

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA EM SAÚDE**

**DIEGO GARCIA**

**SUORTE A DECISÃO COM INCORPORAÇÃO DE DIRETRIZES CLÍNICAS AO  
REGISTRO ELETRÔNICO EM SAÚDE BASEADO EM ARQUÉTIPOS OPENEHR**

**CURITIBA**

**2014**

**DIEGO GARCIA**

**SUORTE A DECISÃO COM INCORPORAÇÃO DE DIRETRIZES CLÍNICAS AO  
REGISTRO ELETRÔNICO EM SAÚDE BASEADO EM ARQUÉTIPOS OPENEHR**

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologia em Saúde, da Escola Politécnica, da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de mestre em Tecnologia em Saúde.

Orientador: Prof.<sup>a</sup> Dr.<sup>a</sup>. Claudia Maria Cabral Moro Barra

**CURITIBA**

**2014**

Dados da Catalogação na Publicação  
Pontifícia Universidade Católica do Paraná  
Sistema Integrado de Bibliotecas – SIBI/PUCPR  
Biblioteca Central

G216s  
2014 Garcia, Diego  
Suporte a decisão com incorporação de diretrizes clínicas ao registro eletrônico em saúde baseado em arquétipos openehr / Diego Garcia; orientadora, Cláudia Maria Cabral Moro Barra. -- 2014  
144 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2014  
Bibliografia: f. 124-130

1. Nefrologia. 2. Tecnologia médica. 3. Armazenamento de dados.  
4. Documentos eletrônicos. I. Barra, Cláudia Maria Cabral Moro. II. Pontifícia Universidade Católica do Paraná Pós-Graduação em Tecnologia da Saúde.  
III. Título.

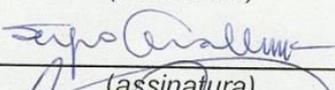
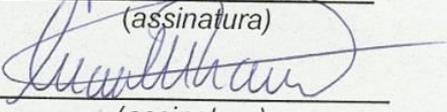
CDD 20. ed. – 616.61

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO  
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA EM SAÚDE

DEFESA DE DISSERTAÇÃO Nº 200

ÁREA DE CONCENTRAÇÃO: TECNOLOGIA EM SAÚDE

Aos 11 dias do mês de novembro de 2014, no auditório Irmão Albano, realizou-se a sessão pública de Defesa da Dissertação: **“Suporte à decisão com incorporação de diretrizes clínicas ao registro eletrônico em saúde baseado em arquétipos openehr”**, apresentada pelo aluno **Diego Garcia**, sob orientação da Prof<sup>a</sup>. Dr<sup>a</sup>. **Claudia Maria Cabral Moro Barra** como requisito parcial para a obtenção do título de **Mestre em Tecnologia em Saúde**, perante uma Banca Examinadora composta pelos seguintes membros:

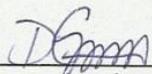
Prof. Dr. Claudia Maria Cabral Moro Barra, PUCPR (Orientador e presidente)	 _____ (assinatura)	<u>APROVADA</u> (Aprov/Reprov.)
Prof. Dr. Sérgio Ossamu Ioshii PUCPR (Examinador)	 _____ (assinatura)	<u>Aprovado</u> (Aprov/Reprov.)
Prof. Dr. Ivan Luiz Marques Ricarte, UNICAMP (Examinador)	 _____ (assinatura)	<u>Aprovado</u> (Aprov/Reprov.)

Início: 9:00 Término: 11:10

Conforme as normas regimentais do PPGTS e da PUCPR, o trabalho apresentado foi considerado APROVADO (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora.

Observações: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

O aluno está ciente que a homologação deste resultado está condicionada: (I) ao cumprimento integral das solicitações da Banca Examinadora, que determina um prazo de 30 dias para o cumprimento dos requisitos; (II) entrega da dissertação em conformidade com as normas especificadas no Regulamento do PPGTS/PUCPR; (III) entrega da documentação necessária para elaboração do Diploma.

ALUNO: Diego Garcia   
\_\_\_\_\_  
(assinatura)

Prof<sup>a</sup>. Dr<sup>a</sup>. Claudia Maria Cabral Moro Barra,  
Coordenadora do PPGTS PUCPR



À minha mãe, Maria das Graças pelo exemplo de dedicação e força, à minha orientadora, Prof.<sup>a</sup> Dr.<sup>a</sup> Claudia Moro pela confiança e empenho e a todos os profissionais e pacientes que possam ser, direta ou indiretamente, beneficiados pela aplicação dos resultados desta pesquisa, objetivo principal de todo o esforço empregado.

## AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me permitido atingir mais este objetivo, provendo a força necessária nos mais variados momentos durante este percurso.

À minha mãe, pela paciência de sempre e pela compreensão nos momentos de tensão, por ter me dado todas as condições necessárias para chegar até aqui, por vezes abdicando de seu próprio conforto e por todos os momentos de preocupação que só ela deve entender. Por ter sempre me apoiado em minhas decisões e incentivado a buscar meus sonhos e atingir meus objetivos.

À minha orientadora, Prof.<sup>a</sup> Dr.<sup>a</sup> Claudia Moro, exemplo de profissional, pela confiança em mim depositada, pela dedicação e atenção com seus alunos e orientandos, por todos os meses de convivência e pelas oportunidades a mim dadas, permitindo que eu fosse mais longe para alcançar meus objetivos. Palavras não são suficientes para expressar toda minha gratidão e admiração por esta pessoa.

À minha orientadora de graduação, Prof.<sup>a</sup> Dr.<sup>a</sup> Priscyla Simões, pois foi por meio dela que conheci esta área e tive contato com minha atual orientadora.

Ao grupo de pesquisa em arquétipos, Dandara, Daiane, Hugo, Alex e em especial, ao Paulo, com quem tive o prazer de trabalhar no início desta pesquisa e à Lilian, exemplo de determinação e perseverança, com quem aprendi muito e dividi muitas manhãs e tardes no LAIS.

À importante colaboração da Betha Sistemas e meus gestores, por permitirem me ausentar das minhas atividades, sempre que necessário, para o desenvolvimento desta pesquisa. À minha equipe, Vertical Saúde, pelo companheirismo do dia-a-dia.

Ao corpo docente que me proporcionou conhecimento e crescimento durante este projeto, principalmente à Prof.<sup>a</sup> Dr.<sup>a</sup> Deborah Carvalho, pela experiência compartilhada, contribuições e amizade.

Aos colegas de mestrado que me acompanharam nesta caminhada, mesmo que por pouco tempo, Janaína, Simone, Elaine, Andréia, Denilsen, Carina, Lucas, Mariane, Roni, Rebeca e a todos os demais colegas que compartilharam um pouco do seu tempo.

À Izelde, meu primeiro contato nas manhãs nas quais eu estava presente na Universidade, por todo o auxílio prestado e o bom humor de sempre.

À Pontifícia Universidade Católica do Paraná, pela bolsa de estudos concedida.

Aos meus familiares que sempre me apoiaram e me incentivaram.

A todos que participaram direta ou indiretamente desta caminhada e fizeram dela um caminho melhor.

*Não se mantenha sempre na via pública,  
indo somente onde os outros foram [...].*

*Deixe a trilha batida, ocasionalmente, e  
mergulhe na floresta.*

*Toda vez que você fizer isso, terá certeza de  
encontrar algo que você nunca viu antes.*

*Claro que será algo pequeno, mas não o  
ignore.*

*Siga-o, explore tudo a sua volta: uma  
descoberta leva a outra, e antes que você  
perceba você terá algo em que vale a pena  
pensar para ocupar sua mente.*

*Todas as descobertas realmente grandes  
são resultados do pensamento.*

*(Alexander Graham Bell)*

## RESUMO

A integração entre diferentes sistemas de Registro Eletrônico em Saúde (RES), que contêm dados de um indivíduo, é fundamental para o acompanhamento de saúde com qualidade, e deve ser baseada em padrões de conteúdo e informações específicos. Arquétipos da openEHR podem ser utilizados para essa finalidade, pois além de favorecer a interoperabilidade, facilitam a incorporação no RES dos dados e dos relacionamentos entre eles especificados em diretrizes clínicas. Contudo, nos trabalhos descritos na literatura, as diretrizes clínicas não estão completamente integradas ao RES. Considerando o aumento na incidência da Doença Renal Crônica (DRC) no Brasil e que a utilização de Sistemas de Apoio a Decisão auxilia no diagnóstico e facilita as recomendações baseadas em diretrizes, a prevenção da DRC é uma das áreas clínicas em que pode ser beneficiada pela integração de diretrizes ao RES. **Objetivo:** integrar diretrizes clínicas ao RES baseado na arquitetura de arquétipos openEHR, fornecendo suporte a decisão a partir das regras definidas nas diretrizes clínicas. **Método:** pesquisa exploratória, aplicada e com desenvolvimento, dividida em três fases: concepção do modelo de integração, implementação dos componentes do sistema e integração do motor de inferência para execução das regras de decisão. A validação do funcionamento do RES foi realizada por meio de arquétipos e regras baseados em uma diretriz para a prevenção da DRC. **Resultados:** foi proposto um modelo de desenvolvimento de RES baseado em arquétipo que permite o suporte a decisão por meio de regras extraídas de diretrizes clínicas e especificadas na Linguagem de Definição de Diretrizes (GDL – *Guideline Definition Language*), utilizando repositórios para os dados e serviços RESTful para acesso a estes, integrando o motor de inferência Drools Expert para a execução das regras de decisão. Foi implementada parcialmente a especificação do Modelo de Referência (RM – *Reference Model*) openEHR para permitir o desenvolvimento da interface para a coleta dos dados. Foram desenvolvidos os componentes do modelo de RES proposto, demonstrando a integração entre o RES baseado em arquétipos e diretrizes clínicas da prevenção de DRC especificadas em GDL. A pesquisa resultou também na definição de uma forma de armazenamento dos dados estruturados em arquétipos. **Conclusão:** o modelo de desenvolvimento proposto e implementado permitiu com sucesso a integração de regras de decisão extraídas de diretrizes clínicas para a prevenção de DRC em um RES baseado em arquétipos. Por ter sido desenvolvido com base na especificação do RM openEHR, podem ser integradas diretrizes de outras especialidades e com outros focos. O RES elaborado demonstrou ser genérico considerando-se que também foram realizados testes com diversos arquétipos não relacionados à DRC, e o RES foi capaz de processá-los adequadamente.

**Palavras-chave:** Arquétipos. Integração de Sistemas. Registros Eletrônicos de Saúde. Diretrizes Clínicas. Sistemas de Suporte a Decisão Clínica.

**Descritores:** Integração de Sistemas. Registros Eletrônicos de Saúde. Diretriz para a Prática Clínica. Sistemas de Apoio a Decisões Clínicas.

## ABSTRACT

The integration between different Electronic Health Record (EHR) systems, containing data from a subject, is the key to healthcare monitoring with quality and should be based on specific content and information standards. OpenEHR archetypes can be used for this purpose because in addition to support interoperability, they facilitate the integration, to the EHR, of data and relationship between them specified in clinical practice guidelines. However, in the work described in the literature, clinical practice guidelines are not fully integrated to the EHR. Considering the increase in the incidence of Chronic Kidney Disease (CKD) in Brazil and that the use of Clinical Decision Support Systems assists in the diagnosis and facilitates the recommendations based on clinical practice guidelines, CKD prevention is one of the clinical areas which can be benefited by the integration of clinical practice guidelines to the EHR. **Objective:** integrate clinical practice guidelines to the EHR based on openEHR archetypes architecture, providing decision support from rules defined in clinical practice guidelines. **Method:** exploratory, applied research and with development, divided in three phases: conception of the integration model, implementation of system components and integration of the inference engine to execution of the decision rules. The validation of EHR operation was performed by use of archetypes and rules based on a clinical practice guideline to CKD prevention. **Results:** it was proposed a development model to archetype-based EHR that allows decision support by the use of rules extracted from clinical practice guidelines and specified in Guideline Definition Language (GDL), using repositories for data and RESTful services to access them, integrating the inference engine Drools Expert for executing the decision rules. It was partially implemented the openEHR Reference Model (RM) specification to allow the development of the interface to data collect. It was developed the components of the proposed EHR model, demonstrating the integration between the archetype-based EHR and clinical practice guidelines to CKD prevention specified in GDL. The research resulted also in the definition of a storage form of data structured in archetypes. **Conclusion:** the proposed and implemented development model successfully has allowed the integration of decision rules extracted from clinical practice guidelines to CKD prevention in an archetype-based EHR. Since it was developed based in the openEHR RM specification, clinical practice guidelines from other specialties and with targets other than prevention, can be integrated to it. The elaborated EHR demonstrated to be generic considering that was also performed tests with various archetypes not related to CKD and the EHR was able to process them properly.

**Keywords:** Archetypes. Systems Integration. Electronic Health Records. Clinical Practice Guidelines. Clinical Decision Support Systems.

**Descriptors:** Systems Integration. Electronic Health Records. Clinical Practice Guidelines. Clinical Decision Support Systems.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Hierarquia dos componentes do modelo de referência da arquitetura openEHR.....	29
Figura 2 - Tabela de decisão em planilha eletrônica .....	38
Figura 3 - Fluxograma das Diretrizes Clínicas para o Cuidado ao Paciente com Doença Renal Crônica - DRC no Sistema Único de Saúde .....	41
Figura 4 - Evolução do projeto .....	44
Figura 5 - Método proposto para integração .....	46
Figura 6 - Modelo de desenvolvimento proposto.....	47
Figura 7 - Tecnologias empregadas no desenvolvimento .....	48
Figura 8 - Estrutura de pacotes da especificação openEHR.....	51
Figura 9 - Pacotes do modelo de referência implementado .....	52
Figura 10 – Diagrama de atividades do processamento do arquétipo para geração de interface .....	54
Figura 11 - Execução das regras no Drools Expert.....	64
Figura 12 - Resultado principal da pesquisa .....	66
Figura 13 - Modelo de arquitetura do RES desenvolvido .....	86
Figura 14 - Diagrama ER do repositório de arquétipos .....	87
Figura 15 - Diagrama ER do repositório de regras.....	89
Figura 16 - Listagem dos arquétipos disponíveis .....	99
Figura 17 - Interface gerada para o arquétipo openEHR-DEMOGRAPHIC-PERSON.person-patient.v1.....	100
Figura 18 - Interface gerada para o arquétipo openEHR-DEMOGRAPHIC-PERSON.person-patient.v1.....	101
Figura 19 - Interface gerada para o arquétipo openEHR-EHR-OBSERVATION.blood_pressure.v1 .....	102
Figura 20 - Indicação da existência de regras para o arquétipo.....	103
Figura 21 - Indicação de dados adicionais gerados .....	103
Figura 22 - Diagrama de classes: mapeamento entre arquétipos e banco de dados .....	105
Figura 23 - Diagrama ER do repositório de RES.....	106
Figura 24 - Dados de arquétipos adicionais.....	109

Figura 25 - Trecho retirado das Diretrizes Clínicas para o cuidado ao paciente com Doença Renal Crônica - DRC no Sistema Único de Saúde .....	110
Figura 26 - Regra para diagnóstico de diabetes na ferramenta GDL Tools .....	111
Figura 27 - Mapeamento entre regras em GDL e arquétipos .....	113
Figura 28 – Exemplo da execução de regras no Drools Expert .....	115
Quadro 1 – Fases da pesquisa .....	42
Quadro 2 - Componentes desenvolvidos .....	49
Quadro 3 - Arquétipos utilizados para testes do sistema desenvolvido .....	60
Quadro 4 - Modelos de Informação implementados na pesquisa .....	67
Quadro 5 - Pacotes da especificação openEHR, implementados na pesquisa.....	68
Quadro 6 - Classes implementadas do pacote rm.ehr .....	68
Quadro 7 - Classes implementadas do pacote rm.composition .....	69
Quadro 8 - Classe implementada do pacote rm.composition.content .....	69
Quadro 9 - Classes implementadas do pacote rm.composition.content.entry .....	70
Quadro 10 - Classes implementadas do pacote rm.demographic.....	70
Quadro 11 - Classes implementadas do pacote rm.common.archetyped .....	71
Quadro 12 - Classes implementadas do pacote rm.common.generic.....	72
Quadro 13 - Classes implementadas do pacote rm.common.directory .....	73
Quadro 14 - Classes implementadas do pacote rm.common.change_control .....	73
Quadro 15 - Classes implementadas do pacote rm.common.resource .....	74
Quadro 16 - Classe implementada do pacote rm.data_structures .....	75
Quadro 17 - Classes implementadas do pacote rm.data_structures.item_structure .	75
Quadro 18 - Classes implementadas do pacote rm.data_structures.representation .	75
Quadro 19 - Classes implementadas do pacote rm.data_structures.history .....	76
Quadro 20 - Classes implementadas do pacote rm.data_types.basic.....	77
Quadro 21 - Classes implementadas do pacote rm.data_types.text .....	77
Quadro 22 - Classes implementadas do pacote rm.data_types.quantity .....	78
Quadro 23 - Classes implementadas do pacote rm.data_types.quantity.date_time..	79
Quadro 24 - Classes implementadas do pacote rm.data_types.encapsulated.....	79
Quadro 25 - Classes implementadas do pacote rm.data_types.uri .....	80
Quadro 26 - Classe implementada do pacote rm.support.assumed.....	81
Quadro 27 - Classes implementadas do pacote rm.support.identification.....	81
Quadro 28 - Classes implementadas do pacote rm.support.terminology .....	83
Quadro 29 - Classe implementada do pacote rm. support.measurement .....	83

Quadro 30 - Classes implementadas do pacote rm.support.definition .....	84
Quadro 31 - Classe implementada do pacote rm.integration .....	85
Quadro 32 - Classe implementada do pacote rm.security .....	85
Quadro 33 - Operações disponibilizadas pelo serviço ArchMan .....	90
Quadro 34 - Operações disponibilizadas pelo EHRMan para o serviço de Pacientes .....	93
Quadro 35 - Operações disponibilizadas pelo EHRMan para o serviço de dados do RES.....	94
Quadro 36 - Operações disponibilizadas pelo serviço RulesMan .....	97
Quadro 37 - Informações extraídas a partir do trecho da diretriz .....	111
Quadro 38 - Regras utilizadas na exemplificação do funcionamento do Drools Expert .....	114

## LISTA DE ABREVIATURAS E SIGLAS

ACR	<i>American College of Radiology</i>
ADL	<i>Archetype Definition Language</i>
ADT	<i>Admission, Discharge and Transfers</i>
AM	<i>Archetype Model</i>
ANSI	<i>American National Standards Institute</i>
ASTM	<i>American Society of Testing and Materials</i>
BC	Base de Conhecimento
BD	Banco de Dados
B-RES	Base de Registro Eletrônico em Saúde
CDA	<i>Clinical Document Architecture</i>
CEN	<i>European Committee for Standardization, do francês Comité Européen de Normalisation</i>
CfH	<i>Connecting for Health</i>
CID	Classificação Estatística Internacional de Doenças e de Problemas Relacionados à Saúde
CKD	<i>Chronic Kidney Disease</i>
CKM	<i>Clinical Knowledge Manager</i>
CNS	Cadastro Nacional de Saúde
CR	<i>Carriage Return</i>
dADL	<i>Data ADL</i>
DCE	<i>Distributed Computing Environment</i>
DICOM	<i>Digital Imaging and Communication in Medicine</i>
DRC	Doença Renal Crônica
DRL	<i>Drools Rule Language</i>
EHR	<i>Electronic Health Record</i>
EN	<i>European Standard / European Norm</i>
ER	Entidade-Relacionamento
GDL	<i>Guideline Definition Language</i>
GEHR	<i>Good European Health Record</i>
GEM	<i>Guideline Elements Model</i>
GLIF	<i>GuideLine Interchange Format</i>

