.
- `getVariableName`: obtém o nome da classe `Variable` criada dentro do modelo concreto da *DSML4ClinicalScore*.
- `verifyRuleWithSensor`: retorna verdadeiro ou falso para a consulta de objetos do tipo `Rule` especificados dentro do modelo concreto da *DSML4ClinicalScore*, que estejam associados a um objeto do tipo `Sensor`.

Para exemplificar o uso de *queries* pelo *MDD4ClinicalScores*, a *query* `verifyRuleWithSensor` é exposta no Código 4.1. O que esse trecho de código faz é percorrer a árvore de objetos do tipo `Expression` especificados dentro do modelo concreto da *DSML4ClinicalScore*, procurando por uma ocorrência em que o objeto do tipo `RefVariable` que compõe a regra está associado a uma instância do tipo `Sensor`. Esta consulta retorna verdadeiro se a busca for bem sucedida ou falso, caso contrário.

**Código 4.1:** Código da *query* `verifyRuleWithSensor`.

```
1 [query private verifyRuleWithSensor(rule:Rule) : Boolean =
2 rule.eAllContents(Expression)->exists(exp:Expression |
3 exp.getExpressionName().equalsIgnoreCase('RefVariable')
4 and exp.eGet('variable').oclIsTypeOf(Sensor))]
```

## 4.4 Ambiente de Execução

Para executar os artefatos de software criados pelo componente de Geração Automática de Código, o *MDD4ClinicalScores* utiliza um ambiente de suporte baseado em uma infraestrutura de middleware, que recebe a ontologia *Deklaer* recém-produzida e gera de forma semiautomática uma aplicação para IoT com as configurações descritas nesta ontologia. A arquitetura do Ambiente de Execução da abordagem baseia-se no trabalho de [Pinheiro et al. 2018], no qual os autores descrevem um framework dirigido por ontologias que usa o middleware *ContextNet*, conforme ilustrado na Figura 4.6.

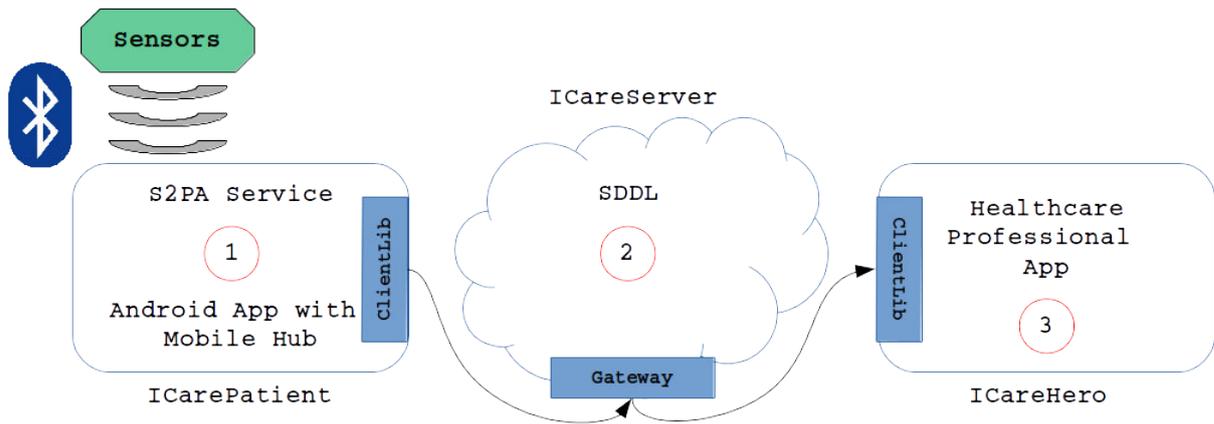


Figura 4.6: Ambiente de Execução do *MDD4ClinicalScores*.

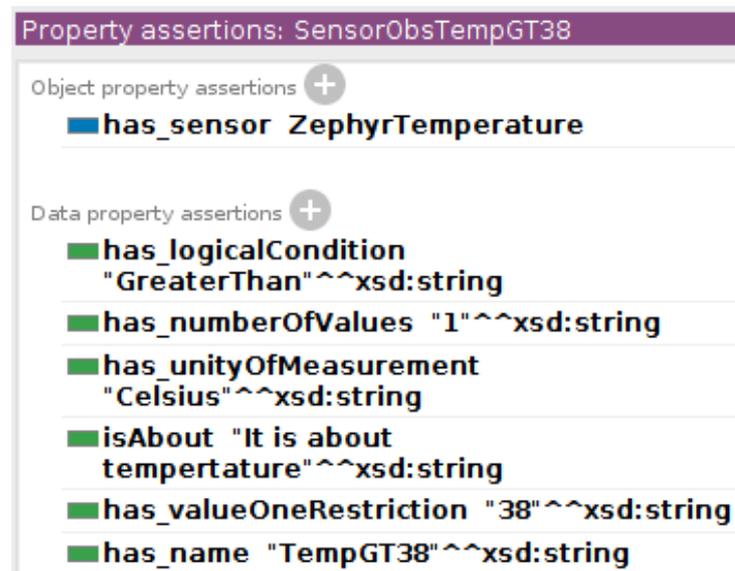


Figura 4.7: Exemplo de uma instância da classe *SensorObs*.

A parte identificada com o número 2 na Figura 4.6 representa o servidor *ICareServer* localizado no núcleo do SDDL. O *ICareServer* é responsável por configurar o Ambiente de Execução com a ontologia *Deklaer* gerada pelo componente de Geração Automática de Código do *MDD4ClinicalScores*. Quando este ambiente executa pela primeira vez, ele usa a API OWL do *Protégé* para ler a ontologia e carregá-la. As classes da ontologia são consumidas pelo *ICareServer* como classes Java. Assim que o desenvolvedor da aplicação carrega a ontologia dentro deste servidor, ele identifica todas as classes *SensorObs* (as observações ou regras das variáveis do escore clínico que os sensores devem verificar) e cria uma regra CEP correspondente. Em seguida, estas regras CEP são enviadas aos M-Hubs que estão conectados ao *ICareServer*. Para esta pesquisa, o código deste servidor foi alterado para que ele possa enviar todas as

informações associadas ao evento criado, não somente uma mensagem de texto como está na proposta inicial de [Pinheiro et al. 2018].

Exemplificando, suponhamos que o *ICareServer* receba dentro da ontologia *Deklaer* nele carregada a instância de *SensorObs* mostrada na Figura 4.7. Esta instância indica que o sensor **ZephyrTemperature** deve verificar quando a temperatura do paciente exceder 38°C. Esta observação será convertida pelo *ICareServer* na regra CEP do Código 4.2.

**Código 4.2:** Regra CEP associada à instância de *SensorObs* da Figura 4.7.

```
1  { MEPAQuery:
2    { type:"stop",
3      label:"ZephyrTemperature",
4      rule:"SELECT (sensorValue[0]) as value FROM SensorData
5        (sensorName='zephyrtemperature')
6          WHERE sensorValue[0] > 38",
7      target:"global" } }
```

A parte identificada com o número 1 na Figura 4.6 representa a aplicação *ICarePatient* que executa o M-Hub em um dispositivo móvel com suporte ao sistema operacional Android. O *ICarePatient* é responsável pela comunicação com os sensores corporais do paciente ou presentes no ambiente para coletar as observações descritas pela ontologia *Deklaer* carregada no *ICareServer*. O ambiente de execução provê uma aplicação Android padrão com a implementação dos drivers de alguns sensores pré-configurados dentro do serviço S2PA. Novos sensores com tecnologia Bluetooth podem ser incorporados ao Ambiente de Execução do *MDD4ClinicalScores*, desde que o código para identificação e comunicação com este novo sensor seja inserido dentro do serviço S2PA que executa no *ICarePatient*. O *ICarePatient* usa uma biblioteca Java chamada *ClientLib* para conectar-se ao núcleo SDDL. Para esta pesquisa, o código do serviço S2PA do *ICarePatient* foi modificado para que ele possa reconhecer outros sensores que não constam da implementação inicial de [Pinheiro et al. 2018].

A parte identificada com o número 3 na Figura 4.6 representa a aplicação *ICareHero*, podendo ser uma aplicação desktop ou móvel. O *ICareHero* é responsável por receber as informações do *ICareServer* referentes às pontuações das regras que foram observadas pelo *ICarePatient*, computar o somatório dos valores e informar a avaliação do escore clínico de acordo com o resultado obtido. Da mesma forma que o

*ICarePatient*, o *ICareHero* também utiliza a biblioteca *ClientLib* para comunicar-se com o núcleo SDDL. Para esta pesquisa, a aplicação do *ICareHero* foi desenvolvida por completo.

## 4.5 Considerações Finais

Este capítulo discutiu o *MDD4ClinicalScores*, uma abordagem para desenvolvimento de software que utiliza conceitos de MDD para resolução de escores clínicos em ambientes de computação ubíqua. Aqui foram descritos os componentes desta abordagem, ressaltando o papel de cada um para o alcance do objetivo por ela proposto.

O primeiro componente do *MDD4ClinicalScores* apresentado foi a Geração Automática de Código. Este componente engloba a *DSML4ClinicalScore* e uma transformação M2T. A *DSML4ClinicalScore* é uma linguagem de domínio específico voltada para modelagem de escores clínicos de modo geral, que constitui a principal contribuição desta pesquisa. Este capítulo detalhou as motivações e a metodologia para o desenvolvimento desta linguagem, além dos conceitos do seu metamodelo. Em seguida, foram descritas as regras de transformação implementadas no escopo desta abordagem. A transformação recebe o escore clínico especificado pelo modelo concreto da *DSML4ClinicalScore* e gera uma ontologia OWL correspondente, baseada na linguagem *Deklaer* [Pinheiro et al. 2018].

O segundo componente do *MDD4ClinicalScores* detalhado foi o Ambiente de Execução. Este componente foi desenvolvido com base no trabalho de [Pinheiro et al. 2018], no qual preconizam um framework dirigido por ontologias capaz de gerar e executar semiautomaticamente uma aplicação para IoT que envolve sensores. O Ambiente de Execução desta abordagem recebe os artefatos de software gerados pelo componente da Geração Automática de Código, e cria de maneira semiautomática uma aplicação mHealth para monitoramento remoto de pacientes com base no escore clínico previamente especificado. Este capítulo descreveu os elementos deste ambiente, pontuando o papel de cada um na geração da aplicação final.