HL7	<i>Health Level 7</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IM	<i>Information Model</i>
ISO	<i>International Organization for Standardization</i>
JSF	<i>Java Server Faces</i>
JSON	<i>JavaScript Object Notation</i>
LF	<i>Line Feed</i>
LOINC	<i>Logical Observation Identifiers Names and Codes</i>
MG	Minas Gerais
NeHTA	<i>National e-Health Transition Authority</i>
NEMA	<i>National Electrical Manufacturers Association</i>
NHS	<i>National Health Service</i>
ORM	<i>Object/Relational Mapping</i>
RES	Registro Eletrônico em Saúde
REST	<i>Representational State Transfer</i>
RIM	<i>Reference Information Model</i>
RM	<i>Reference Model</i>
SAD	Sistema de Apoio à Decisão
SIS	Sistema de Informação em Saúde
SGBD	Sistema Gerenciador de Banco de Dados
SNOMED	<i>Systematized Nomenclature of Medicine</i>
SNOMED-CT	<i>SNOMED Clinical Terms</i>
SUS	Sistema Único de Saúde
TI	Tecnologia da Informação
TISS	Troca de Informações na Saúde Suplementar
TM	<i>Template Model</i>
TR	<i>Technical Report</i>
URI	<i>Universal Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
UUID	<i>Universally Unique Identifier</i>
XML	<i>eXtensible Markup Language</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>17</b>
1.1	OBJETIVO GERAL .....	19
1.2	OBJETIVOS ESPECÍFICOS.....	19
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>21</b>
2.1	REGISTRO ELETRÔNICO EM SAÚDE .....	21
2.2	INTEROPERABILIDADE .....	23
<b>2.2.1</b>	<b>Padrões de Informação em Saúde</b> .....	<b>23</b>
2.3	ARQUÉTIPOS .....	25
<b>2.3.1</b>	<b>Arquitetura de arquétipos</b> .....	<b>28</b>
<b>2.3.2</b>	<b>Adoção de arquétipos</b> .....	<b>33</b>
2.4	DIRETRIZES CLÍNICAS .....	34
<b>2.4.1</b>	<b>Padrões de representação de diretrizes clínicas</b> .....	<b>34</b>
2.5	SISTEMAS DE APOIO À DECISÃO .....	35
2.6	DOENÇAS RENAIAS CRÔNICAS.....	39
<b>3</b>	<b>MÉTODO</b> .....	<b>42</b>
3.1	FASE I – CONCEPÇÃO DO MODELO DE INTEGRAÇÃO .....	43
<b>3.1.1</b>	<b>Método proposto para integração</b> .....	<b>45</b>
<b>3.1.2</b>	<b>Modelo desenvolvido</b> .....	<b>47</b>
3.2	FASE II – IMPLEMENTAÇÃO DOS COMPONENTES DO SISTEMA.....	48
<b>3.2.1</b>	<b>Manipulação de arquétipos</b> .....	<b>49</b>
<b>3.2.2</b>	<b>Geração de interfaces para entrada de dados</b> .....	<b>50</b>
<b>3.2.3</b>	<b>Manipulação dos dados do RES</b> .....	<b>56</b>
<b>3.2.4</b>	<b>Manipulação de regras</b> .....	<b>58</b>
<b>3.2.5</b>	<b>Testes dos componentes desenvolvidos</b> .....	<b>59</b>
3.3	FASE III – INTEGRAÇÃO E APLICAÇÃO DO MOTOR DE INFERÊNCIA.....	61
<b>3.3.1</b>	<b>Construção da base de conhecimento</b> .....	<b>61</b>
<b>3.3.2</b>	<b>Inserção dos fatos na base de conhecimento</b> .....	<b>62</b>
<b>3.3.3</b>	<b>Execução das regras e atualização dos dados</b> .....	<b>63</b>
<b>3.3.4</b>	<b>Integração do motor de inferência ao RES</b> .....	<b>65</b>
<b>4</b>	<b>RESULTADOS</b> .....	<b>66</b>
4.1	IMPLEMENTAÇÃO PARCIAL DA ESPECIFICAÇÃO DO MODELO OPENEHR.....	67

<b>4.1.1 O IM de RES e o Pacote <i>EHR</i></b> .....	<b>68</b>
<b>4.1.2 Pacote <i>Composition</i></b> .....	<b>69</b>
4.1.2.1 Pacote <i>Content</i> .....	69
<b>4.1.3 O IM Demográfico e o Pacote <i>Demographic</i></b> .....	<b>70</b>
<b>4.1.4 O IM Comum</b> .....	<b>71</b>
4.1.4.1 Pacote <i>Archetyped</i> .....	71
4.1.4.2 Pacote <i>Generic</i> .....	72
4.1.4.3 Pacote <i>Directory</i> .....	73
4.1.4.4 Pacote <i>Change Control</i> .....	73
4.1.4.5 Pacote <i>Resource</i> .....	74
<b>4.1.5 O IM de Estruturas de Dados e o Pacote <i>Data Structures</i></b> .....	<b>74</b>
4.1.5.1 Pacote <i>Item Structure</i> .....	75
4.1.5.2 Pacote <i>Representation</i> .....	75
4.1.5.3 Pacote <i>History</i> .....	76
<b>4.1.6 O IM de Tipos de Dados</b> .....	<b>76</b>
4.1.6.1 Pacote <i>Basic</i> .....	76
4.1.6.2 Pacote <i>Text</i> .....	77
4.1.6.3 Pacote <i>Quantity</i> .....	78
4.1.6.4 Pacote <i>Encapsulated</i> .....	79
4.1.6.5 Pacote <i>Uri</i> .....	80
<b>4.1.7 O IM de Suporte</b> .....	<b>80</b>
4.1.7.1 Pacote <i>Assumed</i> .....	80
4.1.7.2 Pacote <i>Identification</i> .....	81
4.1.7.3 Pacote <i>Terminology</i> .....	82
4.1.7.4 Pacote <i>Measurement</i> .....	83
4.1.7.5 Pacote <i>Definition</i> .....	84
<b>4.1.8 O IM de Integração e o Pacote <i>Integration</i></b> .....	<b>84</b>
<b>4.1.9 O IM de Segurança e o Pacote <i>Security</i></b> .....	<b>85</b>
4.2 MODELO PARA O DESENVOLVIMENTO DE UM RES BASEADO EM ARQUÉTIPOS.....	85
<b>4.2.1 Repositório ArchRepo</b> .....	<b>87</b>
<b>4.2.2 Repositório EHRRepo</b> .....	<b>88</b>
<b>4.2.3 Repositório RulesRepo</b> .....	<b>88</b>
<b>4.2.4 Serviço ArchMan</b> .....	<b>89</b>

<b>4.2.5 Serviço EHRMan .....</b>	<b>92</b>
<b>4.2.6 Serviço RulesMan .....</b>	<b>96</b>
4.3 GERAÇÃO AUTOMÁTICA DE INTERFACE PARA A COLETA DE DADOS ESTRUTURADOS EM ARQUÉTIPOS.....	98
4.4 FORMA DE ARMAZENAMENTO DE DADOS ESTRUTURADOS EM ARQUÉTIPOS.....	104
4.5 EXECUÇÃO DE REGRAS DE DECISÃO BASEADAS EM DIRETRIZES....	107
4.6 INTEGRAÇÃO DE DIRETRIZES CLÍNICAS EM UM RES BASEADO EM ARQUÉTIPOS.....	109
<b>5 DISCUSSÃO .....</b>	<b>117</b>
<b>6 CONCLUSÃO.....</b>	<b>122</b>
6.1 TRABALHOS FUTUROS.....	122
<b>REFERÊNCIAS.....</b>	<b>124</b>
<b>APÊNDICE A – DIAGRAMAS DE CLASSES DO MODELO DE REFERÊNCIA ...</b>	<b>132</b>
<b>APÊNDICE B – CÓDIGOS HTTP RETORNADOS PELOS SERVIÇOS .....</b>	<b>133</b>
<b>APÊNDICE C – JSON DOS DADOS ESTRUTURADOS EM ARQUÉTIPO.....</b>	<b>134</b>

## 1 INTRODUÇÃO

Os Registros Eletrônicos em Saúde (RES) envolvem uma grande quantidade de dados clínicos que devem estar disponíveis durante toda a vida de um indivíduo. Por este motivo, deve ser garantida a disponibilidade das informações a partir de qualquer local, independente da instituição onde foram coletadas e do nível de atendimento clínico que as originou, sendo necessário para isto, que os sistemas que as armazenam, mesmo que em diferentes formatos, sejam interoperáveis (PÄUN et al., 2011).

Um dos principais requisitos para a interoperabilidade eficiente entre RES é a aplicação de padrões de conteúdo e informações em saúde possibilitando aos sistemas identificar as informações, seu formato e o contexto em que estão associadas.

Considerando este cenário, o padrão europeu (EN – *European Standard*) de arquitetura para RES definido pela Organização Internacional para Padronização (ISO – *International Organization for Standardization*) ISO/EN 13606 e a Arquitetura de Documentos Clínicos *Health Level 7* (HL7) release 2 (CDA – *Clinical Document Architecture*) têm seu uso bastante discutido e reportado atualmente.

A arquitetura proposta pela Fundação openEHR também se destaca neste cenário, mesmo ainda não sendo considerada como um padrão. Tanto o padrão ISO 13.606 quanto a arquitetura openEHR incorporam a abordagem de desenvolvimento de software chamada de modelo de dois níveis, caracterizados pela separação entre conhecimento e informação e introduz a metodologia dos arquétipos (RINNER et al., 2010).

Além de favorecer a interoperabilidade, a padronização dos dados facilita a incorporação de diretrizes clínicas nos RES, pois as informações contidas nestas, nada mais são que informações de natureza clínica, as mesmas tratadas pelos arquétipos, sendo essa integração, uma das tendências apresentadas atualmente. Marcos e Martínez-Salvador (2011) apresentam um exemplo de um formato de representação de arquétipos para a integração das diretrizes clínicas ao RES e citam a dificuldade de integração como um dos maiores obstáculos no aproveitamento dos potenciais benefícios destas, em uma forma computadorizada.

Contudo, deve-se considerar também que as diretrizes clínicas contêm, além das informações clínicas, relacionamentos entre os dados para a tomada de decisão

(CICOGNA, 2013). Dessa forma, a utilização de um mecanismo que permita o processamento destes relacionamentos favorece a utilização integrada das diretrizes clínicas no RES, conforme descrito por Marcos e Martínez-Salvador (2011) e Chen, Georgii-Hemming e Åhlfeldt (2009).

Além do acompanhamento clínico, as diretrizes podem auxiliar também em questões jurídicas, para justificar intervenções, por exemplo, em auditorias, monitoramento da qualidade do cuidado, liberações e indicações de exames e procedimentos com melhor custo benefício, entre outros, demonstrando sua importância também no âmbito da Saúde Suplementar (PORTELA, 2004).

No estudo de Cicogna (2013) foi utilizada uma diretriz clínica para o tratamento de pacientes com Doença Renal Crônica (DRC) em hemodiálise e elaborado um método para a representação desta diretriz clínica em arquétipos e regras de decisão para um motor de inferência, integrados a um RES.

Baseado no aumento na incidência e prevalência da DRC no Brasil percebe-se que este cenário pode ser alterado por meio de ações de diagnóstico precoce e tratamento adequados (MINISTÉRIO DA SAÚDE/DATASUS, 2008). Considerando-se a utilização das diretrizes clínicas por meio de Sistemas de Apoio a Decisão (SAD), mesmo que não completamente integradas ao RES, existe um aumento no desempenho da equipe profissional envolvida, conforme descrito por Barretto et al. (2003), permitindo diagnóstico mais ágil e de acordo com as recomendações estabelecidas nestas diretrizes.

Contudo, nos trabalhos apresentados, as diretrizes clínicas não estão completamente integradas ao RES, pois geralmente utilizam sistemas separados para a coleta dos dados e o suporte a decisão, como no trabalho de Anani et al. (2014) e não descrevem em detalhes as possíveis formas para esta integração.

Neste contexto, a proposta desta pesquisa é a integração de diretrizes clínicas de DRC em RES desenvolvido, baseado em arquétipos, utilizando a arquitetura openEHR, fornecendo suporte a decisão a partir das regras definidas nestas diretrizes clínicas. Algumas características importantes do RES, como segurança e de-identificação dos dados não foram abordadas neste documento pois não constituem o foco da pesquisa, porém devem ser implementadas no RES para que este possa ser utilizado na prática clínica.

O cenário ideal é que o RES seja desenvolvido já utilizando arquétipos como modelo de informação, porém sabe-se que o a substituição completa de um sistema

já existente possui alto custo e não é trivial. Desta forma, pode-se realizar uma substituição gradual do RES e até mesmo integrar dados provenientes de sistemas legados utilizando os arquétipos uma vez que existem estruturas genéricas definidas na especificação openEHR para a representação de informações que utilizam formatos próprios dos sistemas onde são coletadas. Considerando este cenário, o modelo proposto pode ser utilizado na evolução de sistemas existentes para realizar esta substituição. Sistemas que utilizam outros padrões, como o HL7 podem incorporar os arquétipos uma vez que esses dois modelos podem ser utilizados em conjunto, o primeiro para a comunicação entre sistemas e o segundo para a representação dos conceitos clínicos e conhecimento.

A contribuição científica desta pesquisa é o modelo de integração de diretrizes em um RES baseado em arquétipos de forma integrada, utilizando um mesmo formato para representação dos dados coletados e utilizados no apoio à decisão. Nos trabalhos anteriores do projeto no qual esta pesquisa está inserida, foram representados conjuntos de dados em arquétipos e realizado o apoio à decisão de forma separada, sem a utilização direta dos arquétipos pelo sistema. No trabalho de Cintho (2014) foram representados dados e regras de uma diretriz em arquétipos e proposto o modelo de RES com as diretrizes integradas porém o apoio à decisão também foi realizado de forma isolada ao RES. Já os trabalhos publicados não apresentam a integração entre o RES e as diretrizes. Estes descrevem o suporte à decisão, mas utilizam formalismos específicos para diretrizes, diferentes dos utilizados no RES onde os dados são coletados.

## 1.1 OBJETIVO GERAL

Integrar diretrizes clínicas ao Registro Eletrônico em Saúde baseado na arquitetura de arquétipos openEHR.

## 1.2 OBJETIVOS ESPECÍFICOS

- Especificar a forma de armazenamento de dados padronizados, baseado na arquitetura de modelagem de dois níveis;
- Desenvolver o mecanismo de leitura e processamento de arquétipos para a geração de interfaces para a captura e apresentação dos dados;
- Elaborar o mecanismo de processamento das regras extraídas das diretrizes clínicas para o suporte a decisão utilizando os dados padronizados;

- Construir um Sistema de Apoio à Decisão para prevenção da DRC integrado ao RES baseado na arquitetura de arquétipos openEHR.

## 2 REFERENCIAL TEÓRICO

Nesta seção será apresentada a definição de RES e seus aspectos principais de interoperabilidade. Serão abordados também alguns padrões em saúde, com ênfase na arquitetura de dois níveis, padronizada pela norma EN/ISO 13606 e proposta pela Fundação openEHR.

Em seguida, serão apresentadas as diretrizes clínicas com sua definição e os sistemas de apoio à decisão. Finalmente, será brevemente definida doença renal crônica e apresentada a diretriz para a prevenção da DRC, que será utilizada como prova de conceito para esta pesquisa.

### 2.1 REGISTRO ELETRÔNICO EM SAÚDE

Van Bemmelen e Musen (1997) descrevem que o prontuário do paciente tradicional em papel utilizado na prática clínica, geralmente é composto por anotações médicas e de outros provedores de cuidados, sendo complementado por dados de outras fontes como resultados de laboratórios e relatórios contendo o resultado de outros exames como raios-X, ultrassom e endoscopia.

Os relatórios médicos, ainda de acordo com van Bemmelen e Musen (1997), foram fortemente influenciados, no século V a.C., por Hipócrates que defendia que esses registros deveriam refletir com precisão, o curso da patologia e indicar as causas desta, sendo as observações ordenadas cronologicamente.

As anotações médicas passaram por diversas mudanças, agregando os achados médicos e de enfermagem com a invenção do estetoscópio em 1816 e de outros instrumentos para diagnóstico, sendo que em 1920, na Clínica Mayo, em Rochester – Minnesota, foi definido um conjunto mínimo de dados que todos os médicos da Clínica deveriam registrar, que veio a se tornar praticamente a estrutura atual para registros médicos.

Nos anos 60 foi introduzido o registro médico orientado a problemas, onde as anotações eram registradas por problemas, de acordo com a estrutura chamada SOAP, que significa subjetivo (ou queixa), objetivo (achados dos médicos e enfermeiras), avaliação (resultados de exames ou diagnósticos) e plano (planejamento médico, como um tratamento, por exemplo). Essa estrutura permite uma visualização mais clara do relacionamento entre as informações coletadas,

como a qual queixa os achados e tratamentos pertencem (BEMMEL, VAN e MUSEN, 1997).

Os registros médicos atuais normalmente são orientados a fontes. Isso significa que os diversos registros presentes no prontuário são ordenados de acordo com o método pelo qual os dados foram obtidos e, dentro de cada seção, são ordenados cronologicamente.

O uso mais comum dos registros médicos é o suporte ao cuidado de um indivíduo, contudo o desenvolvimento na área da saúde tem tornado mais complexa esta tarefa. Deve-se considerar também, que existe uma alta demanda no uso de dados dos pacientes para outros propósitos como o relato legal de ações médicas, suporte a pesquisas e gerenciamento de serviços de cuidados, por exemplo.

Além da demanda nos dados, pode-se destacar também diversos aspectos que são considerados como desvantagens do prontuário em papel, como a disponibilidade das informações que são bastante restritas nestes prontuários; a falta de estruturação dos registros e ilegibilidade da escrita; e a incapacidade de utilização dos dados coletados para o suporte a decisão clínica, como avisos e alertas. Esses fatores, combinados com o avanço no desenvolvimento da computação, despertou grande interesse no desenvolvimento de RES.

Um RES, de acordo com o relatório técnico (TR – *Technical Report*) ISO/TR 20.514, é um repositório de informações acerca do estado de saúde de um indivíduo, em uma forma processável por computador (ISO, 2005).

A definição de RES pode ser complementada pela apresentada na norma ISO 13.606, que considera o RES como um registro longitudinal e potencialmente multi-institucional e multinacional relacionados aos cuidados a um único sujeito da atenção (o paciente), criados e armazenados em um ou mais sistemas físicos (ISO, 2008).

Considerando que o RES pode ser utilizado como ferramenta para facilitar a continuidade do cuidado, o mesmo é considerado longitudinal, pois acompanha o paciente durante toda a sua vida, bem como interinstitucional, pois deve ser independente do local onde as informações são coletadas, considerando que seja realizada uma troca efetiva de informações com outros RES.

Contudo, do ponto de vista de padronização, a característica mais importante do RES é a habilidade deste em compartilhar informações entre diferentes usuários autorizados, sendo necessário para isso, em termos técnicos, interoperabilidade de

informação no RES e interoperabilidade de sistemas de RES que compartilham essa informação (ISO, 2005).

## 2.2 INTEROPERABILIDADE

A interoperabilidade pode ser considerada como a capacidade de troca de informações entre diferentes fontes sem que seja necessário conhecimento prévio de sua estrutura interna e arquitetura. Existem dois níveis principais de interoperabilidade de informação, conforme a norma ISO 20.514 (ISO, 2005): funcional e semântica.

A interoperabilidade funcional ou sintática se refere à habilidade de dois ou mais sistemas em trocar informação de forma que esta seja humanamente legível pelo receptor.

Na interoperabilidade semântica, a informação compartilhada deve ser compreensível no nível de conceitos de domínio formalmente definidos, permitindo que esta seja processável computacionalmente pelo sistema receptor. O nível de interoperabilidade semântica depende do nível de acordo nas terminologias e conteúdo das informações utilizadas pelos sistemas envolvidos.

Para alcançar a interoperabilidade semântica de informação, é necessário o atendimento de quatro pré-requisitos: padronização da arquitetura do RES, padronização dos modelos de interfaces de serviços, conjunto padronizado de modelos de conceitos específicos do domínio e terminologias padronizadas.

### 2.2.1 Padrões de Informação em Saúde

Para que haja uma troca efetiva de informações entre diferentes sistemas de informação em saúde (SIS), é necessário que sejam adotados padrões em saúde, sendo que, dentre os mais utilizados, pode-se destacar o padrão para imagens médicas (DICOM – *Digital Imaging and Communication in Medicine*), o padrão para troca de informações HL7, a HL7 CDA, o padrão para identificação de observações laboratoriais e clínicas (LOINC – *Logical Observation Identifiers Names and Codes*) e o padrão de arquétipos ISO 13.606.

O padrão de imagens médicas criado em 1985 pelo Colégio Americano de Radiologia (ACR – *American College of Radiology*) e pela Associação Nacional de Fabricantes de Materiais Elétricos dos Estados Unidos (NEMA – *National Electrical Manufactures Association*), com o nome de ACR/NEMA 300 e posteriormente

chamado de DICOM, especifica um protocolo de intercâmbio de informações não proprietário, um formato de imagem digital e uma estrutura de arquivo para imagens biomédicas e informações relacionadas a imagens. Este padrão define a forma e o fluxo das mensagens eletrônicas que transferem imagens e informações relacionadas entre computadores (BIDGOOD et al., 1997).

O padrão de troca de informação HL7, em sua versão 2 é o mais utilizado como padrão para interoperabilidade em saúde no mundo, sendo utilizado em mais de 90% dos hospitais nos Estados Unidos e altamente suportado por provedores de tecnologia da informação (TI) em saúde no mundo (BENSON, 2010). Este padrão foi elaborado pela organização HL7, certificada pelo Instituto Americano Nacional de Padrões (ANSI – *American National Standards Institute*), que desenvolve especificações amplamente utilizadas para permitir a troca de informações entre aplicativos médicos, integrando informações clínicas e administrativas, possuindo como representante no Brasil, o Instituto HL7 Brasil (HEALTH LEVEL SEVEN INTERNATIONAL, [S.d.]; INSTITUTO HL7 BRASIL, [S.d.]).

O HL7, originado em 1987, focava inicialmente na troca de informação sobre admissões, altas e transferências (ADT – *Admission, Discharges and Transfers*) nos hospitais. Em 1988, com a publicação da versão 2, também chamada de HL7 V2.0, o padrão passou a incluir mensagens para a troca de requisições e relatórios para resultados e tratamentos, baseado no padrão da Sociedade Americana para Testes e Materiais (ASTM – *American Society of Testing and Materials*) E.1238.88.

Enquanto o HL7 V2.0 utiliza mensagens compostas por segmentos, em formato de texto plano<sup>1</sup>, a versão 3 do HL7, iniciada em 1992, utiliza uma metodologia de desenvolvimento orientada a objetos e um Modelo de Referência de Informação (RIM - *Reference Information Model*) para criar mensagens, sendo o RIM a parte essencial da metodologia de desenvolvimento do HL7 V3, pois este modelo provê uma representação explícita das conexões semânticas e léxicas existentes entre as informações transmitidas nos campos das mensagens HL7 (BENSON, 2010). O maior uso do HL7 V3 consiste na aplicação da HL7 CDA, publicada pela primeira vez em 2000.

---

<sup>1</sup> Neste trabalho, será utilizado o termo texto plano para indicar texto simples, sem formatação, como tradução para o termo *plain text*.

A HL7 CDA é um padrão de marcação de documentos que especifica a estrutura de documentos clínicos intercambiados. Um documento CDA é um objeto de informação definido e completo que pode conter texto, imagens, sons e outros conteúdos multimídia e que podem ser transportados dentro de uma mensagem HL7 (DOLIN et al., 2000).

A HL7 CDA é definida em níveis, que representam um conjunto de especializações nos quais restrições adicionais são aplicadas. O nível um, que é a raiz da especificação, consiste conceitualmente em três componentes: cabeçalho, corpo do nível um e os tipos de dados do HL7 Versão 3. Os componentes codificados possuem domínios de vocabulários associados, representando o conjunto de valores permitidos para o componente. Estes domínios podem incluir conceitos definidos pelo HL7 ou sistemas de codificação reconhecidos pelo HL7 como Nomenclatura Sistematizada de Medicina (SNOMED – *Systematized Nomenclature of Medicine*) e LOINC (DOLIN et al., 2000).

O sistema de códigos LOINC, criado em 1994 por um grupo de pesquisadores no Instituto Regenstrief, possuía como objetivo inicial prover identificadores universais para observações em mensagens HL7.

Contudo o LOINC atualmente pode ser utilizado em mensagens de ultrassom DICOM e nas mensagens de indústria farmacêutica do Consórcio de Padrões para Intercâmbio de Dados Clínicos, para identificar observações clínicas e laboratoriais, respectivamente. Sua utilização pode se estender também a bancos de dados clínicos e de pesquisas, para os mesmos propósitos. O LOINC foi largamente adotado, por exemplo, nos Estados Unidos, onde a maioria das agências federais interessadas em cuidados de saúde o adotou. Países como Austrália, Coréia, Estônia, Brasil e Nova Zelândia também utilizam o LOINC (MCDONALD et al., 2003).

No âmbito de SIS, a arquitetura de dois níveis tem sido bastante relatada nos últimos anos. Essa metodologia, que incorpora a abordagem dos arquétipos, foi iniciada pela Fundação openEHR e padronizada pela norma ISO 13.606.

### 2.3 ARQUÉTIPOS

Os arquétipos podem ser definidos como expressões computáveis de um modelo de domínio na forma de declarações de restrições estruturadas baseadas

em um modelo de referência e correspondem ao modelo de conhecimento da arquitetura de dois níveis (BEALE e HEARD, 2007).

A arquitetura de dois níveis, inicialmente introduzida no projeto *Good European Health Record* (GEHR) nos anos 80, é baseada na definição de dois diferentes níveis conceituais, que utilizam seu próprio modelo de dados: nível de informação e nível de conhecimento (MENÁRGUEZ-TORTOSA, MARTÍNEZ-COSTA e FERNÁNDEZ-BREIS, 2012).

O nível de informação utiliza o modelo de referência (RM – *Reference Model*) que define o conjunto de entidades que forma os blocos construtores do RES, representa as características globais do sistema, reflete as características estáveis de um RES e inclui uma sintaxe flexível e alguns tipos genéricos de informações clínicas (LEZCANO, SICILIA e RODRÍGUEZ-SOLANO, 2011).

Este nível contém o mínimo necessário para suportar os requisitos médico-legais e funções de gerenciamento de registros, garantindo que as informações sejam enviadas de um RES a outro de forma compreensível, provendo a interoperabilidade de dados ou interoperabilidade sintática (GARDE et al., 2007).

O nível de conhecimento, representado pelos arquétipos, define os conceitos clínicos na forma de combinações estruturadas e restritas das entidades contidas no RM, definindo assim, o conhecimento clínico (MARTÍNEZ-COSTA, MENÁRGUEZ-TORTOSA e FERNÁNDEZ-BREIS, 2010) e provê a interoperabilidade semântica do RES (GARDE et al., 2007).

Dentre os níveis apresentados, somente o primeiro, ou seja, o RM, é implementado no software reduzindo significativamente a dependência dos sistemas implantados e dados, com as definições variáveis do conteúdo (BEALE e HEARD, 2008).

Nesta arquitetura, a parte interna do sistema é baseada no RM e modelo de arquétipos (AM – *Archetype Model*), incluindo a lógica genérica para armazenamento, busca entre outras ações, enquanto a semântica do domínio é geralmente delegada aos especialistas de domínio, responsáveis pela modelagem dos arquétipos – elementos reutilizáveis, *templates* – elementos de uso local, e pela definição das terminologias utilizadas. Dessa forma, considera-se que os arquétipos são livres de terminologia, ou seja, podem se utilizar de qualquer sistema de classificação adequado ao conceito clínico, sem restrição, como por exemplo, a

Classificação Estatística Internacional de Doenças e de Problemas Relacionados à Saúde (CID) ou a SNOMED para a definição de diagnósticos.

Com essa abordagem, desenvolvedores de software concentram suas atividades em componentes genéricos como gerenciamento de dados e interoperabilidade – o RM – enquanto grupos de especialistas atuam fora do processo de desenvolvimento, gerando as definições que serão utilizadas no sistema em tempo de execução – o AM (BEALE e HEARD, 2008).

A criação e modificação do conteúdo e subseqüentes buscas dos dados são controlados por arquétipos, sendo que estes são separados dos dados e armazenados em seus próprios repositórios, que normalmente incluem arquétipos provenientes de bibliotecas ou repositórios on-line bem conhecidos. Os arquétipos são implantados nos sistemas em tempo de execução por meio de *templates* que especificam grupos de arquétipos para um uso particular e frequentemente correspondem a um formulário de entrada de dados (BEALE e HEARD, 2008).

Atualmente a arquitetura dos arquétipos é padronizada pela norma do Comitê Europeu para Normalização da ISO (CEN – *European Committee for Standardization*) 13.606 que, em 2002, com o projeto de revisão da primeira versão, passou a adotar a abordagem de modelagem de dois níveis do openEHR, conhecida como metodologia de arquétipos e também o RM openEHR no padrão.

Contudo, o padrão ISO 13.606 é uma especificação para troca de extratos do RES e não considera outros requisitos como gerenciamento de versão das informações, gerenciamento de fluxos de trabalho e interfaces para outros sistemas, impedindo, por exemplo, que um extrato ISO 13.606 seja trocado com sistemas baseados na arquitetura openEHR ou qualquer outro sistema onde a preservação do versionamento da informação seja um requisito, pois estes extratos não possuem as informações necessárias apropriadas.

Por outro lado, a arquitetura openEHR provê, além da comunicação de extratos de RES de vários níveis de complexidade, especificação completa para criação, armazenamento, manutenção e consultas de RES, possuindo também um RM para RES, ausente tanto no padrão ISO/EN 13.606 quanto no RIM HL7 (SCHLOEFFEL, BEALE e HAYWORTH, 2006).

A Arquitetura de Registro Eletrônico em Saúde openEHR, ou simplesmente openEHR, é resultado de mais de 15 anos de pesquisas de diversos projetos e padrões e foi projetada com base em requisitos capturados durante vários anos. Por

ser altamente genérica, esta arquitetura satisfaz muitos requisitos além do conceito original de RES clínico, principalmente por ser orientada a arquétipos (BEALE e HEARD, 2008).

Esta arquitetura possui uma especificação aberta, detalhada e testada para uma plataforma computacional de informação em saúde interoperável, para RES e outros serviços como terminologias. Nos últimos anos, influenciou significativamente o desenvolvimento de padrões internacionais do CEN, HL7 e ISO, as três principais organizações de desenvolvimento de padrões em e-saúde (SCHLOEFFEL, BEALE e HAYWORTH, 2006).

Considerando que o padrão ISO/EN 13.606 é um subconjunto da arquitetura openEHR conforme descreve Schloeffel, Beale e Hayworth (2006), a seguir é apresentada a arquitetura de arquétipos, baseado na especificação 1.0.2 da arquitetura openEHR, de 2008 (BEALE e HEARD, 2008).

### 2.3.1 Arquitetura de arquétipos

Conforme já citado, a especificação dos arquétipos é baseada em uma arquitetura de dois níveis que separa o nível de informação, definido pelo RM do nível de conhecimento, definido pelo AM. Adicionalmente, a openEHR define um modelo de *templates* (TM – *Template Model*), que é uma extensão do modelo de arquétipos.

O nível de informação, que contém a especificação da estrutura básica dos registros e os elementos básicos da estrutura, é composto pelos seguintes componentes (BEALE e HEARD, 2008):

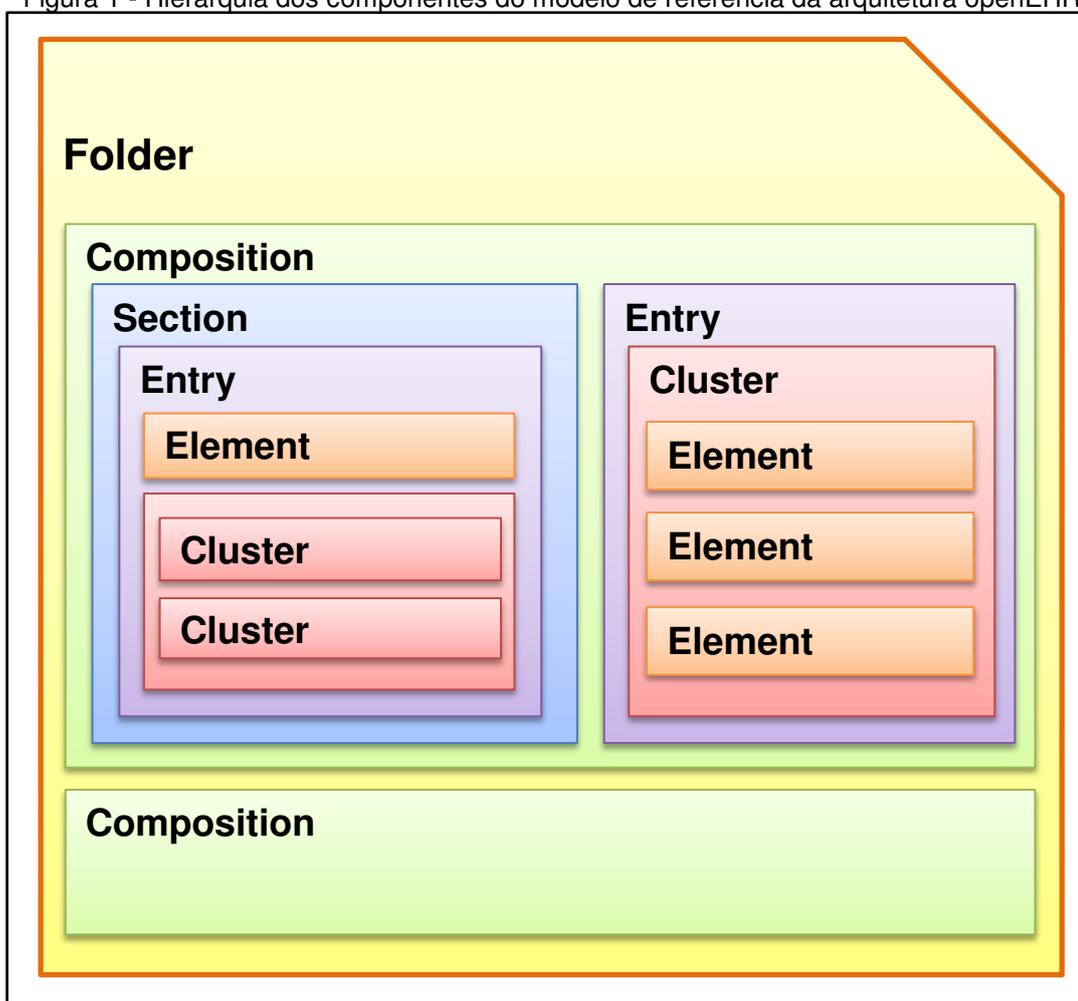
- *folder*: utilizado para agrupar e organizar diferentes *compositions* com características semelhantes, como por exemplo, episódios de atenção primária;
- *composition*: conjunto de informações clínicas, provenientes do registro de um episódio (uma consulta, um procedimento, registro de sumário de alta);
- *section*: facilita a navegação dentro de uma *composition*, identificando as diferentes seções do RES, como exames físicos, histórico familiar entre outros;
- *entry*: registro independente de uma observação, instrução, ação ou interpretação clínica, sendo considerado como uma sentença clínica simples, podendo conter um dado simples (como uma narrativa) ou dados complexos

(como *clusters* ou árvores). Exemplos de *entries* incluem a pressão arterial ou peso;

- *element*: um valor simples, como o nome de um medicamento ou sintoma;
- *cluster*: combinação de componentes organizada como uma estrutura aninhada dentro de uma mesma *entry*, podendo-se citar como exemplo, os detalhes de um exame de imagem.

A Figura 1 apresenta uma representação da hierarquia dos componentes do RM, de acordo com a especificação openEHR:

Figura 1 - Hierarquia dos componentes do modelo de referência da arquitetura openEHR



Fonte: o autor, 2014.

Adicionalmente, a arquitetura openEHR possui quatro especializações clínicas do componente *entry*, para cada categoria de registros apresentada anteriormente:

- *observation*: representa informações não interpretadas, derivadas diretamente de uma observação, como o registro da pressão sanguínea, um eletrocardiograma, peso, entre outros;
- *evaluation*: representa a interpretação ou avaliação de uma observação, podendo-se citar avaliação de risco, problema/diagnóstico e reações adversas;
- *instruction*: declarações que devem acontecer no futuro, como a solicitação de início de um fluxo de um processo ou a prescrição de medicamentos, por exemplo;
- *action*: indica uma ação que já foi realizada, como a administração de um medicamento prescrito. Comumente as ações são elaboradas de forma a complementar as instruções; uma *action* pode registrar estados subsequentes de uma *instruction*; quando uma *action* está associada a uma *instruction*, esta pode indicar a situação atual da *instruction* como aguardando, em execução, ou concluída, por exemplo. Contudo não é necessário que uma *instruction* preceda uma *action*, bem como não é necessário o registro de uma *action* após o registro de uma *instruction*.

O RM possui um conjunto de tipos de dados claramente definidos que provê os tipos necessários genérica e clinicamente, para basicamente todos os tipos de informação em saúde. O RM define as seguintes categorias para os tipos de dados:

- texto: texto plano, texto codificado e parágrafos;
- quantidades: qualquer tipo ordenado incluindo valores ordinais, utilizados para representar valores ordenados simbolicamente como “+”, “++” ou “+++”, quantidades mensuráveis com valores e unidades de medidas, entre outros;
- data e hora: data, hora, tipos de data e hora e tipos de data e hora parciais;
- dados encapsulados: conteúdo multimídia e conteúdos passíveis de análise, como a linguagem de marcação extensível (XML – *eXtensible Markup Language*); e
- tipos básicos: booleano, variáveis de estado.

Além destes tipos padrões, a arquitetura openEHR define ainda as seguintes estruturas de dados:

- *Item\_Table* para representar tabelas;

- *Item\_Tree* para representar árvores;
- *Item\_List* para representar listas;
- *Single\_Item* para representar um elemento simples; e
- *History\_of\_Events* para representar históricos de eventos.

Considerando as diferenças no RM adotado no padrão ISO 13.606 e do openEHR, a comunicação entre diferentes sistemas que utilizem ambos os padrões requer um mapeamento entre as estruturas do ISO 13.606 para o openEHR e vice e versa. Sendo a arquitetura openEHR mais especializada, a maior dificuldade está em transformar as estruturas do ISO 13.606 para a estrutura apropriada do openEHR. Por exemplo, como o ISO 13.606 possui apenas uma estrutura para *entry*, ao mapear um registro para openEHR deve ser identificado qual das *entries* disponíveis é adequada ao registro em questão (*observation, evaluation, instruction* ou *action*).