Conforme ratificado por [Fowler 2010], ao usar uma solução via DSL para resolver problemas de escore clínico via computação ubíqua, *MDD4ClinicalScores*

---

mostra-se mais vantajosa em comparação às demais abordagens. Primeiro, nossa abordagem aumenta a produtividade no desenvolvimento de aplicações dentro do domínio de escores clínicos, porque a DSL projetada disponibiliza uma forma mais expressiva e compreensiva para ler e manipular o problema, o que facilita não somente o entendimento sobre a visão global do sistema, mas também o descobrimento de erros e duplicidades. Segundo, *MDD4ClinicalScores* diminui a complexidade envolvida nessas circunstâncias, pois facilita tanto a construção quanto a manutenção desse tipo de software.

## 5 Avaliação do MDD4ClinicalScores

O presente capítulo aborda a avaliação da abordagem do *MDD4ClinicalScores*. Para avaliar o *MDD4ClinicalScores*, dois aspectos foram considerados: a validação da linguagem *DSML4ClinicalScore* e elaboração de estudos de caso. O processo de validação da *DSML4ClinicalScore* teve como objetivo verificar sua expressividade por meio da modelagem de oito casos concretos de diferentes especificações de escore clínico. Os estudos de caso foram elaborados para checar a viabilidade da proposta do *MDD4ClinicalScores*, englobando o processo desde a criação do modelo concreto através do metamodelo da *DSML4ClinicalScore*, passando pela transformação deste modelo em uma ontologia OWL descrita conforme a linguagem *Deklaer*, até a geração e execução da aplicação mHealth de monitoramento focado no escore especificado.

Além disso, este capítulo discute como a proposta do *MDD4ClinicalScores* consegue resolver as problemáticas descritas no Capítulo 1, da mesma forma que apresenta as principais limitações presentes nesta abordagem.

### 5.1 Validação da DSML4ClinicalScore

Para demonstrar a expressividade da *DSML4ClinicalScore*, das 89 especificações de escore clínico analisadas na Subseção 4.2.1, optou-se pela modelagem de oito casos concretos, de tal modo que estes exemplos explorassem diferentes características relacionadas às variáveis, regras para cálculo da pontuação e avaliações. Para construir estes casos concretos, usamos a ferramenta de validação do EMF para modelos e metamodelos. A Tabela 5.1 traz a relação dos escores clínicos selecionados e suas especificações estão disponíveis na plataforma online do MD+Calc [MDCalc 2005].

Para ilustrar a modelagem de um escore clínico através do metamodelo da *DSML4ClinicalScore*, usaremos o exemplo do escore MEWS descrito na Subseção 2.1.2. Devido este escore clínico possuir uma especificação bastante complexa e por questões

**Tabela 5.1:** Relação de escores clínicos usados para validação da *DSML4ClinicalScore*.

Escore Clínico	Distúrbio Alvo
Escore 4Ts [Lo et al. 2006]	Trombocitopenia
Escore <i>CHADS<sub>2</sub></i> [Gage et al. 2001]	Acidente vascular cerebral
Escore CPIS [Schurink et al. 2004]	Pneumonia
Escore CURB-65 [Lim et al. 2003]	Pneumonia
Escore HAS-BLED [Pisters et al. 2010]	Risco de sangramento
Escore MEWS	Grau de doença
Índice de Risco TIMI [Wiviott et al. 2006]	Doença coronária aguda
Crítérios Wells para EP [Wells et al. 2001]	Embolia pulmonar

de limitação de espaço, apresentamos seu respectivo modelo concreto somente parcialmente nesta dissertação. As modelagens deste caso concreto e das outras 7 especificações contidas na Tabela 5.1 podem ser acessadas através da seguinte URL: <<http://www.lsd.ufma.br/projetos/mdd4clinicalscores>>.

A Figura 5.1 mostra o modelo concreto do escore clínico MEWS conforme o metamodelo proposto, com uma representação por diagrama de classes UML. Para começar o processo de modelagem da especificação, o usuário deve declarar uma série de atributos (*id*, *description* e *type*) para um objeto do tipo *Specification*. Depois disso, o usuário pode definir as variáveis, regras para cálculo da pontuação e avaliações do escore clínico.

Considerando as variáveis declaradas para o escore MEWS, a Figura 5.1 demonstra como suas variáveis e respectivos atributos (*id*, *unity*, *value* e *dataType*) podem ser modelados de acordo com o metamodelo da *DSML4ClinicalScore*. Este escore clínico possui cinco objetos do tipo *Variable* relacionados com a pressão sanguínea sistólica, batimento cardíaco, frequência respiratória, temperatura e escore AVPU.

Com relação às regras para cálculo da pontuação do escore MEWS, a Figura 5.1 evidencia como suas regras são definidas conforme o metamodelo proposto. Esta especificação tem um total de 23 (vinte e três) regras, com 5 (cinco) objetos do tipo *Rule* associados a pressão sanguínea sistólica, 6 (seis) objetos associados ao batimento cardíaco, 5 (cinco) objetos associados a frequência respiratória, 3 (três) objetos associados à temperatura, e 4 (quatro) objetos associados ao escore AVPU. “N = 20” indica que existem 20 objetos do tipo *Rule* representados da mesma forma daqueles mostrados na Figura 5.1. Esta figura mostra apenas os atributos (*id*, *score*

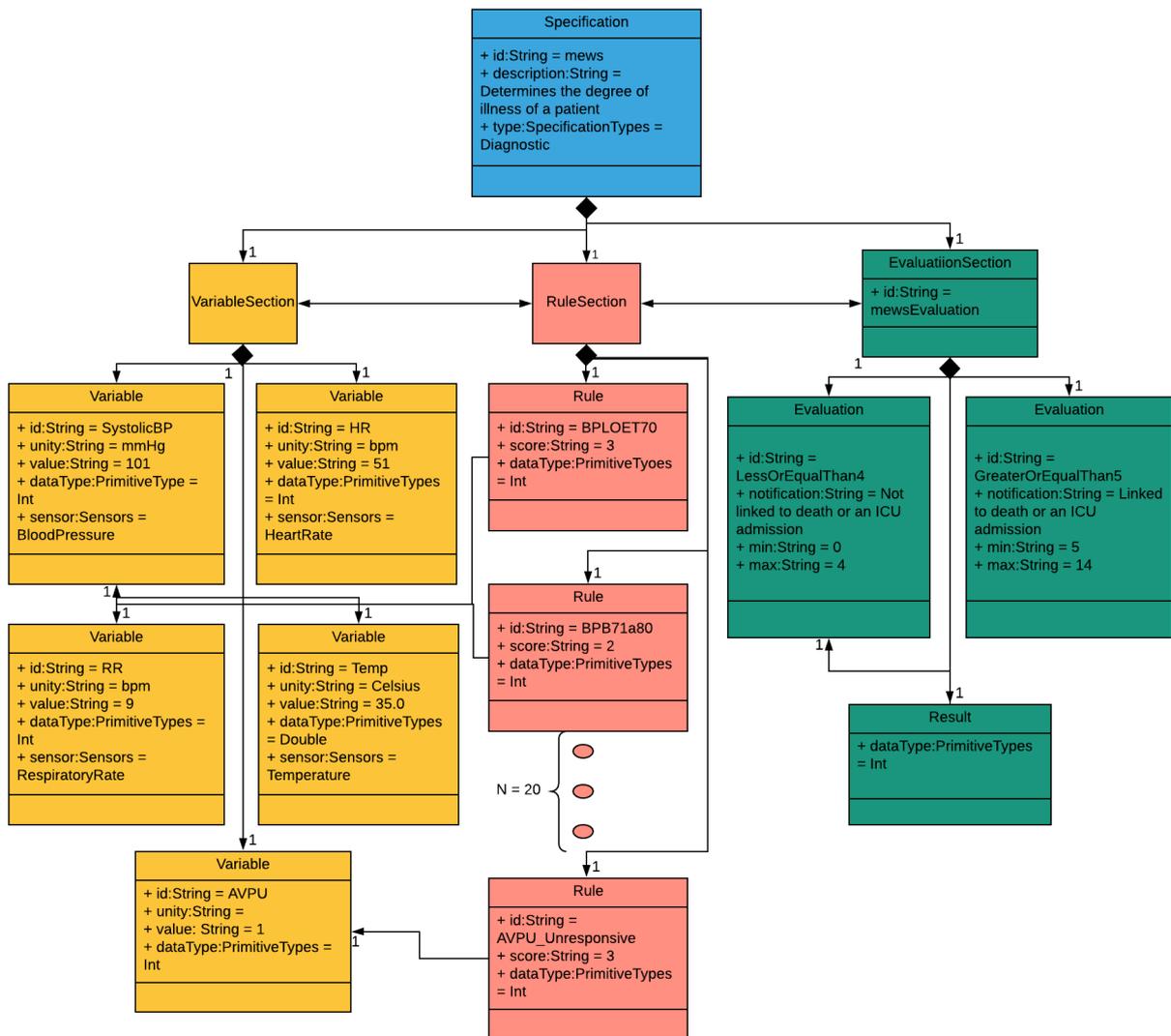


Figura 5.1: Escore MEWS conforme o metamodelo da *DSML4ClinicalScore*.

e `dataType`) de duas regras relacionadas a pressão sanguínea sistólica e de uma regra relacionada ao escore AVPU. Por exemplo, para compor a regra “`id:String = BPB71a80`” (Pressão sanguínea sistólica entre 71 e 80 mmHg), o usuário deve proceder da seguinte forma (ver Figura 5.2):

1. Criar um objeto do tipo `LogicalOperator` com a constante `And`;
2. A partir desse objeto, criar duas instâncias do tipo `RelationalOperator`, uma com a constante `GreaterThan` e outra com a constante `LessOrEqualThan`;
3. A partir do objeto do tipo `GreaterThan`, criar uma instância de `RefVariable` e associá-la ao objeto do tipo `Variable` de “`id = SystolicBP`”, e uma instância de `Constant` de “`value = 70`”;