Além dos componentes já citados, existem outros que satisfazem os requisitos para troca de informação entre os sistemas, sendo:

- *audit information*: todas as *compositions* incluem dados de auditoria, para identificar o sistema de RES de origem, momento de criação entre outras informações;
- *functional role*: informação sobre quem realizou o cuidado, como quais e como tais cuidados foram realizados e quais os papéis desempenhados, por exemplo;
- *attestation information*: certifica a autenticidade da informação, armazenando um atestado de que a informação está correta e foi corretamente interpretada;
- *related party*: que permite o registro de assuntos relacionados ao paciente, de forma a identificá-lo univocamente;
- *link*: permite a ligação semântica entre partes do RES; e
- *demographic*: componente opcional que inclui descritores para qualquer pessoa, organização, dispositivo ou componente de software incluído no sistema de RES evitando repetição de detalhes descritivos e padronizando tais informações; em todo o sistema, um mesmo componente é descrito da mesma forma, como por exemplo, um equipamento, instituição ou um profissional.

Juntamente com o nível de informação, de forma a prover a interoperabilidade semântica das informações, a arquitetura especifica um modelo de arquétipos, que representa o nível de conhecimento e complementa o RM no domínio específico do conhecimento, sendo expresso pela linguagem de definição de arquétipos (ADL – *Archetype Definition Language*). Neste modelo, os blocos genéricos definidos no RM são combinados de forma a representar um conceito clínico completo e são definidos também, os relacionamentos semânticos destes conceitos.

Este modelo especifica a hierarquia dos componentes dos registros, restrições que devem ser aplicadas às informações, quais atributos são relevantes para o conceito, quais são os tipos de dados que representarão cada atributo e quais são os intervalos possíveis para os valores.

O AM, que inclui a definição do arquétipo e a informação necessária para criar um arquétipo, é composto por quatro pacotes:

- *archetype class*: detalha a informação do arquétipo, podendo ser um objeto complexo (*entry* ou *section*), um objeto primitivo (inteiro, data, texto entre outros) com restrições ou um espaço para arquétipo chamado *de archetype slot*;
- *archetype description*: onde são registrados os metadados associados ao arquétipo como autor, contribuidores, data de criação entre outros. Se o arquétipo possui tradução para outros idiomas, estas também são registradas;
- *constraint model*: expressa as restrições entre os nós dos arquétipos como os atributos, referência, cardinalidade e opcionalidade, por exemplo; e
- *archetype ontology*: provê a neutralidade de terminologia. Registra qual terminologia foi utilizada, nome dos termos e seus códigos, para que, se um extrato de RES for enviado para um sistema que utilize outra terminologia e esta possuir mapeamento para a terminologia em uso pelo sistema de origem, a informação possa ser recuperada mantendo o significado de origem.

Adicionalmente, o padrão openEHR define ainda, no nível de conhecimento, um modelo de *templates*, que pode ser definido como o agrupamento e customização de arquétipos adaptados para configurações clínicas locais, sendo

considerado como uma extensão do modelo de arquétipos e encontra-se em especificação, não sendo ainda um padrão.

Tecnicamente os *templates* atuam como arquétipos, porém, por definição, os arquétipos definem componentes largamente reutilizáveis enquanto os *templates* definem componentes (arquétipos) de uso local, de acordo com as necessidades de determinado contexto. Dessa forma, caso necessário, um *template* pode definir que um elemento específico de um arquétipo, desnecessário em um determinado local, não seja apresentado para coleta, por exemplo.

Este modelo pode permitir a reutilização de arquétipos e adoção dos mesmos em diferentes contextos e locais, pois permitem o refinamento dos seus elementos.

### **2.3.2 Adoção de arquétipos**

Atualmente existem diversas iniciativas de utilização dos arquétipos em diferentes países, como Reino Unido, Austrália e recentemente no Brasil.

No Reino Unido, o projeto Conexão para a Saúde (CfH – *Connecting for Health*) do Serviço Nacional de Saúde (NHS – *National Health Service*) levou à criação de um repositório contendo um grande número de arquétipos.

A Autoridade Nacional de Transição de e-Saúde (NeHTA – *National e-Health Transition Authority*) da Austrália define a utilização da metodologia openEHR como base para a modelagem dos conceitos clínicos, conforme NeHTA (2012).

No Brasil, a Portaria Nº 2.073 de 31 de agosto de 2011, publicada pelo Ministério da Saúde, define os arquétipos como forma de interoperabilidade de modelos de conhecimento no Brasil e o RM openEHR para a definição do RES (BRASIL, 2011).

Além disso, outros padrões internacionais também são citados nesta Portaria, como o HL7 para a interoperabilidade entre sistemas, a terminologia SNOMED termos clínicos (SNOMED-CT – *SNOMED Clinical Terms*) para a codificação de termos clínicos e mapeamentos das terminologias nacionais e internacionais em uso no país de forma a prover o suporte a interoperabilidade semântica entre os sistemas, o LOINC para a codificação de exames laboratoriais e o DICOM para a representação da informação relativa a exames de imagens (BRASIL, 2011).

## 2.4 DIRETRIZES CLÍNICAS

As diretrizes clínicas são documentos textuais, escritos em linguagem natural e que têm por objetivo nortear as tomadas de decisão e melhorar a qualidade do cuidado à saúde. Também possuem como objetivo diminuir o uso de intervenções desnecessárias, ineficientes ou perigosas (RŮZICKA e SVÁTEK, 2004; STEINDORFER, 2006; VESELÝ, ZVÁROVÁ e PELEŠKA, 2004).

Essas diretrizes são resultantes de processo de revisão sistemática das melhores evidências científicas disponíveis e contém dados clínicos e relacionamento entre os dados (STEINDORFER, 2006).

### 2.4.1 Padrões de representação de diretrizes clínicas

A utilização das diretrizes clínicas em sistemas computadorizados requer uma estruturação de seu conteúdo, de forma a extrair as informações constantes nestas (VESELÝ, ZVÁROVÁ e PELEŠKA, 2004). Existem disponíveis alguns métodos e padrões para a extração e representação de diretrizes clínicas, podendo-se citar os métodos baseados em marcação *Stepper* e *Guideline Elements Model* (GEM) e os padrões de representação de diretrizes *GuideLine Interchange Format* (GLIF) e *Arden Syntax*.

Com a finalidade de tentar integrar e possibilitar a troca de informações entre instituições de saúde foi desenvolvido em 1989 pelo mesmo grupo do HL7, o *Arden Syntax*. Atualmente encontra-se na versão 2.7 e é composto por módulos lógicos médicos focados em processos simples tomando apenas uma única decisão (CICOGNA, 2013).

O *Stepper* é uma ferramenta de marcação centrada em documento, que permite a decomposição do processo de formalização de diretrizes clínicas narrativas em diversos passos definidos pelo usuário e que correspondem a transformações XML interativas, sendo semelhante ao processo apresentado pelo GEM *cutter* (RŮZICKA e SVÁTEK, 2004). O resultado do processo pode ser exportado para ambientes operacionais de engenharia de conhecimento, como o Prolog ou para linguagens de programação convencionais, como estruturas de classes em Java (SVÁTEK e RŮŽIČKA, 2003).

O GEM é um modelo de documentos baseado em XML que foi adotado pela ASTM como padrão para representação de diretrizes utilizando XML. Este modelo utiliza uma hierarquia multinível para armazenar os diferentes tipos de informação

contidos em diretrizes clínicas. Estas informações incluem identificação de quem desenvolveu a diretriz, descrição do processo de desenvolvimento, definição do propósito da diretriz, o público-alvo, a população-alvo de pacientes e as próprias recomendações. A utilização deste modelo pode ser vista no trabalho de Gershovich e Shiffman (2001) onde são gerados formulários Web para entrada de dados a partir de arquivos codificados GEM, porém sem integração com o fluxo de trabalho clínico (SHIFFMAN et al., 2004).

O GLIF foi iniciado em 1996 como uma tentativa de padronizar a linguagem especificamente para a representação de diretrizes, por um grupo conduzido por cientistas cognitivos da Universidade de McGill. Inicialmente seu objetivo era ser um meio de tradução entre diversos formalismos de diretrizes clínicas (PELEG et al., 2004).

Em 1998, com o lançamento da segunda versão (GLIF2), este passou a permitir a modelagem de diretrizes clínicas como fluxos de passos estruturados, representando ações clínicas e decisões – chamados de passos condicionais. A versão mais atual – GLIF3 – permite a codificação das diretrizes clínicas em três níveis: um fluxo conceitual, uma especificação computável que pode ser verificada por consistência lógica e por sua integridade, e uma especificação implementável, que é direcionada à incorporação destas em sistemas de informação institucionais em particular (BOXWALA et al., 2004).

Contudo, muitos projetos de transformação de diretrizes clínicas em formato computacional são baseados em modelos conceituais, frequentemente sob a forma de fluxos, formulados pelos especialistas no domínio e gradualmente convertidos em um formato operacional, que é então incorporado ao sistema (RŮZICKA e SVÁTEK, 2004). Normalmente estes padrões ou formas de representação apresentam-se de forma isolada ou restritos a uma especialidade, não sendo completamente integrados ao RES.

## 2.5 SISTEMAS DE APOIO À DECISÃO

Os sistemas de apoio à decisão clínica ou SAD são sistemas que têm por finalidade auxiliar os profissionais da saúde a tomar decisões clínicas, sendo considerado como SAD qualquer sistema computacional que manipula dados ou conhecimentos clínicos e objetiva prover o suporte a decisão (SHORTLIFFE e CIMINO, 2006).

Os SAD podem ser considerados como sistemas interativos, baseados em computador e que auxiliam pessoas na utilização de comunicação, dados, documentos, conhecimentos e modelos computacionais para a resolução de problemas e tomada de decisão. Estes são sistemas auxiliares – não devem ser pensados como substitutos de tomadores de decisões ou especialistas habilitados (POWER, 2002).

No contexto da saúde, por meio da análise dos dados já inseridos em um SIS, os SAD devem gerar orientações aos profissionais médicos, durante o atendimento de forma a orientar o encaminhamento da consulta e solicitações de novos dados, como exames laboratoriais e de imagem, por exemplo, podendo inclusive, indicar tratamentos a ser prescritos para o paciente (BARRETTO et al., 2003).

Conforme Power (2002), os SAD possuem três características principais: são elaborados especificamente para facilitar processos de decisão; devem fornecer suporte ao invés de automatizar a tomada de decisão; e devem estar aptos a responder rapidamente a mudanças nas necessidades dos tomadores de decisão.

Nos anos 70, três sistemas de recomendações permitiram ter uma visão geral da origem dos trabalhos em SAD: sistema para diagnóstico de dor abdominal – *Leeds Abdominal Pain System* – (DOMBAL, DE et al., 1972), o sistema para a entrega de alertas médicos de admissão – HELP (KUPERMAN, GARDNER e PRYOR, 1991) e o MYCIN, um sistema para a seleção de terapia antibiótica (SHORTLIFFE, 1976).

O *Leeds Abdominal Pain System* foi desenvolvido na Universidade de Leeds, a partir dos estudos do processo de diagnóstico nos anos 60, por deDombal e seus associados, utilizando a Teoria de Bayes<sup>2</sup>. Utilizando a sensibilidade, especificidade e dados de prevalência de patologias para diversos sinais, sintomas e resultados de exames, o sistema calculava, por meio da Teoria de Bayes, a probabilidade de sete possíveis explicações para a dor abdominal aguda (SHORTLIFFE e CIMINO, 2006).

O HELP, conforme relatam Shortliffe e Cimino (2006), é um sistema de informação hospitalar integrado que foi desenvolvido no Hospital LDS na cidade de Salt Lake e possui a habilidade de gerar alertas ao encontrar anormalidades nos registros dos pacientes. Este sistema adiciona ao sistema de registro médico

---

<sup>2</sup> A Teoria de Bayes é um método quantitativo para o cálculo da probabilidade pós-teste utilizando a probabilidade pré-teste e a sensibilidade e especificidade do teste. Este teorema deriva da definição de probabilidade condicional e das propriedades de probabilidade (SHORTLIFFE e CIMINO, 2006).

convencional, um programa monitor e um mecanismo para o armazenamento de lógicas de decisão.

O sistema MYCIN focava na indicação de um gerenciamento apropriado de pacientes com doenças infecciosas e utilizava regras de produção<sup>3</sup> para representar o conhecimento destas doenças. Apesar de não ter sido utilizado clinicamente, o MYCIN influenciou fortemente o desenvolvimento de sistemas baseados em conhecimento e a comercialização da abordagem baseada em regras, em diversas áreas, nos anos 80 (SHORTLIFFE, 2006).

Como é possível observar, diversas técnicas podem ser utilizadas no desenvolvimento de SAD, podendo-se citar a teoria de probabilidade ou Teoria de Bayes e a representação de conhecimento em sistemas baseados em conhecimento. Estes sistemas podem ser definidos como programas que codificam, simbolicamente, conceitos derivados de especialistas em uma área específica em uma base de conhecimento (BC) e que utiliza esta base para prover a análise de recomendação para algum tipo de problema que o especialista poderia prover (SHORTLIFFE e CIMINO, 2006).

Uma das formas de codificar o conhecimento dos especialistas é com a utilização de regras de produção, como utilizado no MYCIN, por exemplo, permitindo que um processador de regras – um motor de inferência – possa processar tais regras, pesquisando por situações que disparem as regras relevantes, avaliando as condições destas e executando as ações que as regras pertinentes especificam.

O motor de inferência é o componente de um sistema baseado em conhecimento, responsável pelo desenvolvimento do raciocínio baseado nas informações coletadas e no conhecimento representado em uma BC. Nos sistemas baseados em regras, podem ser utilizadas duas abordagens para o processamento destas: encadeamento regressivo e encadeamento progressivo (REZENDE, 2005).

No encadeamento regressivo ou *backward chaining*, o raciocínio parte da suposição de que cada provável solução é verdadeira, buscando por evidências que comprovem que a solução previamente considerada seja correta. O encadeamento progressivo ou *forward chaining* utiliza as informações fornecidas pelo usuário para

---

<sup>3</sup> Regras de produção, segundo Shortliffe e Cimino (2006) são declarações condicionais que relatam observações associadas a inferências que podem ser elaboradas. Normalmente são apresentadas na forma “SE antecedente ENTÃO consequente” onde o consequente é executado apenas se os antecedentes forem verdadeiros.

obter uma solução para o problema, consistindo na análise destes dados e baseando-se no conhecimento para chegar a uma conclusão (REZENDE, 2005).

A utilização de um motor de inferência no RES permite a separação dos dados e regras, evitando que a lógica seja incluída diretamente no sistema, facilitando a manutenção e alteração das regras.

Um dos motores de inferência disponíveis atualmente, é o JBoss Drools Expert, referenciado a partir deste ponto como Drools Expert, um motor de código aberto, orientado a objetos e desenvolvido em Java que utiliza o encadeamento progressivo, porém permite também a utilização do encadeamento regressivo (THE JBOSS DROOLS TEAM, [S.d.]). Este motor utiliza uma linguagem própria para a definição das regras, a Linguagem de Regras do Drools (DRL – *Drools Rule Language*) e também permite a criação das regras em forma de tabelas de decisão, utilizando planilhas eletrônicas, conforme exemplificado na Figura 2:

Figura 2 - Tabela de decisão em planilha eletrônica

	RuleTable Deficiência de Ferro				
	Condição	Condição	Condição	Condição	Ação
	exame : ExamesVO				exame
	Saturação Transferrina < \$1	Saturação Transferrina >= \$1	Ferritina >= \$1, Ferritina < \$2	Ferritina >= \$1	setValue("deficiencia Ferro", "\$param")
<b>Deficiencia Ferro Rules</b>	<b>Índice de saturação transferrina</b>	<b>Índice de saturação transferrina</b>	<b>Intervalos de IST</b>		<b>Deficiência de ferro</b>
Deficiencia Absoluta	20		0,200		Absoluta
Deficiencia Relativa			200,800		Relativa
Deficiencia Relativa				800	
Sem deficiência		20			Sem deficiência

Fonte: Cicogna, 2013.

O Drools Expert é baseado no algoritmo Rete<sup>4</sup> e possui uma implementação estendida deste, chamada de ReteOO, uma implementação melhorada e otimizada deste algoritmo para sistemas orientados a objetos (THE JBOSS DROOLS TEAM, [S.d.]).

<sup>4</sup> O algoritmo rete é um método eficiente para comparação de grandes coleções de padrões em uma grande quantidade de objetos, buscando todos os objetos que se enquadram em cada padrão (FORGY, 1982).

Neste motor de inferência, as regras são armazenadas na memória de produção e os fatos que este avalia são mantidos na memória de trabalho, onde podem ser modificados ou removidos. Um sistema com uma grande quantidade de regras e fatos pode resultar na ativação de diversas regras para uma mesma avaliação de um fato, sendo que neste caso, diz-se que estas regras estão em conflito. Neste ponto, um componente do Drools Expert, chamado de Agenda, gerencia a ordem de execução destas regras em conflito utilizando uma estratégia de resolução de conflito.

De uma forma simplificada, pode-se resumir a execução do Drools Expert sobre os dados ou fatos inseridos, como descrito a seguir, conforme Bali (2009):

- O motor de inferência inicialmente ativa todas as regras em que as condições são satisfeitas, ou seja, seleciona todas as regras aplicáveis;
- Uma das regras ativadas é selecionada, com base em uma estratégia de resolução de conflitos e então é executada;
- Após a execução, o motor de inferência reavalia as mudanças que podem ter ocorrido pela execução da regra anterior, que pode levar outras regras a serem ativadas ou desativadas;
- Este processo é repetido até que não haja mais regras ativadas (ou selecionadas).

Baseado na execução resumida do motor de inferência apresentada, mais de uma regra pode ser disparada para cada registro submetido a este, o que permite a utilização deste em SAD que possuam diretrizes clínicas integradas que possuem mais de uma ação possível para um mesmo caso (ou fato), conforme é apresentado no trabalho de Cicogna (2013), onde é utilizada uma diretriz para o tratamento de pacientes portadores de doença renal crônica.

## 2.6 DOENÇAS RENAIIS CRÔNICAS

Definida como a presença de dano renal ou diminuição da função renal por três meses ou mais, as DRC constituem atualmente um importante problema médico e de saúde pública (NATIONAL KIDNEY FOUNDATION, 2002; ROMÃO JUNIOR, 2004).

De acordo com National Kidney Foundation (2002) e Barros et al. (2006), a DRC é classificada em cinco estágios que variam de acordo com a presença do

dano renal e grau de diminuição das funções renais, sendo o quinto e último estágio, a insuficiência renal terminal.

Neste último estágio é realizada a terapia renal substitutiva que pode ser hemodiálise ou diálise peritoneal. Para garantir uma adequação do tratamento realizado, todo o acompanhamento necessário do paciente segue diretrizes clínicas elaboradas por sociedades médicas da área de nefrologia (CICOGNA, 2013).

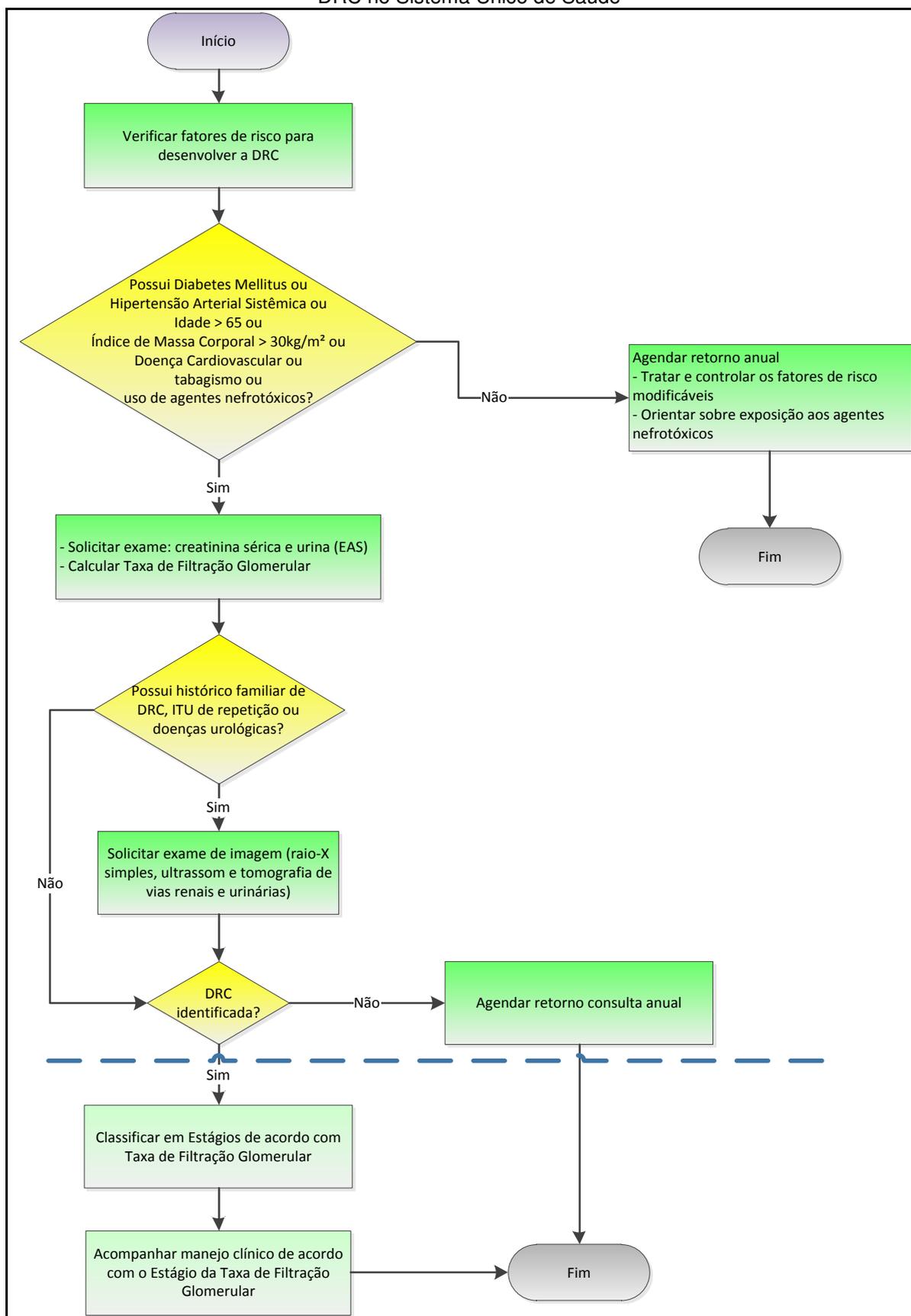
Considerando os altos custos envolvidos no tratamento desta doença no Brasil pelo Sistema Único de Saúde (SUS) que, no período de 2013, foi a fonte pagadora das diálises realizadas por cerca de 84% dos pacientes, totalizando aproximadamente 43 mil pacientes tratados e o registro de quase 18% de mortalidade destes pacientes, evidências científicas demonstram que ações preventivas podem modificar esta realidade, como diagnóstico precoce e tratamentos adequados à prevenção da evolução da DRC (BASTOS, BASTOS e TEIXEIRA, 2007; SOCIEDADE BRASILEIRA DE NEFROLOGIA, 2013).

Entretanto, até o ano de 2013 não existia uma diretriz no Brasil a respeito da prevenção da DRC, apenas um manual técnico publicado pelo Ministério da Saúde e cadernos de atenção básica que objetivavam estimular a mudança do modelo de atenção à saúde por meio de programas de promoção da saúde e prevenção de riscos e doenças (BRASIL, 2006).

No ano de 2014 foram publicadas as diretrizes clínicas para o cuidado ao paciente com DRC no SUS, após integrar a DRC no Plano de Enfrentamento das Doenças Crônicas Não Transmissíveis. A Figura 3 apresenta um fluxograma resumido para o atendimento à prevenção, diagnóstico e acompanhamento e manejo desta patologia de acordo com as Diretrizes Clínicas para o Cuidado ao Paciente com Doença Renal Crônica - DRC no Sistema Único de Saúde (BRASIL, 2014).

Desta forma, os profissionais que atuam diretamente com estes pacientes podem basear suas ações nesta diretriz publicada, permitindo maior qualidade no atendimento, principalmente considerando a integração desta diretriz ao RES, pois poderá auxiliar os profissionais de saúde, favorecendo a agilização e a padronização das condutas, em conformidade com os protocolos estabelecidos, na tomada de decisão (CINTHO, 2014).

Figura 3 - Fluxograma das Diretrizes Clínicas para o Cuidado ao Paciente com Doença Renal Crônica - DRC no Sistema Único de Saúde



Fonte: adaptado de Cintho, 2014.

### 3 MÉTODO

Trata-se de uma pesquisa exploratória, uma vez que objetivou a identificação de possíveis formas de integração de diretrizes clínicas ao RES, possibilitando o suporte a decisão; aplicada, considerando que o resultado desta pesquisa é direcionado à resolução do problema de integração das diretrizes clínicas ao RES; e com desenvolvimento, pois foi elaborado um RES contendo o método proposto na resolução do problema apresentado.

A pesquisa foi dividida em 3 fases, como apresentado no Quadro 1:

Quadro 1 – Fases da pesquisa

<b>Fase I</b>	Concepção do modelo de integração
<b>Fase II</b>	Implementação dos componentes do sistema
<b>Fase III</b>	Integração e aplicação do motor de inferência

Fonte: o autor, 2014.

A escolha de algumas tecnologias e forma de utilização destas, que não impactam diretamente no resultado desta pesquisa, foi realizada de acordo com a simplicidade de uso para permitir a operacionalização do modelo proposto, sendo que não foram realizados estudos acerca destas para aplicação na pesquisa. O sistema gerenciador de banco de dados (SGBD), por exemplo, foi selecionado por prover funcionalidades específicas em algumas etapas da pesquisa, porém em outras, foi utilizado por conveniência, de forma a reduzir a complexidade na utilização de mais de um SGDB desnecessariamente.

A utilização de uma base de conhecimento *stateless*, que não mantém o estado entre uma execução e outra, para a execução das regras no motor de inferência, por não apresentar impactos negativos nos experimentos realizados, foi selecionada na etapa correspondente para permitir a execução do apoio à decisão sem utilizar muitos recursos computacionais, principalmente a memória principal já que os objetos utilizados durante a execução são descartados em seguida. Isto permite que o desenvolvimento e os testes sejam executados mesmo em equipamentos que não disponham de muitos recursos de hardware.

Estas escolhas foram realizadas durante o desenvolvimento do projeto, desde o seu início, de acordo com a necessidade de cada etapa, baseadas em

experiências anteriores, familiaridade do pesquisador com os recursos envolvidos e relatos publicados na literatura.

### 3.1 FASE I – CONCEPÇÃO DO MODELO DE INTEGRAÇÃO

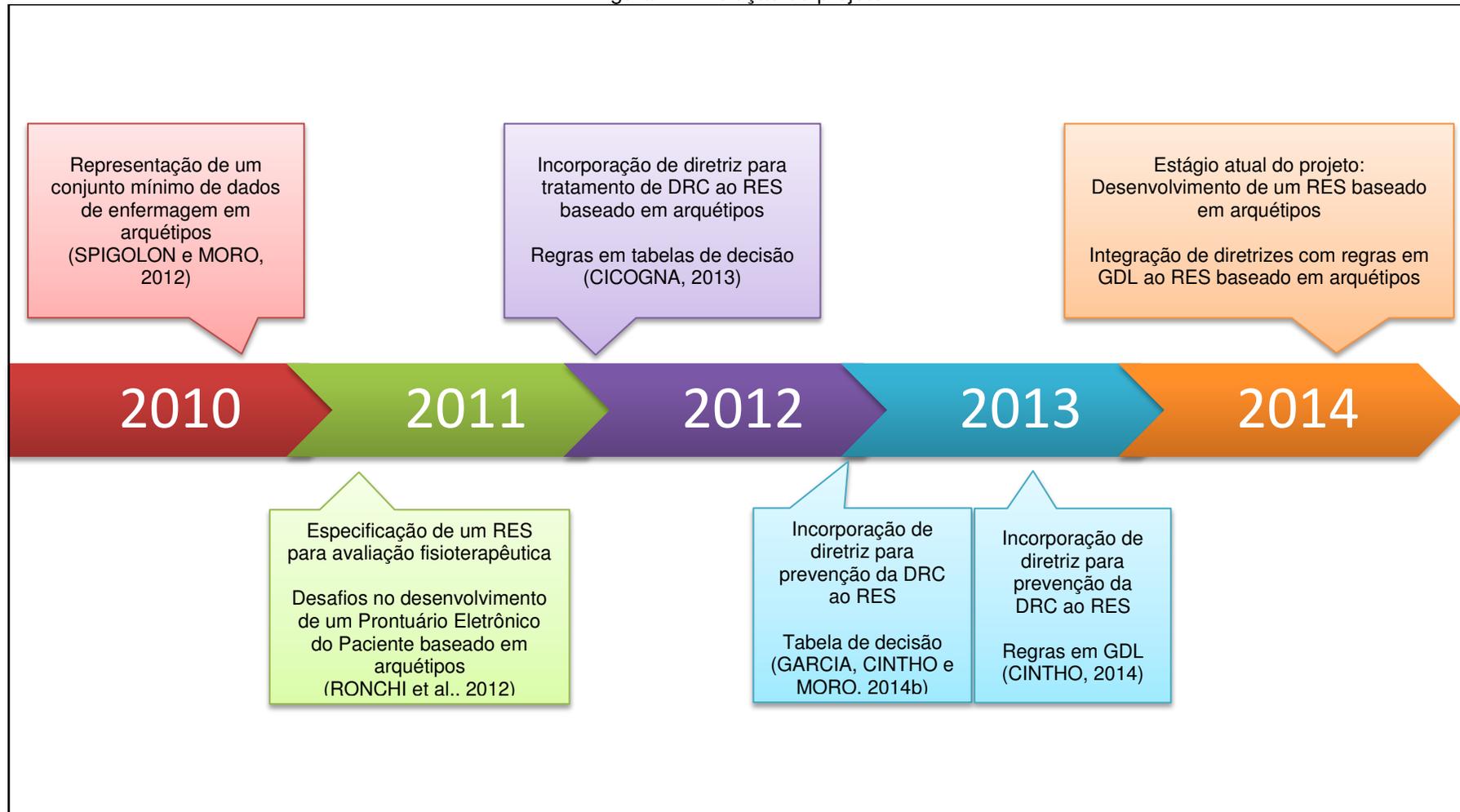
Considerando-se que modelo de integração é a principal contribuição desta pesquisa, é importante conhecer a evolução do projeto no qual ela está inserida. Na Figura 4 é possível acompanhar esta evolução, onde são apresentados os principais pontos de cada estudo que contribuíram para esta pesquisa.

O projeto iniciou-se no final de 2010 com a representação do conjunto mínimo de dados de enfermagem proposto para o atendimento de portadores de endometriose por meio de arquétipos (SPIGOLON e MORO, 2012).

A partir desta primeira experiência que permitiu a representação dos dados utilizando arquétipos, foi iniciado o trabalho para especificação do RES para avaliação fisioterapêutica funcional de pacientes com lesão medular espinal. Com este estudo foram iniciadas análises dos desafios no desenvolvimento de prontuários eletrônicos baseados neste modelo (RONCHI et al., 2012). Porém, somente a representação de dados utilizando-se arquétipos não resolve as dificuldades de integração, incluindo também a possibilidade de suporte a decisão utilizando diretrizes clínicas.

Visando resolver estas questões, no final de 2011 foi iniciado um estudo para a incorporação da diretriz para o tratamento de portadores de DRC ao RES baseado em arquétipos (GARCIA et al., 2013). Neste estudo os dados foram representados por arquétipos, mas estes não foram utilizados no protótipo do SAD e na elaboração das regras. A especificação das regras identificadas na diretriz foi feita utilizando tabelas de decisão elaboradas em uma planilha eletrônica. Para a execução das regras foi aplicado o motor de inferência, Drools Expert. Isto é, o protótipo desenvolvido não processava os arquétipos. Assim, a questão da integração ainda não estava resolvida e também não havia publicações que relatassem proposições para o atendimento desta necessidade.

Figura 4 - Evolução do projeto



Fonte: o autor, 2014.

No final de 2012 dando continuidade ao projeto, foi proposto um estudo com foco na prevenção da DRC, o qual iniciou com a mesma abordagem de definição de regras em tabelas de decisão (GARCIA, CINTHO e MORO, 2014b). Porém, no final do 1º semestre de 2013, identificou-se que estava disponível no site da openEHR uma ferramenta para a elaboração de regras de decisão utilizando arquétipos, denominada GDL Tools<sup>5</sup>. Esta utiliza uma linguagem própria baseada na ADL, para representação de regras de decisão contidas nas diretrizes clínicas: Linguagem de Definição de Diretrizes (*Guideline Definition Language – GDL*).

Nesta época não existiam relatos da utilização da GDL. Rong Chen, um dos autores da ferramenta GDL Tools, em agosto de 2013, durante a conferência MEDINFO 2013 (XIV Congresso Mundial de Informática Médica) apresentou um trabalho (CHEN et al., 2013) referente à representação de diretriz clínica, citou a aplicação do GDL, mas não descreveu o emprego desta linguagem.

Dando continuidade ao projeto, a partir dos arquétipos elaborados para o SAD de prevenção da DRC foram iniciados experimentos para elaborar as regras de decisão com a GDL (CINTHO, 2014). Os arquétipos e as regras, assim como o protótipo foram validados por especialistas com sucesso. Contudo ainda existia uma carência na integração, pela falta de um RES estruturado em arquétipos que fornecesse os dados para a execução das regras de decisão.

### **3.1.1 Método proposto para integração**

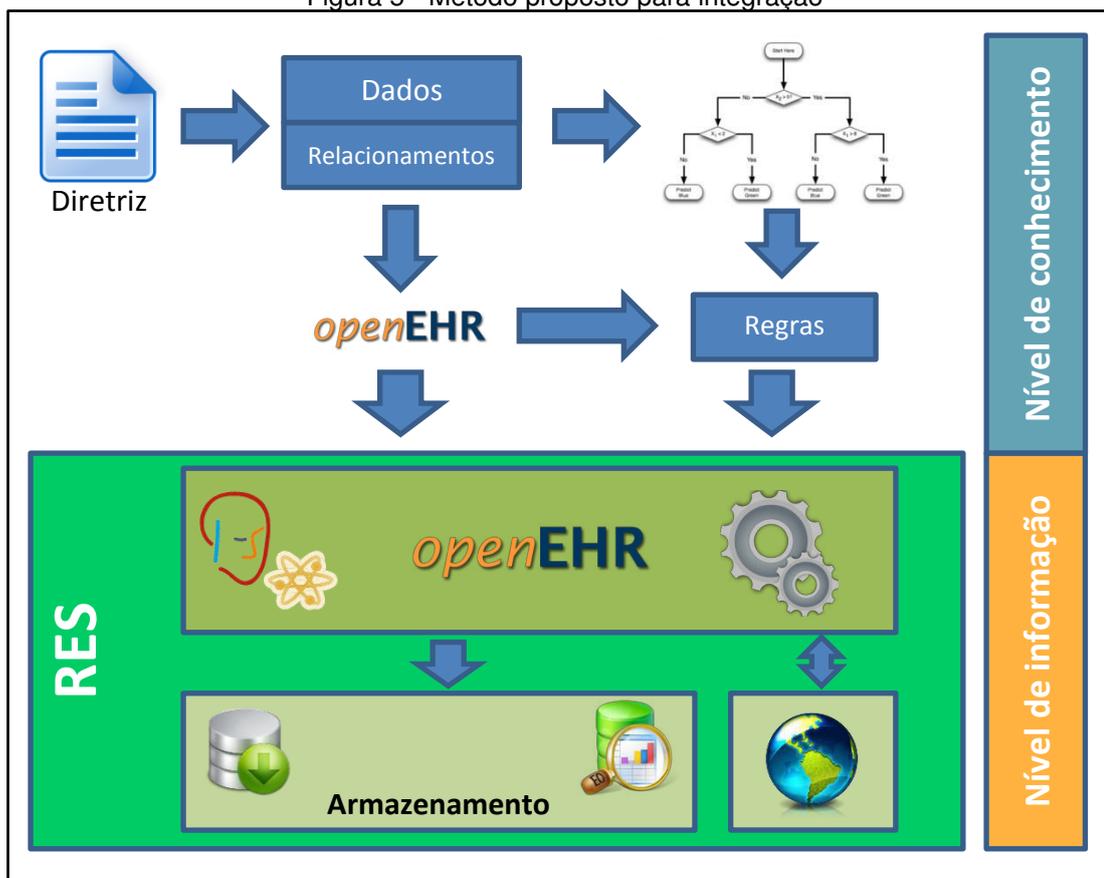
O método proposto, apresentado na Figura 5, foi iniciado em conjunto com o estudo de Cicogna (2013), onde foi primeiramente aplicado. A seguir, foi utilizado por Cintho (2014) que adequou a sumarização das informações da diretriz.

É importante ressaltar que, conforme descrito anteriormente, Cicogna elaborou as regras sem considerar a estrutura dos arquétipos, utilizando tabelas de decisão em planilhas eletrônicas e Cintho, por meio da GDL, especificou as regras extraídas baseando-se na estrutura e definição dos arquétipos utilizados. Porém, mesmo o estudo baseado em tabelas de decisão já utilizava o Drools Expert como motor de inferência, que também é aplicado na GDL Tools.

---

<sup>5</sup> [http://www.openehr.org/downloads/ds\\_and\\_guidelines](http://www.openehr.org/downloads/ds_and_guidelines)

Figura 5 - Método proposto para integração



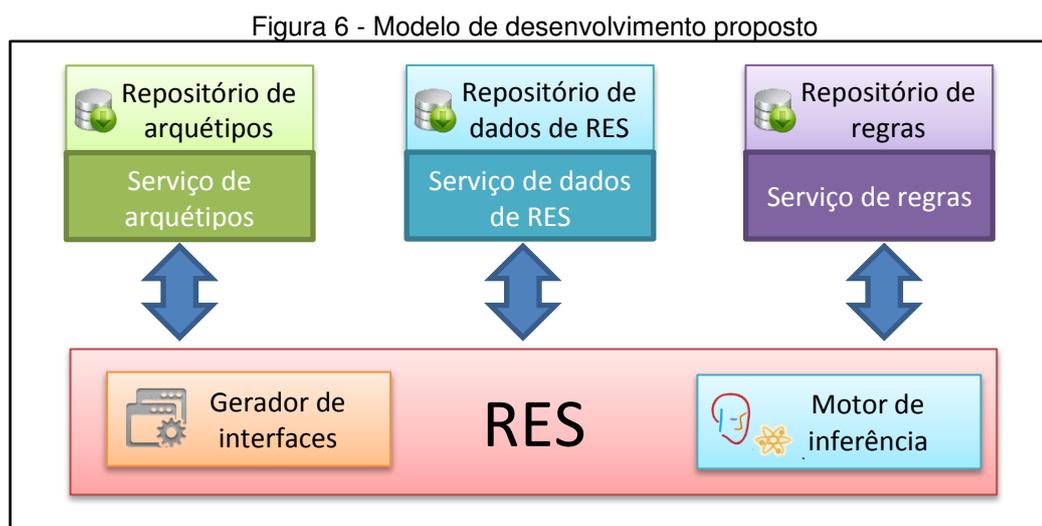
Fonte: o autor, 2014.

Neste método, constituído por 5 fases, inicialmente são identificados na diretriz, os dados e relacionamento entre eles, relacionamento estes que irão compor as regras de decisão. Em seguida, são elaborados os arquétipos necessários para a representação dos dados identificados na fase anterior, que não encontram-se disponíveis já elaborados. Na fase seguinte, as regras de decisão são representadas em um formato computacional, para serem processadas em um motor de inferência e o SAD é desenvolvido com base nestas regras.