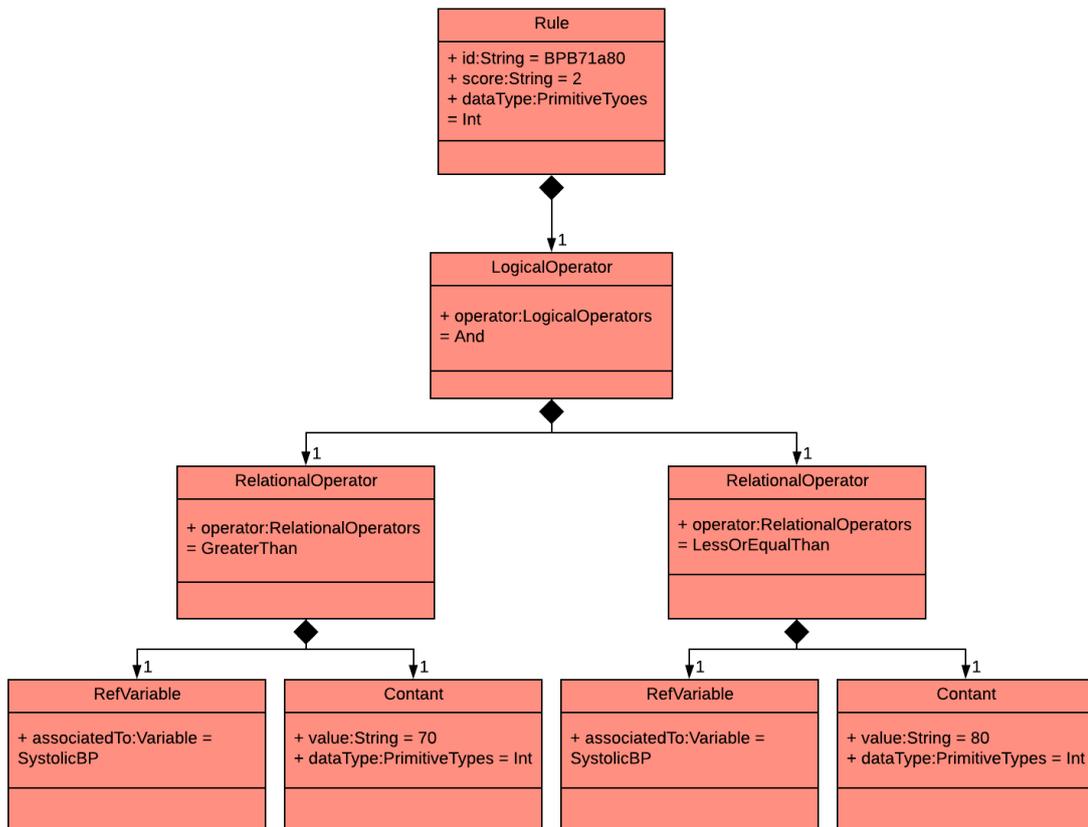


Figura 5.2: Composição da regra “**id:String = BPB71a80**” do escore MEWS conforme o metamodelo da *DSML4ClinicalScore*.

4. A partir do objeto do tipo `LessOrEqualThan`, criar uma instância de `RefVariable` e associá-la ao objeto do tipo `Variable` de “**id = SystolicBP**”, e uma instância de `Constant` de “**value = 80**”.

Por último, a Figura 5.1 também apresenta os atributos dos modelos de avaliação do escore MEWS, segundo o metamodelo da *DSML4ClinicalScore*. Na Figura 5.1, estão representados 2 (dois) objetos do tipo `Evaluation` com seus respectivos atributos (`id`, `notification`, `min`, `max` e `dataType`) e 1 (um) objeto do tipo `Result`, que compõem a avaliação desta especificação, indicando o intervalo de pontuação de cada modelo de interpretação e sua respectiva notificação.

Na modelagem dos oito casos concretos, a linguagem da *DSML4ClinicalScore* permitiu que o usuário modelasse corretamente variáveis de entrada para um escore clínico de diferentes tipos, seja ela obtida manual ou eletronicamente. Por exemplo, é possível expressar a idade ou o gênero de um paciente, medidas de batimento cardíaco ou pressão sanguínea, resultados de exames, históricos de várias doenças, entre outros. Além disso, a linguagem também

consegue expressar os atributos necessários para identificar uma variável dentro das 89 especificações examinadas nesta pesquisa, independente de como esse atributo é representado (numérico ou textual).

No tocante às regras para cálculo da pontuação, a *DSML4ClinicalScore* permitiu que o usuário modelasse corretamente diversos formatos de regras lógicas, relacionais e matemáticas. Por exemplo, é possível expressar a fórmula matemática que calcula o Índice de Risco TIMI representada na fórmula 5.1, que usa três variáveis relacionadas a batimento cardíaco, idade e pressão sanguínea sistólica para compor sua regra para cálculo da pontuação. Igualmente, é possível expressar uma regra lógica do exemplo do Escore CURB-65 descrito na fórmula 5.2, que é composta por duas expressões relacionais envolvendo as variáveis das pressões sanguíneas sistólica e diastólica deste escore clínico. Por fim, também é possível expressar uma regra que contém opções de múltipla escolha como o exemplo mostrado na fórmula 5.3, que é parte do Escore 4Ts para trombocitopenia. Resumindo, a *DSML4ClinicalScore* consegue expressar regras que retornam um valor verdadeiro ou falso, bem como fórmulas puramente matemáticas.

$$TIMI\ Index = Heart\ rate * (Age/10)^2 / Blood\ pressure \quad (5.1)$$

$$IF\ SystolicBP < 90\ OR\ DiastolicBP \leq 60\ THEN\ +1 \quad (5.2)$$

$$\left[ \begin{array}{l} Other\ causes = None\ apparent\ +2 \\ \quad \quad \quad = Possible\ +1 \\ \quad \quad \quad = Definite\ 0 \end{array} \right] \quad (5.3)$$

Para a seção de avaliação, a *DSML4ClinicalScore* permitiu que o usuário modelasse corretamente diversos modelos de avaliação. Desde modelos de avaliação bem simples, como a avaliação do Índice de Risco TIMI que define apenas uma probabilidade como resultado, até os mais complexos, como o estabelecido pelos Critérios Wells para embolia pulmonar que apresenta modelos de avaliação em duas ou três camadas. O desenvolvimento dos oito casos concretos demonstrou que a linguagem consegue expressar informações que são coerentes com uma avaliação de escore clínico.

Para mais detalhes, os plugins contendo o metamodelo da *DSML4ClinicalScore*, os projetos para edição e testes deste metamodelo, e a modelagem desses oito casos concretos estão disponíveis no seguinte endereço da Web: <<http://www.lsd.ufma.br/projetos/mdd4clinicalscores>>.

## 5.2 Estudos de Caso

Para verificar a viabilidade da abordagem proposta pelo *MDD4ClinicalScores*, optou-se pela implementação de dois estudos de caso. O estudo de caso é um método que serve para avaliar uma teoria de pesquisa, ratificando que em um caso específico a teoria funciona [Yin 2014]. Cada estudo de caso engloba todas as atividades necessárias para a modelagem da especificação do escore clínico, o processo de transformação desta especificação, e a criação e execução da aplicação que utiliza o escore clínico em questão.

Para elaborar os estudos de caso, levamos em consideração a descrição da aplicação para IoT usando a linguagem *Deklaer* discutida na Subseção 4.3.1, criada por meio da transformação M2T do *MDD4ClinicalScores* a partir do modelo concreto da *DSML4ClinicalScore* (escore clínico especificado). Além disso, desenvolvemos uma aplicação desktop (*ICareHero*) com a biblioteca Java do *ClientLib* para comunicação com o núcleo do SDDL. Variáveis dos escores clínicos utilizados nestes estudos de caso, que são provenientes de fontes que não sejam sensores (entrevista, prontuário eletrônico ou outro escore), foram simuladas dentro do código do *ICareHero* para que fossem gerados valores aleatórios correspondentes dentro do intervalo permitido pela própria especificação. Os testes foram realizados por intermédio de um notebook Dell Inspiron com Windows 10, processador Core I7 de até 4.0 GHz e 8 GB de RAM.

Levando em consideração o Ambiente de Execução descrito na Seção 4.4, ambas as aplicações do *ICareServer* e *ICareHero* estão implementadas em Java. Em particular, *ICareServer* foi implantada em um nó de processamento dentro do SDDL e usou a API OWL 3.4.2 e o banco de dados *TimescaleDB*<sup>1</sup> para armazenar as ontologias e os dados de contexto que são enviados aos M-Hubs, respectivamente. Por sua vez, a

---

<sup>1</sup><http://www.timescale.com>.

aplicação do *ICarePatient* está implementada em Android com o serviço S2PA e executa em um smartphone com M-Hub para coletar os dados dos sensores.

### 5.2.1 Escore MEWS

Esta subseção descreve as etapas realizadas desde a criação da especificação do escore MEWS através do metamodelo da *DSML4ClinicalScore*, até a geração da aplicação via Ambiente de Execução do *MDD4ClinicalScores*, que avalia a pontuação deste escore clínico durante o monitoramento remoto de um paciente por meio de sensores.

#### Criação do Modelo Concreto

A primeira etapa para a criação da aplicação mHealth de monitoramento focado no escore MEWS é a modelagem desta especificação de acordo com o metamodelo definido pela *DSML4ClinicalScore*. Esta modelagem foi discutida na Seção 5.1 (ver Figura 5.1). As variáveis, as regras para cálculo da pontuação e os modelos de avaliação deste escore clínico estão descritos na Subseção 2.1.2.

#### Geração da Ontologia *Deklaer*

A segunda etapa para a criação da aplicação mHealth de monitoramento do escore clínico MEWS é a geração da ontologia OWL conforme a linguagem *Deklaer*, a partir do modelo concreto especificado pela *DSML4ClinicalScore*. Levando em consideração o modelo concreto descrito na Seção 5.1, o *MDD4ClinicalScores* gerou via transformação M2T a ontologia exibida na Figura 5.3. A ontologia resultante ficou composta da seguinte forma: 8 (oito) indivíduos da classe *ActRule*, 1 (um) de *Entity*, 8 (oito) de *Notification*, 8 (oito) de *ObsActRule*, 8 (oito) de *ObsRule*, 4 (quatro) de *Sensor* e 8 (oito) de *SensorObs*. Em mais detalhes, a Figura 5.3 revela os indivíduos criados para as classes *Entity*, *Sensor* e *SensorObs*, respectivamente.