O SAD desenvolvido na fase anterior é integrado ao RES para que o suporte a decisão seja realizado sobre os dados do RES, com base nas regras elaboradas. Para isso, o RES disponibiliza os dados ao motor de inferência que contém as regras e executa a inferência. Por fim, os resultados gerados pela execução das regras são validados de forma a assegurar que as regras estão corretamente definidas.

### 3.1.2 Modelo desenvolvido

A partir dos experimentos e resultados obtidos nos estudos anteriores, foi definido um modelo para o desenvolvimento do RES, apresentado na Figura 6, com a separação dos componentes de acordo com os três tipos de recursos que manipulam: arquétipos, regras e dados de RES.



Fonte: o autor, 2014.

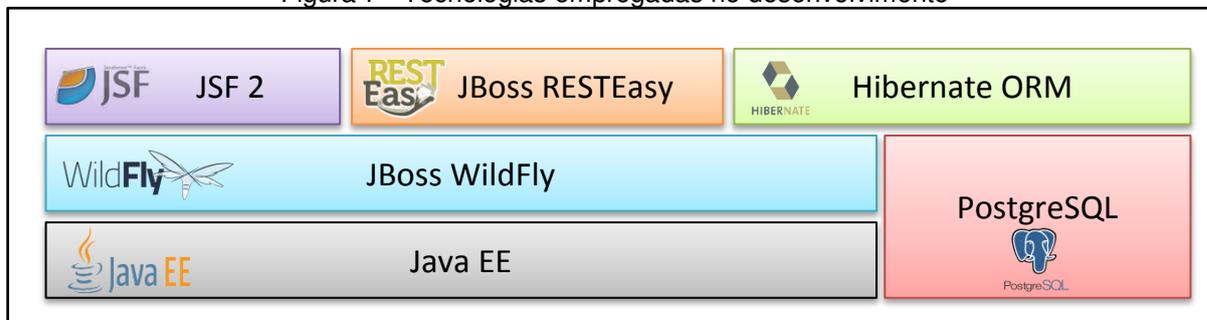
Utilizando uma abordagem de repositórios, foram desenvolvidos os componentes necessários: três repositórios e três serviços de acesso aos repositórios, bem como um sistema capaz de comunicar-se com estes, manipulando os recursos disponibilizados para prover suporte a decisão com base nos dados estruturados em arquétipos e regras baseadas nas diretrizes clínicas.

Para a coleta dos dados, baseados na estrutura dos arquétipos, de forma dinâmica, foi necessário também o desenvolvimento de um componente para a geração automática de interfaces ou telas a partir da definição dos arquétipos, para a entrada de dados. Este componente foi desenvolvido de forma integrada ao sistema, considerando que está fortemente relacionado ao tipo de sistema onde será utilizado, no caso desta pesquisa, um sistema web.

O desenvolvimento dos componentes foi realizado utilizando a tecnologia Java para a plataforma JavaEE (ORACLE, [S.d.]) e executados no servidor de

aplicação WildFly 8<sup>6</sup>, conforme demonstrado na Figura 7 que apresenta também outras tecnologias empregadas no desenvolvimento desta pesquisa.

Figura 7 - Tecnologias empregadas no desenvolvimento



Fonte: o autor, 2014.

A comunicação entre alguns componentes foi realizada por meio de serviços web RESTful desenvolvidos utilizando o estilo arquitetural denominado Transferência de Estado Representativo (REST – *Representational State Transfer*) (FIELDING, 2000) com a implementação JBoss RESTEasy<sup>7</sup>. Para a persistência dos dados foi utilizado o framework para mapeamento Objeto/Relacional (ORM – *Object-Relational Mapping*) Hibernate ORM<sup>8</sup> versão 4.3.1.Final e como SGBD PostgreSQL<sup>9</sup> versão 9.2, que apresenta características necessárias para alguns dos componentes, descritas nas seções a seguir.

### 3.2 FASE II – IMPLEMENTAÇÃO DOS COMPONENTES DO SISTEMA

Nesta fase foram implementados os sete componentes necessários para o funcionamento do sistema: repositório e serviço para a manipulação de arquétipos; mecanismo para geração de interfaces e entrada de dados; repositório e serviço para o armazenamento e recuperação dos dados do RES; e o repositório e serviço para manipulação das diretrizes clínicas.

Para facilitar a leitura, o Quadro 2 apresenta o nome atribuído a cada componente desenvolvido, que será utilizado durante o texto:

<sup>6</sup> <http://wildfly.org/8/>

<sup>7</sup> <http://resteasy.jboss.org/>

<sup>8</sup> <http://hibernate.org/orm/>

<sup>9</sup> <http://www.postgresql.org/>

Quadro 2 - Componentes desenvolvidos

Componente	Nome atribuído
Repositório de arquétipos	ArchRepo
Serviço web de acesso ao repositório de arquétipos	ArchMan ( <i>Archetype Manager</i> )
Repositório de RES	EHRRepo
Serviço web de acesso ao repositório de RES	EHRMan ( <i>EHR Manager</i> )
Repositório de regras	RulesRepo
Serviço web de acesso ao repositório de regras	RulesMan ( <i>Rules Manager</i> )

Fonte: o autor, 2014.

### 3.2.1 Manipulação de arquétipos

Na primeira etapa foi desenvolvido o repositório de arquétipos – ArchRepo – para a manipulação, armazenamento e recuperação dos arquétipos que serão utilizados neste RES. Neste projeto, o repositório foi implementado por meio do SGBD relacional PostgreSQL que possui a funcionalidade de busca por expressões regulares, necessária para a recuperação de arquétipos especificados em *slots*.

Em seguida, foi desenvolvido o serviço web responsável pela comunicação com o repositório, o ArchMan. Este serviço possibilita a realização das operações de inclusão, atualização e recuperação de arquétipos, bem como a recuperação de informações relacionadas a estes.

Nesta etapa, foi utilizada a implementação de referência do RM, disponibilizado pela Fundação openEHR<sup>10</sup>, para a validação do arquétipo a ser manipulado e extração de informações para a indexação, como palavras-chave, conceito entre outras. Neste repositório, além do conteúdo do arquétipo são armazenados os metadados referentes a direitos autorais, versão da ADL, dados do RM relacionados, detalhes e palavras-chave no idioma original e para cada tradução existente, entre outros.

Para inserir os arquétipos no repositório por meio do ArchMan, de forma que estes ficassem disponíveis para testes no sistema, foi utilizado o *plugin* Advanced Rest Client Application<sup>11</sup>, versão 3.1.7, para o navegador Google Chrome, que permite a utilização de serviços REST diretamente do navegador, incluindo o envio de informações no corpo das requisições. No caso do ArchMan, para disponibilizar um arquétipo é necessário submeter uma requisição do Protocolo de Transferência de Hipertexto (HTTP – *Hypertext Transfer Protocol*) POST ao serviço, com o

<sup>10</sup> <http://www.openehr.org/wiki/display/projects/Java+Project+Download>

<sup>11</sup> <http://chromerestclient.appspot.com/>

conteúdo do arquétipo em ADL sendo enviado no corpo da requisição. Este *plugin* foi utilizado de forma a facilitar a disponibilização dos arquétipos necessários, sem a necessidade de desenvolvimento de um sistema apenas para o gerenciamento de arquétipos, considerando que este não é o foco desta pesquisa.

Com o ArchRepo e ArchMan desenvolvidos, foi iniciado o desenvolvimento do componente responsável pela geração das interfaces para entrada de dados, a partir dos arquétipos disponibilizados pelo serviço.

### **3.2.2 Geração de interfaces para entrada de dados**

A etapa seguinte contou com o desenvolvimento do componente responsável pela geração da interface para a entrada de dados. Desenvolvido como um sistema web, utilizando Java Server Faces (JSF) 2, este componente comunica-se com o ArchRepo por meio do ArchMan descrito anteriormente.

Inicialmente, estudou-se a utilização da implementação de referência do RM utilizada também no componente de manipulação de arquétipos, porém foi necessário desenvolver uma implementação do RM adequada ao uso em sistemas web como o desenvolvido.

Esta necessidade surgiu devido à ausência de métodos padronizados para acesso aos atributos, também chamados de *getters* e *setters*, em algumas das classes da implementação de referência disponibilizada pela Fundação openEHR, necessários em uma aplicação web JSF, bem como à abordagem da implementação utilizada onde diversas classes somente podem ser instanciadas caso já possuam informações contextuais, como a qual indivíduo pertencem ou quais os valores correspondentes a cada atributo. Como na entrada de dados as informações ainda são desconhecidas, se fez necessário desenvolver uma implementação do RM que permitisse a atribuição destas informações a qualquer momento.

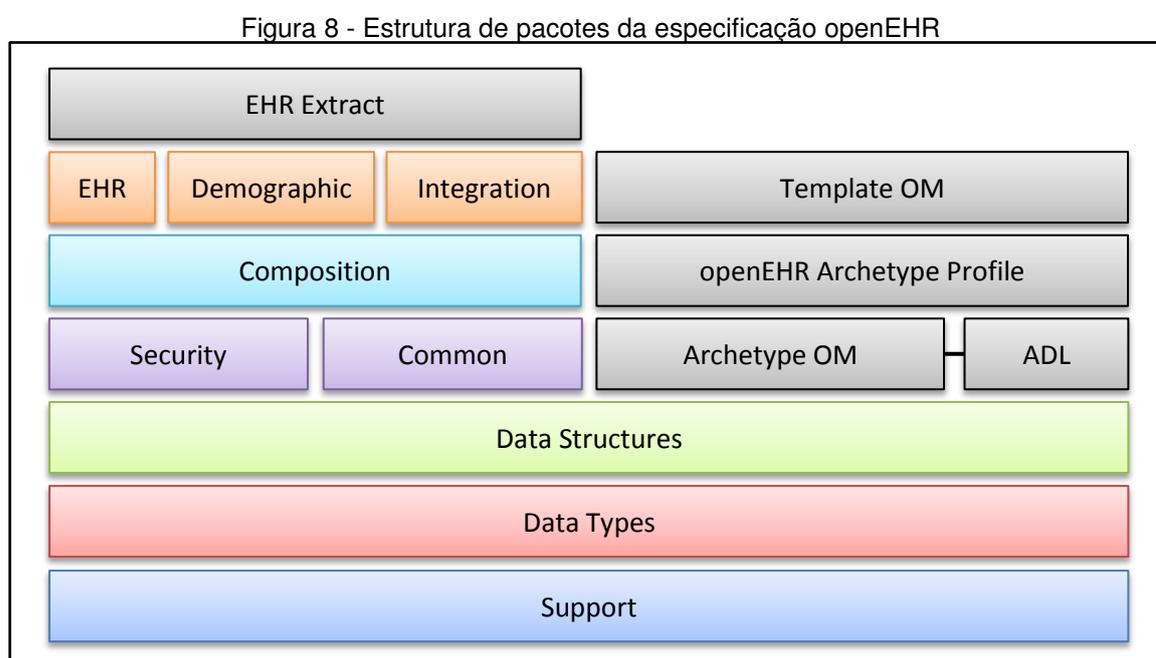
Desta forma, baseado na especificação openEHR Release 1.0.2<sup>12</sup>, foram implementadas as classes necessárias para o desenvolvimento do sistema, com o mínimo de alterações e atendendo aos requisitos descritos nos documentos da especificação, dando origem a uma implementação parcial do RM, utilizada neste trabalho. Classes que não são necessárias à execução das regras de decisão, por

---

<sup>12</sup> <http://www.openehr.org/programs/specification/releases/1.0.2>

exemplo, as utilizadas para a troca de extrato de RES, não foram implementadas neste momento.

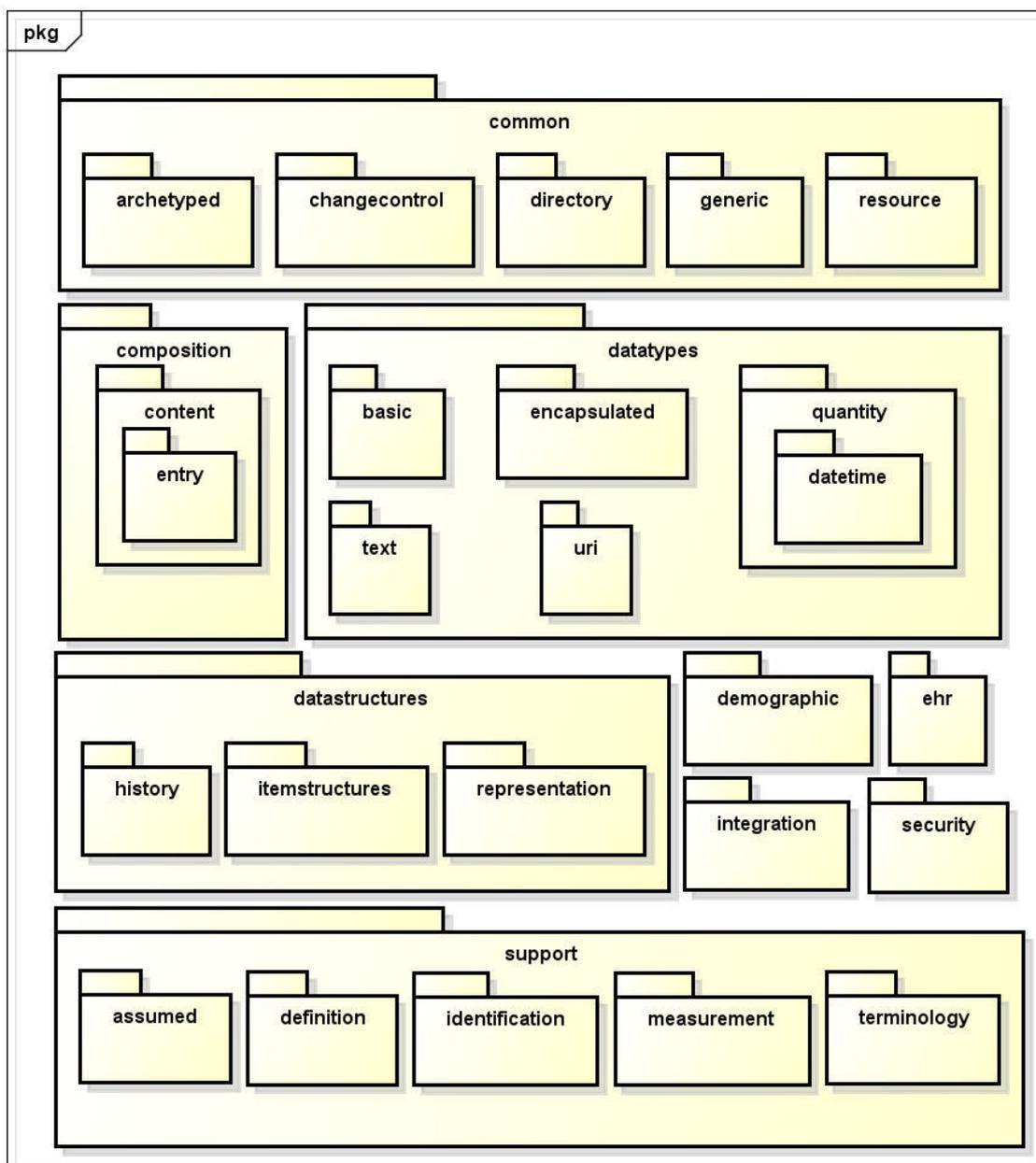
A Figura 8 ilustra as dependências entre os principais pacotes da especificação, sendo que estas existem apenas dos pacotes superiores com os inferiores. Por exemplo, o pacote *Data Structures* depende dos pacotes *Data Types* e *Support*, porém o pacote *Data Types* depende apenas do pacote *Support*. Os pacotes apresentados em cinza não foram implementados, pois as classes destes não foram necessárias para a execução das regras.



Fonte: adaptado de Beale e Heard (2008).

A Figura 9 apresenta um diagrama de classes contendo os pacotes da especificação openEHR implementados no modelo de referência do projeto. Alguns dos pacotes apresentados não possuem classes, mas subpacotes onde estão as classes do modelo, como por exemplo, o pacote *Commons*, que possui os subpacotes *archetyped*, *changecontrol*, *directory*, *generic* e *resources*. Para esta pesquisa, foi necessário implementar classes de 27 pacotes. O Apêndice A apresenta o diagrama de classes contendo as 127 classes implementadas do RM.

Figura 9 - Pacotes do modelo de referência implementado



Fonte: o autor, 2014.

Visando a persistência das informações, esta implementação do RM possui suporte à representação das informações em formato de Notação de Objetos JavaScript (JSON – *JavaScript Object Notation*), um formato em texto plano utilizado para a troca de informações e com suporte disponível em Java, por meio de bibliotecas. Podem ser citados como alternativas, outros formatos de representação, como XML, o formato de dados da ADL (dADL – *Data ADL*) e até mesmo realizado o armazenamento das informações em um modelo relacional de acordo com as entidades contidas na especificação.

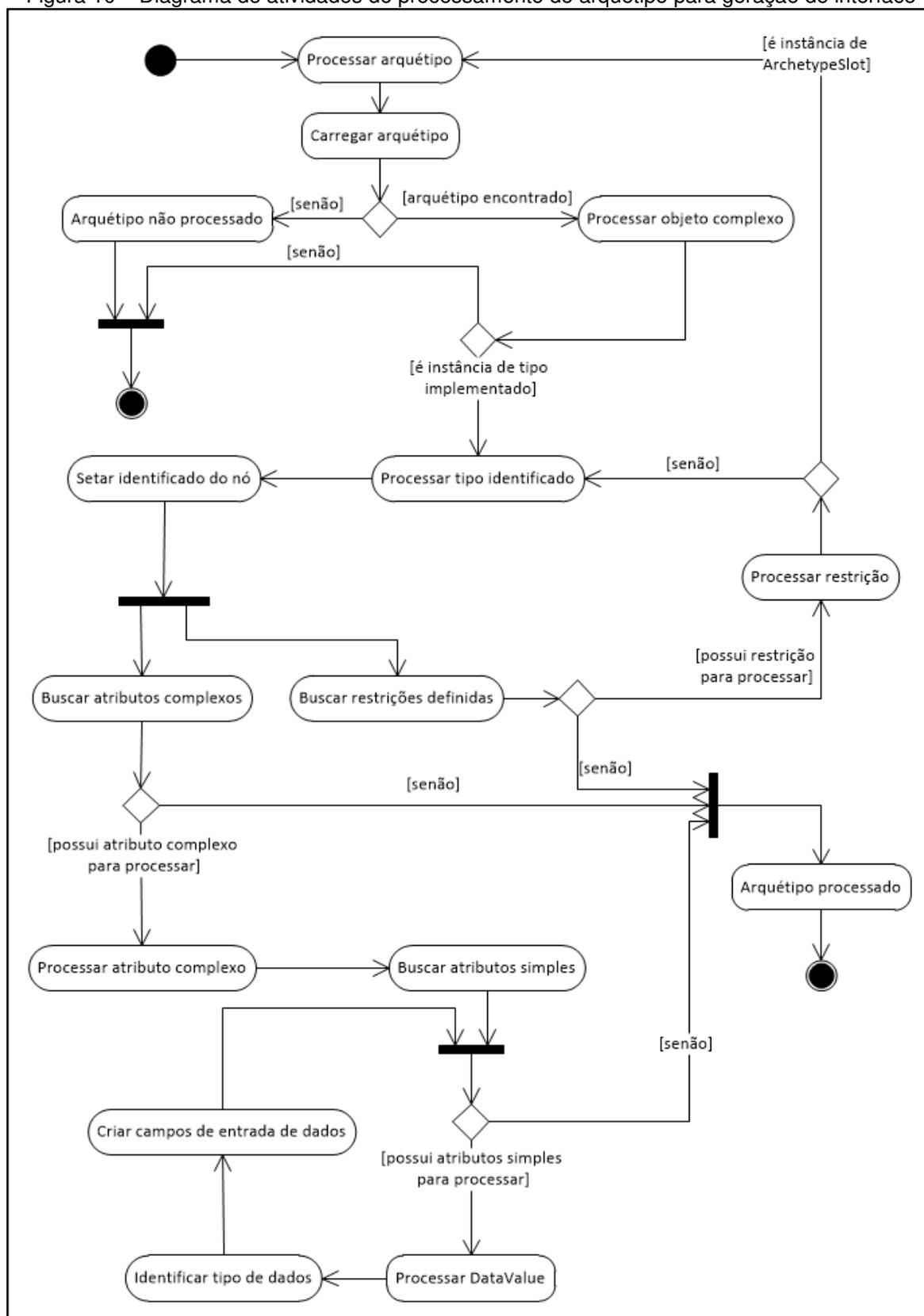
Contudo, optou-se pela utilização de JSON devido à facilidade no uso e por possuir como característica, uma organização hierárquica dos dados representados por este formato, a mesma presente no modelo de dois níveis. Desta forma, foram implementados métodos que permitem representar as entidades do RM em formato JSON, com seus respectivos atributos, bem como a recuperação das entidades e seus atributos a partir de objetos JSON. Contudo, outros formatos como o XML ou o modelo relacional poderiam ser utilizados para esta finalidade.