Tendo em vista as regras de transformação discutidas na Seção 4.3, é importante salientar que a ontologia da Figura 5.3 foi editada manualmente para que o Ambiente de Execução do *MDD4ClinicalScores* fosse capaz de gerar a aplicação mHealth focada na avaliação do escore MEWS. Neste caso, são obrigatórias as

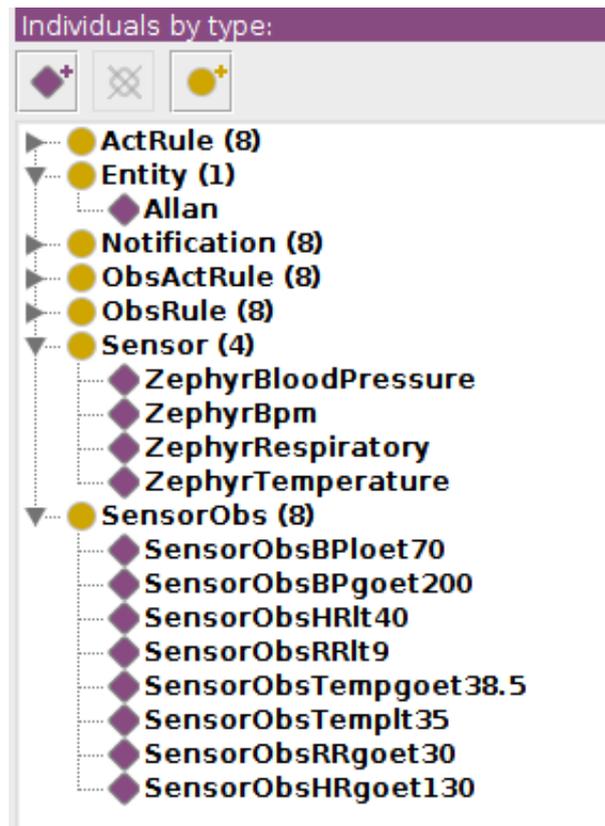


Figura 5.3: Aplicação mHealth gerada para o escore MEWS.

inserções manuais do indivíduo da classe `Entity` (**Allan**) e suas respectivas *data properties* e *object properties*, pois se trata da pessoa a ser monitorada pelos sensores corporais ou espalhados no ambiente, e da *data property* `Sensor:has_sensorName`, que identifica qual o sensor está sendo utilizado no monitoramento, conforme preconizado pela linguagem *Deklaer*. Sem essas informações, o *MDD4ClinicalScores* não reconhece a descrição da ontologia nele carregado e, conseqüentemente, fica impossibilitado de gerar e executar esta aplicação.

As Figuras 5.4(a) e 5.4(b) exibem as métricas da ontologia (aplicação) para o escore clínico MEWS. Na Figura 5.4(a) são mostradas as métricas gerais dessa ontologia, tais como a quantidade de axiomas, classes, indivíduos etc. Já a Figura 5.4(b) apresenta os indivíduos de classe e os relacionamentos entre eles, bem como suas respectivas *data properties* e *object properties*, que foram geradas a partir do modelo concreto da especificação MEWS. Nota-se que a transformação conseguiu produzir 45 (quarenta e cinco) indivíduos de classe, 64 (sessenta e quatro) *object properties* e 94 (noventa e quatro) *data properties*.

Ontology metrics:		Ontology metrics:	
<b>Metrics</b>		<b>Individual axioms</b>	
Axiom	305	ClassAssertion	45
Logical axiom count	227	ObjectPropertyAssertion	64
Declaration axioms count	78	DataPropertyAssertion	94
Class count	15	NegativeObjectPropertyAssertion	0
Object property count	7	NegativeDataPropertyAssertion	0
Data property count	11	SameIndividual	0
Individual count	45	DifferentIndividuals	0
Annotation Property count	0		
DL expressivity	AL(D)		

Figura 5.4: Métricas da aplicação mHealth do escore MEWS.

### Aplicação de Monitoramento

A terceira e última etapa para a criação da aplicação mHealth de monitoramento do escore clínico MEWS é a submissão da ontologia gerada pelo *MDD4ClinicalScores* ao seu Ambiente de Execução. Como dito na Seção 4.4, o monitoramento do paciente via avaliação do escore clínico especificado é efetivado pelo trabalho conjunto das aplicações *ICareServer*, *ICarePatient* e *ICareHero*. *ICareServer* carrega a ontologia da Figura 5.3, cria as 8 (oito) regras CEP nela descritas e envia ao *ICarePatient* que executa o M-Hub responsável pelo monitoramento. Por sua vez, *ICarePatient* trata-se da aplicação gerada semiautomaticamente pelo *MDD4ClinicalScores* e é responsável por disparar os eventos que porventura ocorram de acordo com as regras CEP predefinidas pelo *ICareServer*.

Para testar a viabilidade do *MDD4ClinicalScores*, *ICareHero* foi desenvolvida como uma aplicação desktop para receber as notificações do *ICareServer*, computar o somatório das pontuações das variáveis dessa especificação e exibir a avaliação do escore de acordo com o resultado obtido. Neste caso específico, no código desta aplicação foi implementada uma rotina que gera pseudoaleatoriamente o valor do escore AVPU, cujo valor não pode ser obtido por meio de sensores e varia dentro do intervalo de 0 a 3 (ver Tabela 2.2).

A Figura 5.5 demonstra os valores de saída referentes à execução da aplicação de monitoramento do escore MEWS no momento da coleta dos dados provenientes dos sensores e do escore AVPU. Na Figura 5.5, “*zephyrbloodpressure*” representa a variável da pressão sanguínea sistólica, “*zephyrbpm*” representa o batimento cardíaco, “*zephyrrespiratory*” representa a frequência respiratória e “*zephyrtemperature*” representa a temperatura. No momento da coleta, os sensores

```

Debug: HelloMobileReceiver x
Debugger Console
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Connected to the target VM, address: '127.0.0.1:50183', transport: 'socket'
ReliableSocket: new utils pool size = 3
null
zephyrbloodpressure - 205.0 mmHg - ObsRuleBPgoet200 - Score: 2
zephyrbpm - 37.0 bpm - ObsRuleHRlt40 - Score: 2
zephyrrespiratory - 39.0 rpm - ObsRuleRRgoet30 - Score: 3
zephyrtemperature - 40.0 Celsius - ObsRuleTempgoet38.5 - Score: 2
AVPU Score: Reacts to voice - 1
Statistically linked to death or ICU admission Result: 10

```

Figura 5.5: Execução da aplicação mHealth de monitoramento para o escore MEWS.

registraram o que segue: “205,0 mmHg” de pressão sanguínea sistólica, “37 bpm” de batimento cardíaco, “39 rpm” de frequência respiratória e “40.0 °C” de temperatura. O valor pseudoaleatório gerado para o escore AVPU foi “Reage a voz”. Para os valores registrados, a pontuação das variáveis foram: pressão sanguínea sistólica (+2), batimento cardíaco (+2), frequência respiratória (+3), temperatura (+2) e escore AVPU (+1). Logo, o resultado obtido pelo somatório das pontuações foi “10”, cuja avaliação correspondente é “Estatisticamente ligado a morte ou admissão na UTI”. Esses valores estão descritos na Tabela 2.2.

### 5.2.2 Escore CURB-65

Com o objetivo de demonstrar a abrangência e flexibilidade da nossa abordagem, esta subseção ilustra outro estudo de caso com a utilização do *MDD4ClinicalScores* para o escore CURB-65. O escore CURB-65 objetiva estimar a mortalidade por pneumonia e tem seu modelo de especificação representado na Tabela 5.2. Esta especificação diferencia-se do escore MEWS pelo distúrbio alvo a que se aplica e pela forma como suas variáveis, regras para cálculo da pontuação e avaliações são definidas, possuindo: 5 (cinco) variáveis relacionadas a **confusão**, **taxa de Blood Urea Nitrogen (BUN)** ou **nitrogênio uréico sanguíneo**, **frequência respiratória**, **pressão sanguínea sistólica** e **idade**; 10 (dez) regras para cálculo da pontuação do escore, sendo 2 (duas) associadas a cada variável; e 3 (três) modelos de avaliação [Lim et al. 2003]. Esta diferenciação acarreta mudanças no modelo concreto criado pela *DSML4ClinicalScore* e, conseqüentemente, na aplicação final que será gerada pelo *MDD4ClinicalScores*.

Da mesma forma que o estudo de caso descrito na Subseção 5.2.1, aqui são descritas as etapas percorridas desde a criação do modelo concreto do escore CURB-65 através do metamodelo da *DSML4ClinicalScore*, até a geração da aplicação via Ambiente de Execução do *MDD4ClinicalScores*, que avalia a pontuação deste escore clínico durante o monitoramento remoto de um paciente através de sensores.

**Tabela 5.2:** Especificação do escore CURB-65.

Variável	Valor	Pontuação
<i>Confusão</i>	Não ou Sim	0 ou +1
<i>BUN &gt; 19 mg/dL</i>	Não ou Sim	0 ou +1
<i>Frequência respiratória <math>\geq 30</math></i>	Não ou Sim	0 ou +1
<i>Pressão sanguínea sistólica <math>\leq 90</math> mmHg</i>	Não ou Sim	0 ou +1
<i>Idade <math>\geq 65</math></i>	Não ou Sim	0 ou +1
Avaliação		Resultado
1.5% de risco de mortalidade		0 ou 1
9.2% de risco de mortalidade		2
22% de risco de mortalidade		$\geq 3$

### Criação do Modelo Concreto

A Figura 5.6 ilustra a modelagem do escore CURB-65 em uma representação na forma de diagrama de classes UML, conforme o metamodelo proposto pela *DSML4ClinicalScore*. O modelo concreto resultante é composto por: 1 (um) objeto do tipo *Specification* para identificar o escore clínico; 5 (cinco) objetos do tipo *Variable*, um para cada variável especificada; 10 (dez) objetos do tipo *Rule* para modelar as regras definidas; 3 (três) objetos do tipo *Evaluation* para descrever as avaliações da especificação; e 1 (um) objeto do tipo *Result* para armazenar o resultado do escore clínico. Este modelo foi construído com base nas informações da Tabela 5.2. A modelagem das variáveis, regras para cálculo da pontuação e avaliações do escore CURB-65 seguiu o mesmo procedimento descrito na Seção 5.1.

### Geração da Ontologia *Deklaer*

A partir do modelo concreto da Figura 5.6, o *MDD4ClinicalScores* gerou via transformação M2T a ontologia da Figura 5.7, que é descrita conforme *Deklaer*. A ontologia resultante ficou composta da seguinte forma: 4 (quatro) indivíduos da

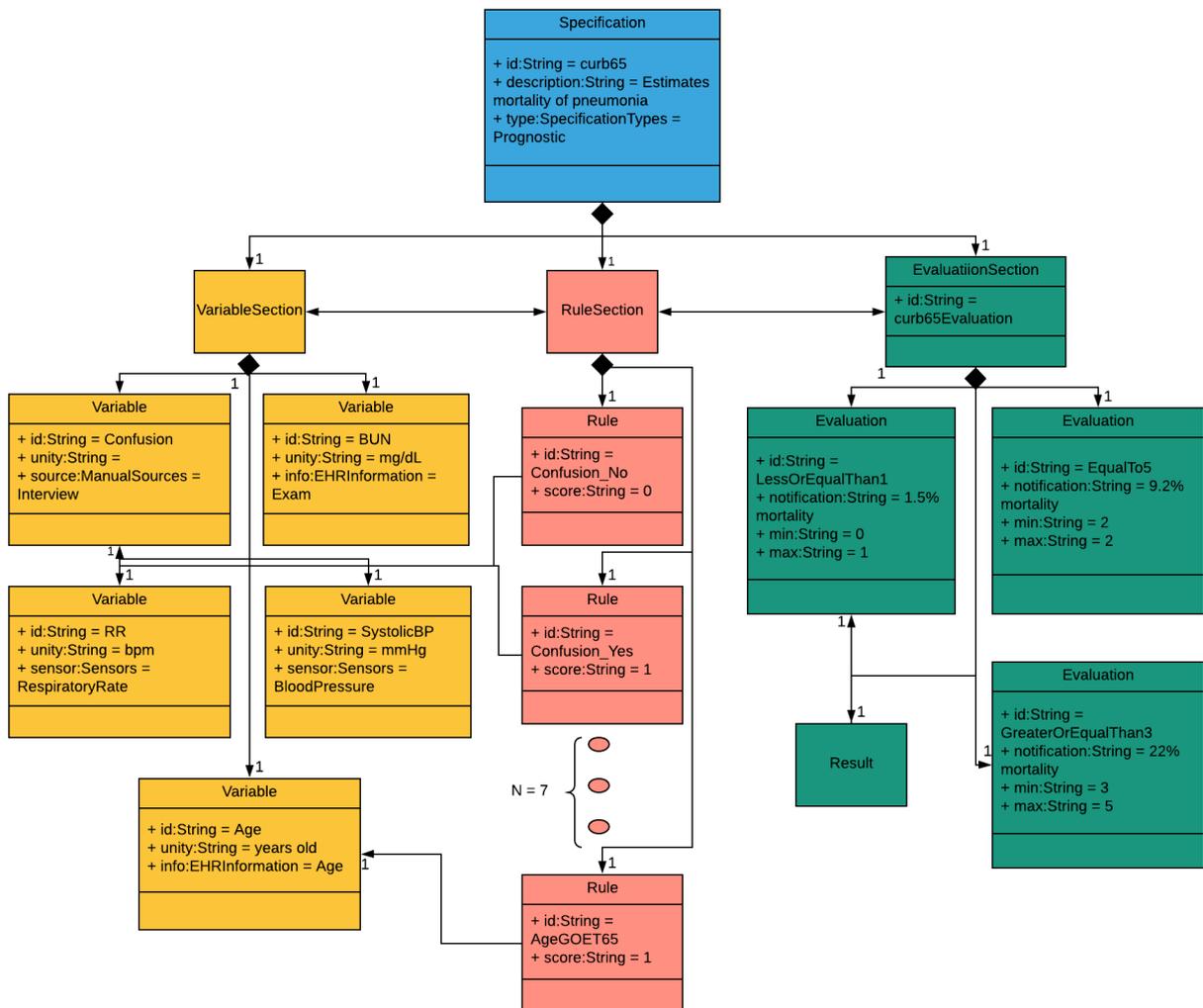


Figura 5.6: Escore CURB-65 conforme *DSML4ClinicalScore*.

classe *ActRule*, 1 (um) de *Entity*, 4 (quatro) de *Notification*, 4 (quatro) de *ObsActRule*, 4 (quatro) de *ObsRule*, 2 (dois) de *Sensor* e 4 (quatro) de *SensorObs*. Em mais detalhes, a Figura 5.7 ressalta os indivíduos criados para as classes *ActRule*, *Notification* e *ObsRule*, respectivamente.

Do mesmo modo que a ontologia descrita na Subseção 5.2.1, a classe *Entity*, seu indivíduo (**Allan**) e respectivas *data properties* e *object properties*, e a *data property* *Sensor:has\_sensorName* também foram inseridos manualmente dentro da ontologia da Figura 5.7. Desta forma, o *MDD4ClinicalScores* teve condições de gerar a aplicação específica para o escore CURB-65. Por sua vez, as Figuras 5.8(a) e 5.8(b) exibem as métricas da ontologia (aplicação) deste escore clínico. Na Figura 5.8(a) são mostradas as métricas gerais da ontologia, como a quantidade de axiomas, classes, indivíduos etc. Já a Figura 5.8(b) apresenta os indivíduos de classe e os relacionamentos entre eles, bem como suas respectivas *data properties* e *object properties*,

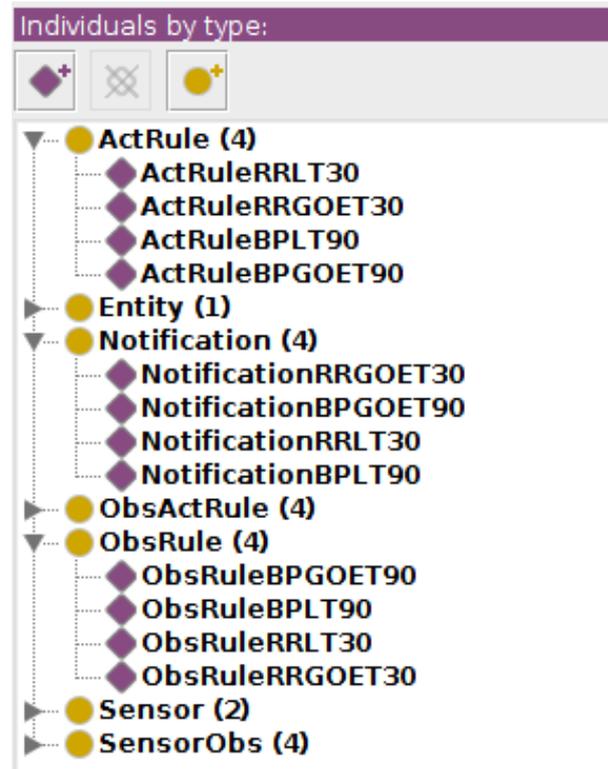


Figura 5.7: Aplicação mHealth gerada para o escore CURB-65.

que foram gerados a partir do modelo concreto da especificação CURB-65. Nota-se que a transformação conseguiu produzir 23 (vinte e três) indivíduos de classe, 33 (trinta e três) *object properties* e 54 (cinquenta e quatro) *data properties*.

Ontology metrics:		Ontology metrics:	
<b>Metrics</b>		<b>Individual axioms</b>	
Axiom	190	ClassAssertion	23
Logical axiom count	134	ObjectPropertyAssertion	33
Declaration axioms count	56	DataPropertyAssertion	54
Class count	15	NegativeObjectPropertyAssertion	0
Object property count	7	NegativeDataPropertyAssertion	0
Data property count	10	SameIndividual	0
Individual count	24	DifferentIndividuals	0
Annotation Property count	0		
DL expressivity	AL(D)		

(a) (b)

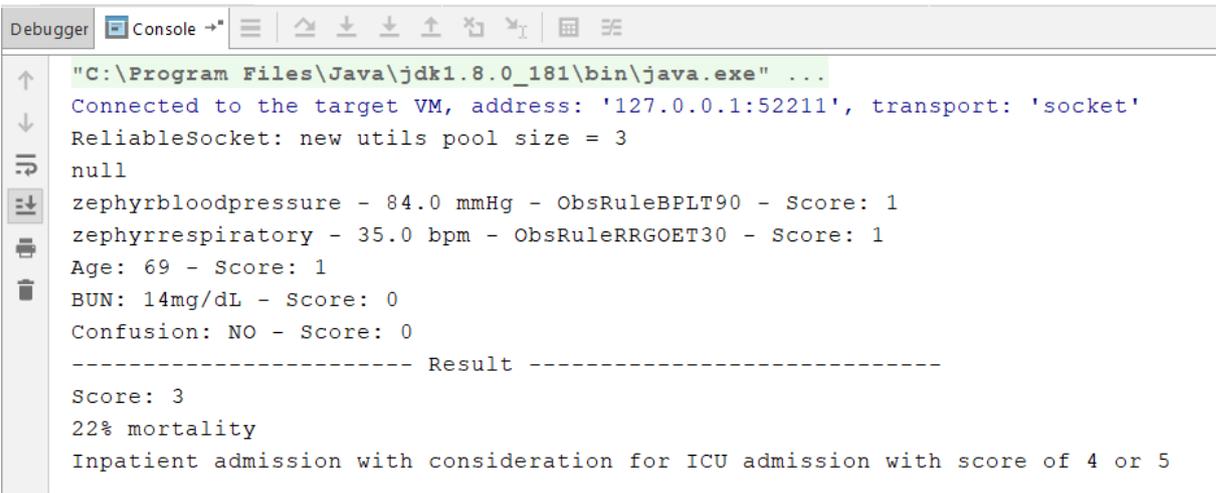
Figura 5.8: Métricas da aplicação mHealth do escore CURB-65.

## Aplicação de Monitoramento

A ontologia da Figura 5.7 foi então submetida ao Ambiente de Execução do *MDD4ClinicalScores* para geração e execução da aplicação mHealth de monitoramento focada no escore CURB-65. Novamente, criou-se uma aplicação do *ICareHero* para gerar pseudoaleatoriamente os valores das variáveis que não são provenientes de

sensores (idade, confusão e BUN), receber as notificações do *ICareServer*, somar as pontuações de todas as variáveis presentes nesta especificação e exibir a avaliação do escore de acordo com o resultado obtido.