Com um modelo de referência adaptado ao tipo de sistema desenvolvido, foi iniciada então, a implementação do componente de geração de interfaces para a entrada de dados. Em um primeiro momento, foi desenvolvida uma interface simples para listar os arquétipos disponíveis no ArchRepo, obtidos por meio do ArchMan, permitindo ao usuário selecionar o arquétipo para o processamento e a geração da interface.

O processamento dos arquétipos para a geração da interface baseada em sua definição foi desenvolvido em dois passos que ocorrem paralelamente: leitura da definição e atributos do arquétipo, utilizando a implementação de referência da especificação openEHR para a geração dos componentes visuais; e instanciação dos respectivos objetos a partir do RM desenvolvido no projeto para a vinculação destes com os componentes visuais gerados.

No primeiro passo são lidos os atributos do arquétipo solicitado e são gerados componentes visuais para a entrada de dados de acordo com o tipo de cada atributo. Por exemplo, para o tipo de dados texto livre, é gerado um campo de texto simples – *inputText*; para tipo de dados numérico inteiro, é gerado um campo que aceita apenas dígitos – *inputText* com restrição; para o tipo de dados de seleção única, é gerado um campo com a lista das opções – *selectOneMenu*. A Figura 10 apresenta o diagrama de atividades correspondente a este passo.

Figura 10 – Diagrama de atividades do processamento do arquétipo para geração de interface



Fonte: o autor, 2014.

Tipos de dados complexos, por exemplo, o tipo proporção que é composto pelos dados do numerador, denominador e unidade, resultam na geração de um conjunto de campos para cada um dos dados que compõem o tipo complexo.

No caso dos *slots* para arquétipos, uma requisição de recuperação de arquétipo é enviada para o ArchMan, com o identificador ou expressão regular dos identificadores dos arquétipos aceitos. O arquétipo retornado é então processado e são gerados os componentes visuais para a entrada de dados, de acordo com o processo descrito anteriormente.

Juntamente com a geração dos campos para a entrada de dados, são instanciados os respectivos objetos – segundo passo – que armazenarão os dados informados pelo usuário, mantendo a estrutura hierárquica do arquétipo de origem e permitindo, ao final, a representação em JSON dos dados coletados.

Após a entrada dos dados na interface gerada, o usuário pode submeter estes ao serviço web desenvolvido na etapa seguinte – o EHRMan – responsável pela persistência e recuperação dos RES comunicando-se com o repositório de RES – EHRRepo – também desenvolvido na etapa seguinte.

O componente que processa os arquétipos e os dados coletados classifica os arquétipos em dois grandes grupos: arquétipos demográficos e arquétipos clínicos e administrativos. O primeiro grupo define informações demográficas, como pacientes, pessoas, profissionais, equipamentos, locais entre outros. O segundo grupo especifica o conteúdo clínico e administrativo que pode ser coletado.

Considerando a classificação citada, os arquétipos que podem ter uma interface gerada para a coleta dos dados são:

- Demográficos: somente arquétipos que possuem como entidade raiz uma especialização de PARTY, que é a classe raiz que representa as entidades do mundo real, como por exemplo, PERSON para pessoas, ORGANISATION para organizações e GROUP para grupos.
- Clínicos e administrativos: apenas arquétipos com entidade raiz COMPOSITION ou qualquer especialização de ENTRY, como OBSERVATION para observações, INSTRUCTION para instruções e ADMIN\_ENTRY para dados administrativos.

Esta restrição nos tipos de arquétipos que podem ser processados como raiz para geração de interfaces para a coleta de dados deve-se ao fato de que as demais

entidades são utilizadas para compor as informações contidas em *compositions*, *entries* e arquétipos demográficos, sendo utilizados como referência nos *slots* dentro destes arquétipos. Desta forma, quando utilizados de forma isolada, não possuem significado próprio e contexto associado.

As instâncias de dados demográficos são enviadas ao EHRMan exatamente como são coletadas. Caso sejam novos registros, é gerado um identificador único antes de submeter ao EHRRepo.

No caso dos dados clínicos e administrativos, para que estes sejam enviados ao EHRMan, a instância da entidade representada pelo arquétipo deve possuir um vínculo com o indivíduo ao qual se referem os dados. Desta forma, fez-se necessário disponibilizar também na interface gerada, uma forma de seleção dos pacientes previamente cadastrados no sistema, por meio dos arquétipos classificados como demográficos.

Como a referência ao indivíduo está presente apenas nas especializações de ENTRY e em EHR<sup>13</sup>, de forma a possibilitar a coleta de informações incluindo também os arquétipos de integração, os dados clínicos e administrativos são incluídos em uma *composition* que é então enviada ao serviço de dados de RES, juntamente com o identificador do paciente selecionado.

Após a entrada dos dados e ajustes necessários na estrutura, é realizada uma solicitação de armazenamento dos dados informados, no repositório de RES – EHRRepo – por meio do serviço de RES – EHRMan – desenvolvido na etapa seguinte. Para isso, o objeto criado para armazenar os dados informados pelo usuário gera a representação JSON dos dados que é submetida ao serviço correspondente.

### **3.2.3 Manipulação dos dados do RES**

A próxima etapa consistiu no desenvolvimento do EHRRepo, também implementado por meio do uso do SGBD relacional PostgreSQL versão 9.2. Este SGBD foi escolhido para esta etapa por ser gratuito, atender às necessidades do sistema, de armazenamento das informações, estar disponível para a plataforma utilizada no desenvolvimento deste trabalho e por permitir a sua utilização com a

---

<sup>13</sup> Na implementação realizada, EHR refere-se à classe EHR do modelo de referência e não à abreviatura da tradução de RES para o inglês (*Electronic Health Record*).

tecnologia empregada no desenvolvimento (Java) por meio de bibliotecas específicas.

A comunicação entre o sistema e o repositório de RES também é realizada por meio de um serviço web específico, desenvolvido utilizando o estilo REST, nos mesmos moldes do serviço de manipulação de arquétipos. Este serviço provê formas de armazenar e recuperar dados de RES, utilizando o formato JSON como entrada e retornando os dados também em formato JSON.

A partir dos dados enviados ao EHRMan, o serviço identifica a entidade correspondente e possíveis mapeamentos existentes para indexação. No caso de entidades correspondentes aos pacientes, alguns dados, como nome, data de nascimento entre outros, são armazenados em colunas específicas no banco de dados (BD) para facilitar a busca dos pacientes.

Para realizar este mapeamento, foram desenvolvidas duas classes auxiliares para representar o mapeamento de arquétipos específico para entidades do BD. Estas classes possibilitam especificar o identificador do arquétipo, o nome da classe correspondente à entidade do BD e os mapeamentos entre os atributos do arquétipo e os atributos da classe.

A partir dos mapeamentos definidos, por meio de método responsável pela busca de nós na estrutura hierárquica de arquétipos a partir de um caminho, cada mapeamento de atributo é processado e os dados são atribuídos aos atributos da classe por meio de reflexão. Os objetos resultantes são então enviados para o framework de persistência utilizado que persiste os dados mapeados em suas respectivas tabelas.

Apenas para fins de testes e validação da funcionalidade, foi definido apenas um mapeamento, descrito anteriormente, para o arquétipo de pacientes e realizado o carregamento de forma estática. Contudo, a estrutura desenvolvida permite que sejam realizados mapeamentos em fontes externas, como arquivos XML, e processados dinamicamente pelo EHRRepo.

Após a implementação do serviço de manipulação dos dados do RES, foi realizado o desenvolvimento dos componentes necessários para a manipulação das regras de decisão especificadas em GDL.

### 3.2.4 Manipulação de regras

Nesta etapa foram desenvolvidos os componentes responsáveis pela manipulação das regras, utilizando a abordagem de repositórios, aplicada aos componentes anteriores. Este repositório também foi implementado por meio de um SGBD PostgreSQL, onde é armazenado o conteúdo dos arquivos GDL que contêm as regras extraídas das diretrizes e metadados destas, de forma a facilitar a busca de regras. Esta etapa contou com a criação de 5 tabelas no BD:

- Guidelines: tabela onde são armazenadas as informações principais das diretrizes, como identificador, que deve ser único no repositório, idioma original da diretriz, versão da GDL utilizada e informações de criação dos dados;
- Guidelines\_Content: armazena o conteúdo do arquivo GDL associado ao seu identificador, possuindo relacionamento com a tabela Guidelines;
- Guideline\_Details: tabela que armazena os detalhes das diretrizes, no idioma original da diretriz e em todos os idiomas para os quais a diretriz foi traduzida. Possui relacionamento com a tabela Guidelines e, de forma a permitir uma identificação única, é utilizado, além do identificador da diretriz, o idioma do registro como identificador composto;
- Guidelines\_Keywords: tabela que armazena as palavras-chave definidas na diretriz, no idioma original e em todos os idiomas para os quais a diretriz foi traduzida. Relaciona-se com a tabela Guideline\_Details, adicionando a própria palavra-chave como identificador, além do identificador da tabela Guideline\_Details;
- Guidelines\_Archetypes\_Mappings: nesta tabela são armazenados os identificadores de todos os arquétipos mapeados na diretriz, nos antecedentes ou nos consequentes das regras especificadas.

Utilizando as tabelas descritas, neste repositório são armazenadas as diretrizes especificadas em GDL bem como seu conteúdo, mapeamentos para arquétipos, detalhes por idioma e palavras-chave. Estas informações permitem a

busca de diretrizes e regras pelo identificador da diretriz bem como pelos arquétipos que possam ser afetados pela mesma.

O acesso a este repositório é realizado por meio de um serviço web desenvolvido nesta etapa que permite o armazenamento e recuperação de regras. Para o armazenamento, este serviço recebe o conteúdo dos arquivos GDL, realiza a análise deste utilizando as bibliotecas disponíveis com o GDL Tools versão 0.91 e extrai metadados como palavras-chave em todos os idiomas definidos no arquivo, identificador da diretriz e arquétipos utilizados, entre outras, armazenando-os no repositório.

A obtenção de regras e metadados é realizada por meio de requisições a um Localizador Padrão de Recursos (URL – *Uniform Resource Locator*) do serviço, de acordo com o tipo de informação desejada (regras ou metadados), com o identificador do recurso. Estas regras são disponibilizadas ao RES que, com após a integração com o motor de inferência, é capaz de processá-las utilizando os dados estruturados em arquétipos. Nesta etapa também foi utilizado o *plugin* Advanced Rest Client Application para o envio de regras ao repositório por meio do RulesMan, considerando que o foco desta pesquisa não consiste em gerenciar as regras disponibilizadas no repositório.

Com a finalização da implementação dos componentes necessários, foram realizados testes a fim de validar a implementação realizada.

### **3.2.5 Testes dos componentes desenvolvidos**

Após a implementação dos componentes do sistema, foram realizados testes com arquétipos disponíveis no repositório da Fundação openEHR, o openEHR *Clinical Knowledge Manager*<sup>14</sup> (CKM), arquétipos desenvolvidos por Cintho (2014) e também arquétipos no formato ISO/EN 13.606 obtidos a partir do Portal Público da Base de Registro Eletrônico em Saúde (B-RES) do Estado de Minas Gerais (MG)<sup>15</sup>, apresentados no Quadro 3.

Estes testes objetivaram validar o sistema a respeito da manipulação dos arquétipos por meio do serviço desenvolvido, à geração das interfaces baseadas na

---

<sup>14</sup> <http://www.openehr.org/ckm>

<sup>15</sup> <http://sres.saude.mg.gov.br/>

definição dos arquétipos e ao armazenamento dos dados coletados, também por meio do serviço responsável pela comunicação com o repositório de RES.

Quadro 3 - Arquétipos utilizados para testes do sistema desenvolvido

Arquétipo	Modelo	Fonte
openEHR-EHR-CLUSTER.device.v1	openEHR	openEHR CKM
openEHR-EHR-CLUSTER.level_of_exertion.v1	openEHR	openEHR CKM
openEHR-EHR-COMPOSITION.report.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-ADDRESS.address.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-ADDRESS.electronic_communication.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-CLUSTER.birth_data_additional_detail_br.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-CLUSTER.person_identifier.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-ITEM_TREE.person_details.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-PARTY_IDENTITY.person_name.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-PERSON.person.v1	openEHR	openEHR CKM
openEHR-DEMOGRAPHIC-PERSON.person-patient.v1	openEHR	openEHR CKM
openEHR-EHR-EVALUATION.family_history.v1	openEHR	openEHR CKM
openEHR-EHR-EVALUATION.problem-diagnosis.v1	openEHR	GDL Tools
openEHR-EHR-OBSERVATION.basic_demographic.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.blood_pressure.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.blood_pressure_alteration.v1		
openEHR-EHR-OBSERVATION.body_mass_index.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.body_temperature.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.body_weight.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.height.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.lab_test-blood_glucose.v1	openEHR	openEHR CKM
openEHR-EHR-OBSERVATION.res_drc_prevrim_a.v1	openEHR	Cintha (2014)
openEHR-EHR-OBSERVATION.res_drc_prevrim_a.v2	openEHR	Cintha (2014)
openEHR-EHR-OBSERVATION.symptoms_list.v1	openEHR	Cintha (2014)
CEN-EN13606-ENTRY.AntecedentesObstetricos.v2	ISO/EN 13606	B-RES MG
CEN-EN13606-ENTRY.AvaliacaoFuncional.v2	ISO/EN 13606	B-RES MG
CEN-EN13606-ENTRY.ImpressaoDiagnostica.v2	ISO/EN 13606	B-RES MG

Fonte: o autor, 2014.

Também foram realizados testes com regras de diretrizes especificadas em GDL, utilizando as regras desenvolvidas por Cintha (2014) e as disponibilizadas como exemplos juntamente com a GDL Tools. Nesta fase, apenas a análise do conteúdo, armazenamento e recuperação das regras foram testados. A execução

das mesmas foi realizada na fase seguinte, onde foi realizada a integração do motor de inferência e regras ao RES.

### 3.3 FASE III – INTEGRAÇÃO E APLICAÇÃO DO MOTOR DE INFERÊNCIA

A execução de regras desacopladas do sistema requer um componente específico, chamado de motor de inferência. Nesta pesquisa foi utilizado o Drools Expert, por ser de uso livre e desenvolvido em Java, o que facilita a integração deste ao RES desenvolvido.

Para a utilização do motor de inferência é necessário que sejam disponibilizadas as regras que irão compor a base de conhecimento do mesmo, os dados sobre os quais serão realizadas as inferências, chamados de fatos, bem como definida a forma de execução das regras, podendo ser com ou sem estado mantido. Desta forma, pode-se dividir a integração e aplicação do motor de inferência em 4 passos: construção da base de conhecimento, a partir das regras; inserção dos fatos na base de conhecimento; execução das regras e atualização dos dados; e integração do motor de inferência ao RES.

#### 3.3.1 Construção da base de conhecimento

A construção da base de conhecimento requer inicialmente a definição da forma de execução das regras que pode ser mantendo ou não o estado da execução.

Na primeira forma, também chamada de *stateful*, após as regras serem executadas sobre os fatos e estes terem seu valor alterado, a base de conhecimento mantém as alterações realizadas para as execuções seguintes, bem como todos os fatos inseridos e não atualizados ou fatos criados em decorrência de alguma regra.

A segunda forma, chamada de *stateless*, permite que as regras sejam executadas, os fatos alterados em decorrência destas, porém o estado destes não é mantido entre uma execução e outra. Isto é, em uma nova execução, é necessário inserir novamente os fatos na base de conhecimento.

Nesta pesquisa, foi definida a utilização de uma base de conhecimento *stateless*, onde após a execução das regras sobre os dados estruturados em arquétipos e sua manipulação, a base de conhecimento não deverá manter o estado, sendo que, para as execuções seguintes, novos dados são inseridos nesta.

Com a definição da forma de execução, devem ser disponibilizadas ao motor de inferência as regras que irão compor a base de conhecimento. O Drools Expert permite que as regras sejam disponibilizadas em alguns formatos, entre eles, tabelas de decisão em planilhas eletrônicas e regras definidas em DRL. Considerando que o repositório de regras desenvolvido utiliza a GDL e que a GDL Tool permite, por meio de suas bibliotecas, a geração de regras em DRL a partir de uma diretriz, a base de conhecimento, nesta pesquisa, é criada a partir das regras em DRL.

A partir dos dados estruturados em arquétipos, o identificador do arquétipo raiz é obtido e enviado ao RulesMan uma requisição para que sejam enviados os identificadores de todas as diretrizes que possuem o arquétipo mapeado.

Para cada diretriz retornada, uma nova requisição ao RulesMan é realizada, desta vez solicitando o conteúdo da diretriz, em GDL. Em seguida, foi desenvolvida uma classe utilitária, responsável pelo carregamento dos arquétipos necessários, dos dados disponibilizados e invocação do conversor de regras da GDL Tools.

Esta classe realiza a análise do conteúdo da diretriz e, a partir da definição instanciada da diretriz, são processadas as informações dos arquétipos mapeados. Para cada um dos arquétipos mapeados, é realizada uma requisição ao ArchRepo para a recuperação do conteúdo do arquétipo. Este conteúdo é analisado pelas bibliotecas da implementação de referência da openEHR e todos os atributos dos tipos ELEMENT, CLUSTER, CODED\_TEXT, ORDINAL e SLOT são armazenados em memória, com seu respectivo caminho, pois estes são utilizados na definição das regras em GDL. Caso algum arquétipo não esteja disponível no ArchRepo, a diretriz não é executada, pois não é possível prover todos os elementos definidos nas regras, e o sistema alerta ao usuário a ausência do referido arquétipo.

Em seguida é realizada a conversão do conteúdo em GDL para DRL e uma nova base de conhecimento é instanciada com as regras contidas. No passo seguinte são buscados os fatos sobre os quais serão executadas as regras.

### **3.3.2 Inserção dos fatos na base de conhecimento**

De acordo com os atributos do arquétipo armazenados em memória, são buscados os valores informados pelo usuário, por meio do método de busca definido na especificação openEHR utilizada – o método `itemAtPath`, implementado como `getItemAtPath`, de acordo com as convenções Java – e são inseridas cópias de

todos estes, na base de conhecimento instanciada, como os fatos sobre os quais as regras serão executadas.