A Figura 5.9 demonstra os valores de saída referentes à execução da aplicação de monitoramento do escore CURB-65 no momento da coleta de dados dos sensores e demais variáveis. Na Figura 5.9, “*zephyrbloodpressure*” representa a variável da pressão sanguínea sistólica e “*zephyrrespiratory*” representa a frequência respiratória. No momento da coleta, os sensores registraram o que segue: “**84,0 mmHg**” de pressão sanguínea sistólica e “**35 rpm**” de frequência respiratória. Os valores pseudoaleatórios gerados para as variáveis de idade, BUN e confusão foram: “**69 anos**”, “**14 mg/dl**” e “**Não**”, respectivamente. Para os valores registrados, a pontuação das variáveis foram: pressão sanguínea sistólica (+1), frequência respiratória (+1), idade (+1), confusão (0) e BUN (0). Logo, o resultado obtido pelo somatório das pontuações foi “**3**”, cuja avaliação correspondente é “**22% de risco de mortalidade: internação hospitalar com UTI a ser considerada se o escore for de 4 ou 5**”, conforme descrita na Tabela 5.2.



```
Debugger Console →
"C:\Program Files\Java\jdk1.8.0_181\bin\java.exe" ...
Connected to the target VM, address: '127.0.0.1:52211', transport: 'socket'
ReliableSocket: new utils pool size = 3
null
zephyrbloodpressure - 84.0 mmHg - ObsRuleBPLT90 - Score: 1
zephyrrespiratory - 35.0 bpm - ObsRuleRRGOET30 - Score: 1
Age: 69 - Score: 1
BUN: 14mg/dL - Score: 0
Confusion: NO - Score: 0
----- Result -----
Score: 3
22% mortality
Inpatient admission with consideration for ICU admission with score of 4 or 5
```

**Figura 5.9:** Execução da aplicação de monitoramento para o escore CURB-65.

Para mais detalhes, os projetos contendo as aplicações do *ICareServer*, *ICarePatient* e *ICareHero*, bem como os vídeos com as demonstrações dos estudos de caso aqui descritos, estão disponíveis para consulta no seguinte endereço da Web: <<http://www.lsdj.ufma.br/projetos/mdd4clinicalscores>>.

## 5.3 Discussão

Na literatura, especificamente dentro da área de computação móvel, a automatização de escores clínicos é resolvida de forma limitada, ou seja, um software (geralmente, aplicação móvel) é desenvolvido para um tipo específico de escore. Diferentemente destas soluções, o *MDD4ClinicalScores* propõe uma nova abordagem para tratar este problema. Focando nisso, esta abordagem propõe: (1) desenvolver uma linguagem específica de domínio que fosse capaz de modelar diferentes especificações de escore clínico; e (2) através de técnicas de transformação de modelos, esta linguagem fosse capaz de gerar automaticamente componentes de software que pudessem ser aproveitados no contexto da computação ubíqua para monitorar pacientes remotamente.

Neste cenário, a abordagem do *MDD4ClinicalScores* ataca o problema de automatização de escores clínicos combinando técnicas de metamodelagem e regras de transformação, além do uso de um ambiente de suporte para geração e execução de aplicações mHealth focadas na validação e avaliação destas especificações. O metamodelo proposto (a linguagem *DSML4ClinicalScore*) foi definido usando o Ecore junto com o IDE do EMF, com base nas variáveis, regras para cálculo da pontuação e avaliações de diversas especificações de escores clínicos. O trabalho de [Aakre et al. 2017] foi a referência para definição do conjunto de escores clínicos que serviram de base para a construção da *DSML4ClinicalScore*. As regras de transformação foram implementadas com o auxílio da ferramenta do *Acceleo*, e indicam como os conceitos de um escore clínico descrito conforme o metamodelo proposto são traduzidos para uma ontologia descrita conforme a linguagem *Deklaer*, uma ontologia OWL que modela uma aplicação para IoT que utiliza sensores. Em seguida, essa ontologia recém-criada é submetida ao Ambiente de Execução do *MDD4ClinicalScores* para a geração e execução da aplicação mHealth de monitoramento baseada no escore clínico previamente modelado.

O *MDD4ClinicalScores* lida diretamente com duas questões que estão associadas à automatização de escores clínicos em ambientes de computação ubíqua: produtividade e complexidade. O aspecto da produtividade está estritamente ligado à forma como as atuais soluções computacionais reproduzidas na literatura resolvem este problema. Sempre que se quer desenvolver uma aplicação para um novo escore

clínico, todas as etapas de um processo de desenvolvimento de software convencional são repetidas, mesmo que esta especificação tenha características semelhantes a outras outrora desenvolvidas.

Com a linguagem *DSML4ClinicalScore*, temos um metamodelo que consegue abranger as características fundamentais dos principais escores clínicos. Logo, para implementar um novo software com determinado escore clínico, basta adaptar suas particularidades ao metamodelo da *DSML4ClinicalScore* e usufruir dos componentes de software dele resultantes. Assim, o *MDD4ClinicalScores* permite economizar o tempo despendido em algumas etapas do processo de desenvolvimento de software tradicional, como levantamento de requisitos, projeto do sistema e parte da implementação, quando o domínio da aplicação for escore clínico.

Por sua vez, o aspecto da complexidade está relacionada ao ambiente para o qual essas aplicações são desenvolvidas. Além do conhecimento específico da área da qual o escore clínico é derivado, um software típico para automação destes modelos requer conhecimento especializado de diversas áreas, como computação móvel, computação ubíqua, redes de sensores pessoais, entre outras. Além disso, o desenvolvedor da aplicação tem que lidar com dados provenientes de várias fontes (sensores, prontuários eletrônicos, entrevistas etc.) e em diferentes formatos.

O *MDD4ClinicalScores* provê uma infraestrutura de software que esconde do programador vários detalhes do ambiente onde estão inseridas as aplicações mHealth para escores clínicos, dentre os quais destacamos: (1) **hardware dos sensores**, com o M-Hub implementando *drivers* de sensores disponíveis no mercado para comunicação e troca de dados com estes dispositivos, além de permitir a atualização dessa configuração para inclusão de novos sensores; (2) **protocolos de comunicação sem fio**, com o M-Hub oferecendo comunicação sem fio rápida e otimizada para conexão com os objetos móveis presentes no ambiente, e com o núcleo do SDDL; (3) **processamento de fluxos de dados**, com o SDDL e M-Hub disponibilizando processamento CEP sobre fluxos de dados provenientes de sensores tanto nas extremidades quanto no núcleo do sistema; e (4) **particularidades dos escores clínicos**, com a *DSML4ClinicalScore* facilitando a tarefa do usuário em especificar características de diferentes escores clínicos através de uma linguagem de modelagem de alto nível.

Por conseguinte, cabe ao programador apenas implementar uma aplicação relativamente simples, que estabelece uma comunicação com o núcleo do SDDL, computa as pontuações fornecidas por ele e exibe a avaliação do escore clínico correspondente.

## 5.4 Limitações

Pela metodologia empregada no desenvolvimento da linguagem *DSML4ClinicalScore*, o *MDD4ClinicalScores* não garante que outras especificações de escore clínico, que não estejam dentro daquela relação anteriormente discutida na Subseção 4.2.1, possam ser devidamente modeladas por essa linguagem. Com relação ao Ambiente de Execução do *MDD4ClinicalScores*, as limitações se restringem na capacidade de processar somente dados provenientes de sensores e na edição manual da ontologia Deklaer para a inserção dos indivíduos que serão monitorados e dos modelos de sensores que serão utilizados no monitoramento. Variáveis de escores clínicos obtidas através de outras fontes, como o prontuário eletrônico do paciente ou entrevistas, não são lidas pela abordagem.

## 5.5 Considerações Finais

Este capítulo discutiu o processo de avaliação da abordagem do *MDD4ClinicalScores*. Esta avaliação consistiu da validação da linguagem *DSML4ClinicalScore* e da elaboração de dois estudos de casos que evidenciaram o uso da abordagem proposta para criação de aplicações mHealth para monitoramento remoto de pacientes baseado em escores clínicos.

A validação da linguagem *DSML4ClinicalScore* fundamentou-se na modelagem de oito casos concretos de diferentes escores clínicos usando o metamodelo aqui proposto. Estes oito escores clínicos foram escolhidos de tal forma que explorassem características diversas quanto às suas variáveis, regras para cálculo da pontuação e avaliações. A linguagem *DSML4ClinicalScore* mostrou boa expressividade e conseguiu modelar todos os aspectos das especificações eleitas. Os referidos modelos concretos estão disponíveis na Internet para consulta.

Para a elaboração dos estudos de caso, optou-se pela escolha de dois escores clínicos específicos: MEWS e CURB-65. Para cada uma destas especificações, este capítulo detalhou o procedimento adotado para construir o modelo concreto através da *DSML4ClinicalScore*, traduzir este modelo via transformação M2T para uma ontologia conforme a linguagem *Deklaer*, e gerar semiautomaticamente a aplicação mHealth de monitoramento para o respectivo escore clínico. Os mencionados estudos de caso ratificaram a viabilidade da proposta do *MDD4ClinicalScores*.

Para finalizar, este capítulo ressaltou como a abordagem do *MDD4ClinicalScores* se propõe a resolver as problemáticas apresentadas no Capítulo 1 desta dissertação. Depois, as limitações desta abordagem foram explicadas, tanto no que tange ao metamodelo da *DSML4ClinicalScore* quanto ao Ambiente de Execução da abordagem proposta.

## 6 Conclusões e Trabalhos Futuros

Sistemas de escores clínicos são ferramentas que auxiliam profissionais da saúde na prática de sua atividade clínica rotineira, sendo comumente usados para diagnosticar uma doença, estimar o risco de vida ou tomar uma decisão terapêutica. Nas últimas décadas, o uso destes mecanismos vem ganhando espaço dentro de ambientes hospitalares, principalmente no manejo de doenças crônicas. Dentro desse contexto, as pesquisas em torno desse tema acompanham esta tendência de forma sistemática, com o surgimento de novos achados empíricos para diferentes especialidades médicas.

As atuais abordagens na área de computação móvel e ubíqua desenvolvem um novo software (geralmente, uma aplicação móvel) para cada especificidade de escore. Estas abordagens apresentam limitações, pois toda vez que necessitar criar uma solução para um novo escore clínico, todas as etapas do processo de desenvolvimento do software serão retomadas, mesmo que a especificação em questão tenha características similares com outras anteriormente desenvolvidas, o que afeta a produtividade. Além disso, as particularidades do ambiente dentro do qual este tipo de sistema está inserido denotam um certo nível de complexidade, pois requerem conhecimento especializado dos programadores em áreas como redes de sensores pessoais, CEP, mHealth, dentre outros.

Este trabalho investigou uma nova abordagem para o desenvolvimento de software voltado para automatização de escores clínicos em ambientes de computação ubíqua, denominada de *MDD4ClinicalScores*. Esta abordagem, diferente das soluções mHealth descritas na literatura, é centrada na linguagem de modelagem específica de domínio *DSML4ClinicalScore*, que permite modelar as características fundamentais de diferentes especificações de escore clínico. Com base na especificação provida através da *DSML4ClinicalScore*, o *MDD4ClinicalScores* implementa uma transformação M2T para produzir uma ontologia OWL baseada na linguagem *Deklaer*. Esta ontologia *Deklaer* recém-criada é então submetida ao Ambiente de Execução do *MDD4ClinicalScores* para geração semiautomática e execução da aplicação mHealth de monitoramento de pacientes focada na avaliação do escore clínico previamente

especificado. Este trabalho avaliou a abordagem proposta por meio de um processo de validação da *DSML4ClinicalScore* e de dois estudos de caso que descrevem o uso do *MDD4ClinicalScores*.

O *MDD4ClinicalScores* teve dois objetivos principais: aumentar a produtividade no desenvolvimento de software dentro do contexto de computação ubíqua e escores clínicos e tratar a complexidade inerente a tal ambiente. No tocante à produtividade, nossa abordagem contribuiu com o provimento da *DSML4ClinicalScore*, dado que permite especificar através de uma linguagem específica de domínio escores clínicos e, combinado com a aplicação de regras de transformação, gerar semiautomaticamente uma aplicação mHealth para determinado escore. Com relação à complexidade, nossa abordagem contribuiu com o fornecimento de uma infraestrutura de software que esconde do programador da aplicação final diversos detalhes do ambiente para o qual ela se destina, que envolve um ambiente distribuído e ubíquo baseado na aquisição de dados de sensores diversos.

As principais contribuições deste trabalho são:

- A investigação do estado da arte em soluções mHealth para a resolução de escores clínicos para os mais diversos ambientes. Através desta investigação, ficaram evidenciadas as limitações destas abordagens em relação a aspectos como produtividade e complexidade, o que motivou o desenvolvimento do *MDD4ClinicalScores*.
- O desenvolvimento de uma abordagem que permite modelar e gerar semiautomaticamente uma aplicação mHealth para um escore clínico, capaz de monitorar remotamente um paciente através de sensores e com base na avaliação deste escore. O *MDD4ClinicalScores* provê, através dos componentes de Geração Automática de Código e Ambiente de Execução, um conjunto de funcionalidades que permite modelar diferentes especificações de escore clínico, além de gerar e executar aplicações mHealth de monitoramento desses escores. Esta abordagem facilita o desenvolvimento de software para o domínio de escores clínicos, fornecendo o suporte do qual os programadores necessitam para este tipo de aplicação.
- A disponibilização de uma linguagem específica de domínio para especificação de escores clínicos de modo geral. A *DSML4ClinicalScore* provê um metamodelo

que engloba as principais características existentes em uma especificação de escore clínico: variáveis, regras para cálculo da pontuação e avaliações. Para desenvolver uma aplicação para um escore clínico específico, basta adaptar as particularidades desta especificação ao metamodelo da *DSML4ClinicalScore* e usufruir dos componentes de software dele resultantes. Isso aumenta a produtividade no processo de desenvolvimento de software neste contexto, pois algumas etapas da metodologia tradicional são desconsideradas.

- A implementação de uma infraestrutura de software capaz de gerar semiautomaticamente e executar uma aplicação para IoT focada no monitoramento remoto de pacientes através de sensores e avaliação de um escore clínico. O *MDD4ClinicalScores* fornece um ambiente de execução dentro do qual está disponível todo o suporte necessário para a criação da aplicação final. Isso diminui a complexidade inerente ao contexto sobre o qual este tipo de sistema é implantado, cabendo ao programador desenvolver uma aplicação mais simples daquela que seria necessária se ele não dispusesse desse aparato tecnológico.

A partir deste trabalho inicial, identificamos algumas possibilidades de trabalhos futuros que poderiam ser desenvolvidos, tais como:

- O metamodelo da *DSML4ClinicalScore* foi implementado usando o Ecore junto com o IDE do EMF. O EMF possibilita desenvolver uma ferramenta gráfica de alto nível a partir de um metamodelo Ecore. Uma ferramenta deste tipo pode auxiliar os profissionais da saúde na especificação de diversos escores clínicos e, conseqüentemente, tornar-se o ponto de partida para o desenvolvimento de diferentes aplicações para IoT. A integração desta ferramenta gráfica ao *MDD4ClinicalScores* pode ser uma alternativa viável para apoiar a prática clínica dentro de ambientes hospitalares.
- O Ambiente de Execução do *MDD4ClinicalScores* foi construído com base no trabalho de [Pinheiro et al. 2018], que descreve um framework orientado por ontologias que semiautomatiza a geração de aplicações para IoT com uso de sensores. Atualmente, esta infraestrutura de software não gerencia informações relacionadas às variáveis de escore que não são provenientes de sensores, tais

como dados de prontuário eletrônico, resultados de outros escores clínicos e entrevistas. Dessa forma, a proposta do *MDD4ClinicalScores* pode ser melhorada com a implementação de interfaces de comunicação com sistemas de prontuário eletrônico e escores clínicos, objetivando integrar automaticamente os dados provenientes dessas fontes ao conjunto de variáveis de um escore clínico. Da mesma forma, há a possibilidade de desenvolver formas alternativas para incorporar dados manuais, cuja obtenção ocorre por intermédio de entrevista com paciente ou médico responsável, ao escopo do *MDD4ClinicalScores*.

## Referências Bibliográficas

AAKRE, C.; DZIADZKO, M.; HERASEVICH, V. Towards automated calculation of evidence-based clinical scores. *World Journal of Methodology*, Baishideng Publishing Group, v. 7, n. 1, Março 2017. ISSN 2222-0682.

AAKRE, C. et al. Automating clinical score calculation within the electronic health record. *Applied Clinical Informatics*, Thieme Medical, v. 8, n. 2, Abril 2017. ISSN 1869-0327.

ADAMS, Z. et al. Mobile devices for the remote acquisition of physiological and behavioral biomarkers in psychiatric clinical research. *Journal of Psychiatric Research*, Elsevier, v. 85, Fevereiro 2017. ISSN 0022-3956.

AMINIAN, A. et al. Who should get extended thromboprophylaxis after bariatric surgery?: A risk assessment tool to guide indications for post-discharge pharmacoprophylaxis. *Annals of Surgery*, Wolters Kluwer Health, v. 265, n. 1, Janeiro 2017a. ISSN 0003-4932.

AMINIAN, A. et al. Individualized metabolic surgery score: Procedure selection based on diabetes severity. *Annals of Surgery*, Wolters Kluwer Health, v. 266, n. 4, Outubro 2017c. ISSN 0003-4932.

AMINIAN, A. et al. Development of a sleeve gastrectomy risk calculator. *Surgery for Obesity and Related Diseases*, Elsevier, v. 11, n. 4, Agosto 2015. ISSN 1550-7289.

AMINIAN, A. et al. Bariatric surgery decision-making calculator: A novel mobile app for evidence-based clinical practice. *Surgery for Obesity and Related Diseases*, Cleveland Clinic, v. 13, n. 10, Outubro 2017b. ISSN 0891-1150.

ANTONIOU, G.; HARMELEN, F. van. Web ontology language: Owl. In: *International Handbooks on Information System: Handbook on Ontologies*. [S.l.]: Springer, 2004. cap. 4. ISBN 978-3-662-11957-0.

BABAU, J. et al. *Model-Driven Engineering for Distributed Real-Time Systems: MARTE Modeling, Model Transformations and their Usages*. 1st. ed. [S.l.]: ISTE Wiley, 2010. ISBN 978-1-84821-115-5.

BRAMBILLA, M.; CABOT, J.; WIMMER, M. *Model-Driven Software Engineering in Practice*. 1st. ed. [S.l.]: Morgan Claypool, 2012. ISBN 978-1-608-45883-7.

CAFEZEIRO, I. et al. Specifying ubiquitous systems through the algebra of contextualized ontologies. *The Knowledge Engineering Review*, Cambridge University Press, v. 29, n. 2, 2014. ISSN 1469-8005.

CAVARARO, R. *Pesquisa Nacional da Saúde 2013: Percepção do estado da saúde, estilos de vida e doenças crônicas*. [S.l.], 2014.

COOK, K.; MODENA, B.; SIMON, R. Improvement in asthma control using a minimally burdensome and proactive smartphone. *The Journal of Allergy and Clinical Immunology*, American Academy of Allergy, Asthma & Immunology, v. 13, n. 10, Outubro 2016. ISSN 2213-2198.

DAVID, L. et al. A dds-based middleware for scalable tracking, communication and collaboration of mobile nodes. *Journal of Internet Services and Applications*, Springer, v. 4, n. 16, Julho 2013. ISSN 1867-4828.

ECLIPSE COMMUNITY. *Package org.eclipse.emf.ecore Description*. [S.l.], 2019. Disponível em <<http://download.eclipse.org/modeling/emf/emf/javadoc/2.9.0/org/eclipse/emf/ecore/package-summary.html>>. Acessado em 22/01/2019.

ENDLER, M.; SILVA, F. J. da Silva e. Past, present and future of the contextnet iomt middleware. *Open Journal of Internet of Things*, RonPub UG, v. 4, n. 1, Julho 2018. ISSN 2364-7108.

FALK, G.; FAHEY, T. Clinical prediction rules. *British Medical Journal*, BMJ, v. 339, n. 1, Agosto 2009. ISSN 1756-1833.

FOWLER, M. *Domain-Specific Languages*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 978-0-321-71294-3.

Fülöp, L. et al. Survey on complex event processing and predictive analytics. In: *Proceedings of the Fifth Balkan Conference in Informatics*. Novi Sad, Serbia: [s.n.], 2010.

GAGE, B. et al. Validation of clinical classification schemes for predicting stroke: results from the national registry of atrial fibrillation. *Journal of the American Medical Association*, American Medical Association, v. 285, n. 22, Junho 2001. ISSN 1538-3598.

GERDS, T.; CAI, T.; SCHUMACHER, M. The performance of risk prediction models. *Biometrical Journal*, Wiley, v. 50, n. 4, Junho 2008. ISSN 1521-4036.

GOETZ, C. et al. Movement disorder society-sponsored revision of the unified parkinson's disease rating scale (mds-updrs): Scale presentation and clinimetric testing results. *Official Journal of Movement Disorder Society*, John Wiley & Sons, v. 23, n. 15, Novembro 2008. ISSN 1531-8257.

GRONBACK, R. C. *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. 1st. ed. Boston, MA, USA: Addison-Wesley Professional, 2009. ISBN 978-0-321-53407-1.

H. IRAWAN. *Ecore*. [S.l.], 2010. Disponível em <<https://wiki.eclipse.org/ecore>>. Acessado em 22/01/2019.

ISTEPANIAN, R.; LAXMINARAYAN, S.; PATTICHIS, C. *M-Health: Emerging Mobile Health Systems*. 1st. ed. [S.l.]: Springer, 2006. ISBN 978-0387-26558-2.

JANKOVIC, J. Parkinson's disease: Clinical features and diagnosis. *Journal of Neurology, Neurosurgery, and Psychiatry*, BMJ, v. 79, n. 4, Abril 2008. ISSN 0022-3050.

JEMAL, A. et al. Global cancer statistics. *CA: A Cancer Journal for Clinicians*, American Cancer Society, v. 61, n. 2, Março-Abril 2011. ISSN 1542-4863.

JENKINSON, C. et al. The parkinson's disease questionnaire (pdq39): development and validation of a parkinson's disease summary index score. *Age and Ageing*, Oxford Academic, v. 26, n. 5, Setembro 1997. ISSN 0002-0729.

KLEPPE, A. *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2008. ISBN 978-0-321-55345-4.

KLEPPE, A.; WARMER, J.; BAST, W. *MDA Explained. The Model-Driven Architecture: Practice and Promise*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2003. ISBN 978-0-321-19442-8.

LEE, Y.; BANG, H.; KIM, D. How to establish clinical prediction models. *Endocrinology and Metabolism*, Korean Endocrine Society, v. 31, n. 1, Janeiro 2016. ISSN 2093-5978.

LIM, W. et al. Defining community acquired pneumonia severity on presentation to hospital: an international derivation and validation study. *Thorax Journal*, BMJ, v. 58, n. 5, Maio 2003. ISSN 1468-3296.

LO, G. et al. Evaluation of pretest clinical score (4 ts) for the diagnosis of heparin-induced thrombocytopenia in two clinical settings. *Journal of Thrombosis and Haemostasis*, International Society on Thrombosis and Haemostasis, v. 4, 2006. ISSN 1538-7836.

LUCKHAM, D. The power of events: An introduction to complex event processing in distributed enterprise systems. In: \_\_\_\_\_. *Lectures Notes in Computer Science: Rule Representation, Interchange and Reasoning on the Web*. [S.l.]: Springer, 2008. ISBN 978-3-540-88807-9.

MACHADO, A. Doenças crônicas. *Jornal Brasileiro de Patologia e Medicina Laboratorial*, SciELO, v. 42, n. 1, Fevereiro 2006. ISSN 1678-4774.

MATEEN, A.; TABASSUM, A. A framework for model driven transformation engineering towards software architecture and performance. *International Journal of Computer Applications*, Foundation of Computer Science, v. 143, n. 8, Junho 2016. ISSN 0975-8887.

MDCALC. *MD+Calc - Medical calculators, equations, algorithms, and scores*. [S.l.], 2005. Disponível em <<https://www.mdcalc.com/>>. Acessado em 24/01/2019.

MERNIK, M.; HEERING, J.; SLOANE, A. When and how to develop domain-specific languages. *ACM Computing Surveys*, ACM, v. 37, n. 4, Dezembro 2005. ISSN 1557-7341.

MINISTÉRIO DA SAUDE. *Asma: o que é, causas, sintomas, tratamento, diagnóstico e prevenção*. [S.l.], 2019. Disponível em <<http://portalms.saude.gov.br/saude-de-a-z/asma>>. Acessado em 17/05/2019.

MOTTET, N. et al. *Guidelines on prostate cancer*. [S.l.], 2015. Available in: <<https://uroweb.org/wp-content/uploads/EAU-Guidelines-Prostate-Cancer-2015-v2.pdf>>.

NIHCE. *Parkinson's disease: diagnosis and management in primary and secondary care: National cost-impact report*. [S.l.], 2006. NICE clinical guideline no. 35.

OBEROI, S. Introduction to complex event processing and data streams. In: \_\_\_\_\_. *SOA Sys-Con Magazine*. [S.l.]: Sys-Con, 2007.

OBJECT MANAGEMENT GROUP. *About OMG*. [S.l.], 2019. Disponível em <<https://www.omg.org/about/>>. Acessado em 10/01/2019.

OLLA, P.; SHIMSKEY, C. mhealth taxonomy: a literature survey of mobile health applications. *Health Technology*, Springer, v. 4, n. 4, Abril 2015. ISSN 2190-7196.

OLLA, P.; TAN, J. *Mobile Health Solutions for Biomedical Applications*. 1st. ed. [S.l.]: IGI Global, 2009. ISBN 978-1-60566-332-6.

OMG, O. M. G. *Data Distribution Service for Real-time Systems Specifications*. [S.l.], 2012. Available in: <[www.omg.org/spec/](http://www.omg.org/spec/)>.

PEREIRA-AZEVEDO, N. et al. Rotterdam prostate cancer risk calculator: Development and usability testing of the mobile phone app. *JMIR Cancer*, JMIR Publications, v. 3, n. 1, Janeiro 2017. ISSN 2369-1999.

PINHEIRO, V. et al. Deklaer: An ontology-driven framework for generating iot applications using contextnet. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. Natal, Brasil: [s.n.], 2018.

PISTERS, R. et al. A novel user-friendly score (has-bleed) to assess 1-year risk of major bleeding in patients with atrial fibrillation: the euro heart survey. *Chest Journal*, Elsevier, v. 138, n. 5, Março 2010. ISSN 0012-3692.

RAPSANG, A.; SHYAM, D. Scoring systems in the intensive care unit: A compendium. *Indian Journal of Critical Care Medicine*, Medknow Publications, v. 18, n. 4, Abril 2014. ISSN 0972-5229.

RIOS, L. et al. The mobile hub concept: Enabling application for the internet of mobile things. In: *12th IEEE Workshop on Managing Ubiquitous Communications and Services (MUCS), IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM)*. St. Louis, MO, USA: [s.n.], 2015.

ROOBOL, M. J. et al. Importance of prostate volume in the european randomised study of screening for prostate cancer (erspc) risk calculators: results from the prostate biopsy collaborative group. *World Journal of Urology*, Springer, v. 30, n. 2, Abril 2012. ISSN 0724-4983.

ROXO, J. P. F. et al. Validação do teste de controle da asma em português para uso no brasil. *Jornal Brasileiro de Pneumologia*, Sociedade Brasileira de Pneumologia e Tisiologia, v. 36, n. 2, Março-Abril 2016. ISSN 1806-3713.

SAHA, G. Web ontology language (owl) and semantic web. *ACM Ubiquity*, ACM, v. 8, n. 35, Setembro 2007. ISSN 1530-2180.

SAMA, P. et al. An evaluation of mobile health application tools. *JMIR MHealth and UHealth*, JMIR Publications, v. 2, n. 2, Abril-Junho 2014. ISSN 2291-5222.

SAÚDE, M. da. *Diretrizes para o cuidado das pessoas com doenças crônicas nas redes de atenção à saúde e nas linhas de cuidado prioritárias*. 1st. ed. Brasília, DF, Brasil: Editora MS, 2013. ISBN 978-85-334-2018-2.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, Elsevier, v. 61, Janeiro 2015. ISSN 0893-6080.

SCHMIDT, D. Guest editor's introduction: Model-driven engineering. *IEEE Computer*, IEEE Computer Society, v. 39, n. 2, Fevereiro 2006. ISSN 0018-9162.

SCHRÖDER, F. et al. Screening and prostate cancer mortality: results of the european randomised study of screening for prostate cancer (erspc) at 13 years of follow-up. *The Lancet*, Elsevier, v. 384, n. 9959, Dezembro 2014. ISSN 0140-6736.

SCHURINK, C. et al. Clinical pulmonary infection score for ventilator-associated pneumonia: accuracy and inter-observer variability. *Journal of Intensive Care Medicine*, SAGE Journals, v. 30, n. 2, Fevereiro 2004. ISSN 1525-1489.

SILVA, B. et al. Mobile health: A review of current state in 2015. *Journal of Biomedical Informatics*, Elsevier, v. 56, n. 1, Agosto 2015. ISSN 1532-0464.

STAMATE, C. et al. The cloudupdrs app: A medical device for the clinical assessment of parkinson's disease. *Pervasive and Mobile Computing*, Elsevier, v. 43, n. 1, Janeiro 2018. ISSN 1574-1192.

STEYERBERG, E. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. 1st. ed. New York, USA: Springer, 2009. ISBN 978-0-387-77243-1.

SUBBE, C. et al. Validation of a modified early warning score in medical admissions. *Monthly Journal of the Association of Physicians*, Oxford University Press, v. 94, n. 10, Outubro 2001. ISSN 1460-2725.

TERRY, N. P. Mobile health: Assessing the barriers. *Chest*, Elsevier, v. 147, n. 5, Maio 2015. ISSN 0012-3692.

THOMPSON, G. Clinical scoring systems in the management of suspected appendicitis in children. In: \_\_\_\_\_. *Appendicitis – A Collection of Essays from Around the World*. [S.l.]: InTech, 2012. cap. 4. ISBN 978-953-307-814-4.

VOELTER, M. et al. *DSL Engineering: Designing, Implementing and Using Domain-Specific Languages*. 1st. ed. [S.l.]: dslbook.org, 2010. Disponível em: <<http://dslbook.org>>.

WELLS, P. et al. Excluding pulmonary embolism at the bedside without diagnostic imaging: management of patients with suspected pulmonary embolism presenting to the emergency department by using a simple clinical model and d-dimer. *Annals of Internal Medicine*, American College of Physicians, v. 135, n. 2, Julho 2001. ISSN 4303-2966.

WIVIOTT, S. et al. Application of the thrombolysis in myocardial infarction risk index in non-st-segment elevation myocardial infarction: evaluation of patients in the national registry of myocardial infarction. *Journal of the American College of Cardiology*, Elsevier, v. 47, n. 8, Março 2006. ISSN 0735-1097.

YIN, R. *Case Study Research: Design and Methods*. 5th. ed. Thousand Oaks, CA: Sage, 2014. ISBN 978-1-84821-115-5.