Contudo, inicialmente apenas os dados que estão sendo coletados são enviados para a execução das regras. Desta forma, faz-se necessário a busca por dados já armazenados no EHRRepo para a execução. A partir dos arquétipos mapeados na diretriz, são realizadas requisições de recuperação do último registro do paciente associado aos dados enviados, para cada arquétipo mapeado. Caso encontrado, todos os atributos do arquétipo, armazenados em memória pelo passo anterior, são recuperados e uma cópia de cada um dos atributos é inserida como um fato na base de conhecimento.

### **3.3.3 Execução das regras e atualização dos dados**

Após o processamento de todos os arquétipos mapeados e disponíveis, as regras são então executadas. Este processo pode atualizar os fatos inseridos ou criar novos fatos, de acordo com as regras definidas na diretriz.

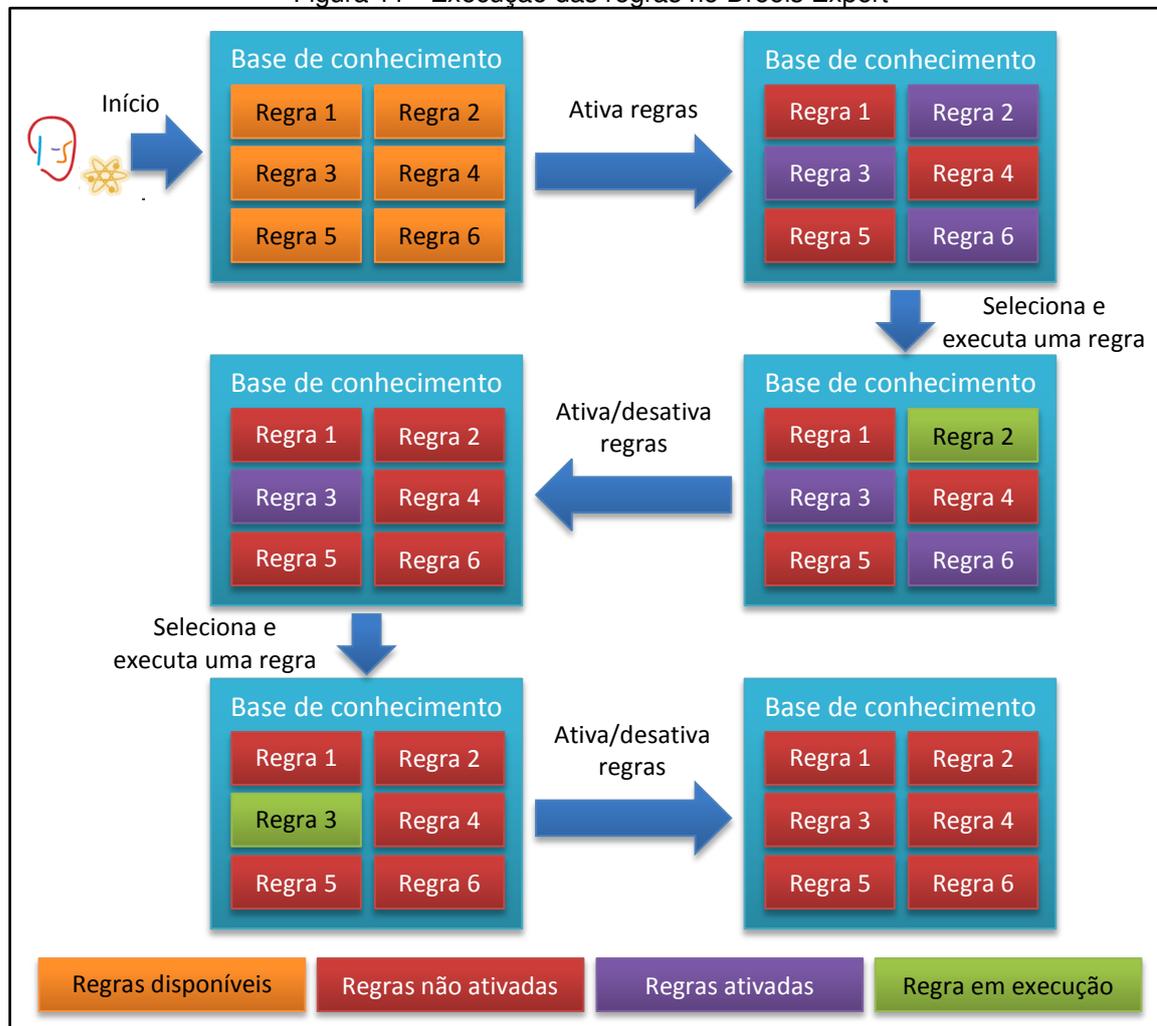
O motor de inferência utilizado nesta pesquisa, o Drools Expert, permite que mais de uma regra seja executada durante um mesmo processo. Na Figura 11, é apresentado, de forma simplificada, o funcionamento da execução. A partir inferência realizada com as regras disponíveis na base de conhecimento, um primeiro ciclo é iniciado, onde as regras que são aplicáveis aos fatos inseridos são ativadas para execução. Em seguida, de acordo com a prioridade das regras, uma delas é selecionada e executada, atualizando os fatos ou criando novos fatos.

Com a execução de uma regra, todas as regras disponíveis são reavaliadas e regras que estavam ativadas para a execução podem ser desativadas, se não forem mais aplicáveis aos fatos disponíveis. Regras que anteriormente não eram aplicáveis aos fatos, mas que com o novo estado dos fatos passaram a ser aplicáveis, são ativadas para o próximo ciclo de execução. Estes ciclos são executados até que nenhuma regra seja ativada para a execução. De forma a impedir execução cíclica de regras, este motor de inferência possui mecanismos de detecção de recursividade.

A forma de execução das regras neste motor de inferência favorece a utilização deste no suporte a decisão baseado em diretrizes, pois permite que mais de uma regra seja aplicada aos mesmos dados, possibilitando que sejam

apresentadas sugestões, por exemplo, para um tratamento de determinado indivíduo, conforme apresentado por Garcia et al. (2013).

Figura 11 - Execução das regras no Drools Expert



Fonte: o autor, 2014.

Após a execução das regras, os fatos inseridos anteriormente são comparados com os dados originais de forma a identificar as alterações realizadas pelas regras. Caso existam, os dados são atualizados na interface de entrada de dados, em seus respectivos componentes visuais, para que o usuário possa conferir as informações antes de prosseguir.

Os novos fatos criados pela execução das regras são disponibilizados para o usuário consultar e submeter ao EHRMan para que sejam armazenados no EHRRepo, juntamente com os dados informados na tela.

Caso algum dado já armazenado no EHRRepo seja alterado pela execução das regras, estes também são disponibilizados para o usuário verificar e, caso este decida armazená-los no EHRRepo, um novo registro é gerado, considerando que os dados de saúde não podem ser alterados, apenas complementados.

### **3.3.4 Integração do motor de inferência ao RES**

O passo final desta fase consiste na integração do motor de inferência ao RES, representado nesta pesquisa pela disponibilização, na interface gerada pelo componente específico, de uma forma de identificação das regras que se aplicam ao arquétipo o qual foi gerado a interface, bem como a execução das mesmas e o retorno dos resultados ao usuário.

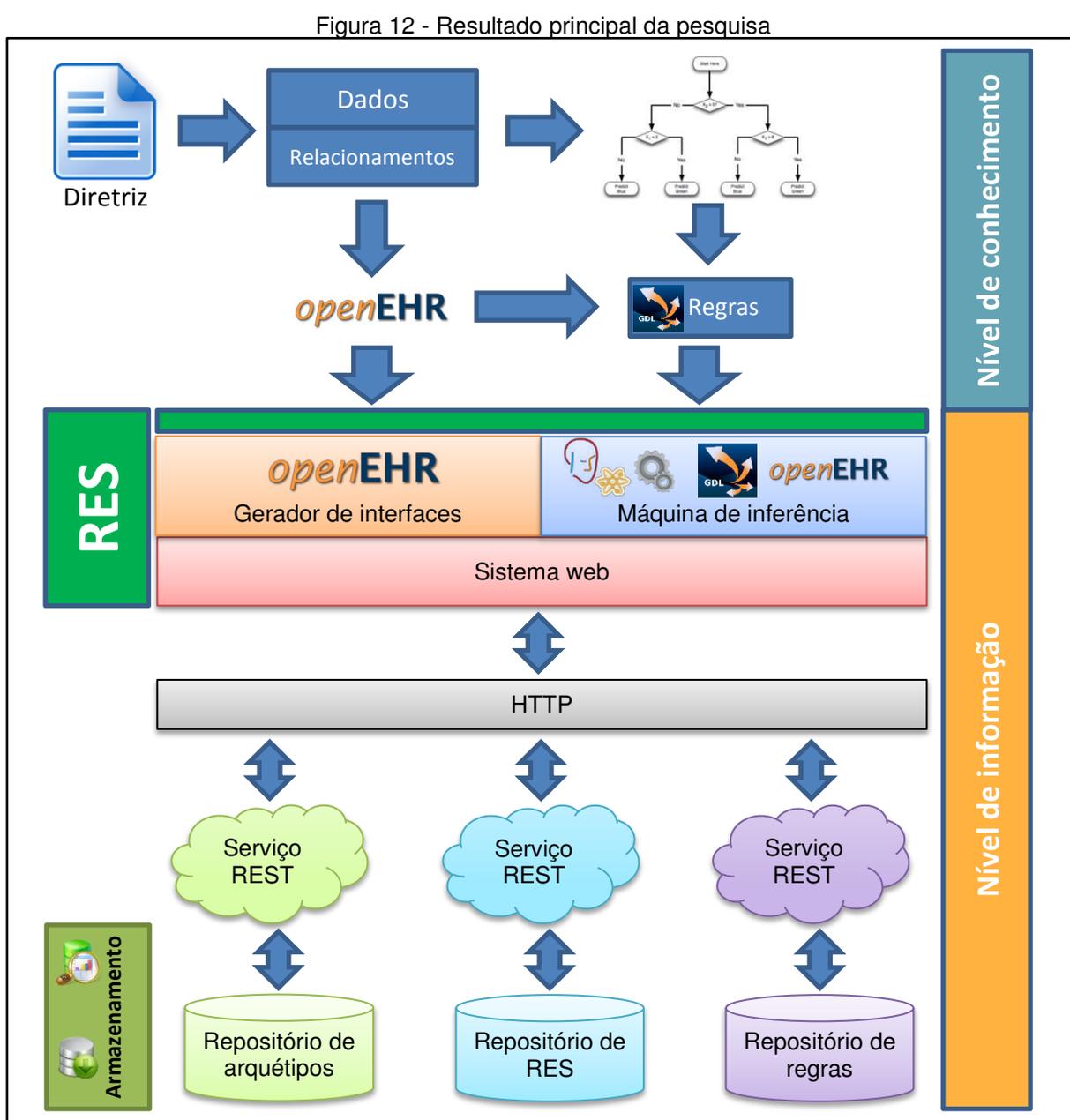
Desta forma, após a geração da interface para o arquétipo selecionado, é realizada uma requisição ao RulesMan informando o identificador do arquétipo e, caso sejam encontradas diretrizes definidas para este, é apresentada uma notificação na tela, informando da existência de diretrizes disponíveis para execução, possibilitando ao usuário a execução destas, por meio de um botão disponibilizado na interface.

Considerando que o SAD deve fornecer o apoio a decisão, porém cabe aos profissionais aceitarem ou rejeitarem os resultados providos pela inferência realizada, após a execução das regras baseadas nas diretrizes, o resultado é disponibilizado na própria interface gerada e os dados somente são enviados ao EHRMan para serem armazenados no EHRRepo, a partir do comando do usuário do sistema.

Após a implementação de todos os componentes necessários, o RES é capaz de fornecer suporte a decisão por meio dos dados padronizados pelos arquétipos, sendo necessária, para a finalização desta pesquisa, a validação do SAD integrado ao RES. Desta forma, foram inseridos dados fictícios na interface gerada, para os arquétipos mapeados em diretrizes e os resultados foram comparados com os esperados de acordo com a definição na diretriz de origem.

## 4 RESULTADOS

O principal resultado desta pesquisa consiste na integração de diretrizes clínicas especificadas em formato computacional, com sua definição baseada na estrutura de arquétipos em um RES também baseado em arquétipos, provendo o suporte a decisão, conforme apresentado na Figura 12, que complementa o método e o modelo de desenvolvimento propostos, mostrados na Figura 5 e na Figura 6, respectivamente.



Fonte: o autor, 2014.

Entretanto, para que este resultado fosse obtido, esta pesquisa contou também com as seguintes contribuições com grande importância: implementação parcial da especificação do modelo openEHR, adequado ao desenvolvimento de uma aplicação JSF; modelo para o desenvolvimento de um RES baseado em arquétipos; geração automática de interface para a coleta de dados estruturados em arquétipos; forma de armazenamento de dados estruturados em arquétipos; e execução de regras de decisão baseadas em diretrizes, em um motor de inferência.

#### 4.1 IMPLEMENTAÇÃO PARCIAL DA ESPECIFICAÇÃO DO MODELO OPENEHR

O desenvolvimento do RES como uma aplicação web JSF necessitou que fosse implementado o RM da especificação openEHR utilizada, com base nos padrões estabelecidos para este tipo de aplicação, no que diz respeito à nomenclatura de métodos, forma de acesso aos atributos de um objeto e possibilidade de definição de valores para estes atributos a qualquer momento.

Com isto, obteve-se um dos resultados desta pesquisa, que foi a implementação parcial do RM, realizada em Java, contando com as classes necessárias para o desenvolvimento desta pesquisa. O desenvolvimento deste RM baseou-se na implementação de referência do RM openEHR, disponibilizado pela Fundação openEHR, utilizando-se de trechos de códigos desta implementação.

Conforme apresentado no método desta pesquisa, apenas as classes necessárias foram implementadas. Métodos que também não possuem utilidade para a execução das regras, nesta pesquisa, também não foram implementados.

Ao todo, foram implementadas classes de 27 pacotes que compõem os 8 modelos de informação (IM – *Information Model*) da especificação listados no Quadro 4, baseados nos documentos da especificação openEHR, *Release 1.0.2* (BEALE, 2006; BEALE et al., 2008a, 2008b, 2008c, 2008d, 2008e, 2008f).

Quadro 4 - Modelos de Informação implementados na pesquisa

IM de RES	IM Demográfico
IM Comum	IM de Estruturas de Dados
IM de Tipos de Dados	IM de Suporte
IM de Integração	IM de Segurança

Fonte: o autor, 2014.

Os pacotes implementados dos IM listados são apresentados no Quadro 5.

Quadro 5 - Pacotes da especificação openEHR, implementados na pesquisa

EHR	Composition	Content
Entry	Demographic	Archetyped
Generic	Directory	Change Control
Resource	Data Structure	Item Structure
Representation	History	Basic
Text	Quantity	Date Time
Encapsulated	Uri	Assumed
Identification	Terminology	Measurement
Definition	Integration	Security

Fonte: o autor, 2014.

Esta seção não tem como intenção definir ou explicar em detalhes as classes da especificação implementada, apenas apresentar as classes que compõem um dos resultados desta pesquisa. A descrição das classes, apresentada a seguir, é baseada nos documentos da especificação 1.0.2 do modelo openEHR, que pode ser consultada para maiores detalhes.

A lista das classes implementadas é extensa, porém é importante apresentá-las para que se possa ter uma visão da implementação realizada no que se refere à especificação do RM openEHR. A descrição das classes foi adaptada para facilitar o entendimento, desconsiderando-se comentários presentes na especificação e detalhes que podem ser consultados nos documentos originais.

#### 4.1.1 O IM de RES e o Pacote *EHR*

O IM de RES contém os pacotes *EHR* e *Composition*, onde estão definidas as classes raízes do IM. No pacote *EHR* está definida a classe que representa o objeto principal (EHR) e informações estruturadas e versionadas, bem como informações de acesso. O Quadro 6 apresenta as 6 classes do pacote *rm.ehr* implementadas.

Quadro 6 - Classes implementadas do pacote *rm.ehr*

Classe	Descrição
EHR	Objeto raiz e ponto de acesso de um RES para um indivíduo.
VERSIONED_EHR_ACCESS	Container de versão para uma instância de EHR_ACCESS.
EHR_ACCESS	Objeto de controle de acesso para todo o registro.
VERSIONED_EHR_STATUS	Container de versão para uma instância de EHR_STATUS.
EHR_STATUS	Objeto único em um RES contendo várias

	informações sobre o RES.
VERSIONED_COMPOSITION	Abstração de uma <i>Composition</i> com versão controlada.

Fonte: adaptado de Beale et al. 2008a.

#### 4.1.2 Pacote *Composition*

Este pacote contém, entre outras, a classe COMPOSITION, considerada como o contêiner primário de dados de um RES baseado no modelo openEHR, sendo o ponto raiz do conteúdo clínico, conforme apresentado no Quadro 7.

Quadro 7 - Classes implementadas do pacote rm.composition

Classe	Descrição
COMPOSITION	Uma versão em uma VERSIONED_COMPOSITION, sendo considerada uma unidade de informações do RES.
EVENT_CONTEXT	Documenta a informação contextual de um evento do cuidado de saúde envolvendo o indivíduo e o sistema de saúde.

Fonte: adaptado de Beale et al., 2008e.

Além das classes descritas acima, este pacote contém também o subpacote *Content*, descrito a seguir.

##### 4.1.2.1 Pacote *Content*

Neste pacote está definida a classe que é a base, ou superclasse, de todas as classes que representam conteúdo clínico, apresentado no Quadro 8.

Quadro 8 - Classe implementada do pacote rm.composition.content

Classe	Descrição
CONTENT_ITEM	Classe base de todas as classes concretas que representam conteúdo do RES.

Fonte: adaptado de Beale et al., 2008e.

O pacote *Content* abrange também o pacote *Entry*, onde estão definidas as classes que são utilizadas para expressar todas as informações criadas em um RES baseado no modelo openEHR. O Quadro 9 apresenta estas classes, podendo-se citar, a classe ENTRY e suas principais especializações, OBSERVATION, EVALUATION, INSTRUCTION e ACTION.

Quadro 9 - Classes implementadas do pacote `rm.composition.content.entry`

Classe	Descrição
ENTRY	Classe abstrata, raiz de dos itens das informações clínicas criadas no contexto de declaração clínica, sendo a menor unidade que uma busca por dados no RES deve retornar.
ADMIN_ENTRY	Especialização de ENTRY para informações administrativas, sem relevância clínica.
CARE_ENTRY	Classe abstrata, utilizada como raiz para todas as classes de informações clínicas. Nesta classe são definidos atributos de protocolo e diretrizes para todas as suas subclasses.
OBSERVATION	Especialização de CARE_ENTRY para representar todos os dados clínicos passados e atuais. Utilizada para dados mensurados de alguma forma, observações e relatos.
EVALUATION	Especialização de CARE_ENTRY para declarações de avaliações.
INSTRUCTION	Especialização de CARE_ENTRY para especificar ações a serem realizadas no futuro.
ACTIVITY	Define uma única atividade em uma INSTRUCTION.
ACTION	Especialização de CARE_ENTRY para registrar ações clínicas realizadas.
INSTRUCTION_DETAILS	Classe utilizada para registrar detalhes da INSTRUCTION que causa uma ACTION.
ISM_TRANSITION	Modelo de uma transição em uma máquina de estados de INSTRUCTION.

Fonte: adaptado de Beale et al., 2008e.

#### 4.1.30 IM Demográfico e o Pacote *Demographic*

O pacote *Demographic* abrange as classes que formam o IM Demográfico genérico que contém informações esperadas em um serviço de dados demográficos, apresentadas no Quadro 10. Este IM objetiva servir como uma especificação de serviços demográficos independentes ou de um serviço intermediário para sistemas existentes como um índice mestre de pacientes.

Quadro 10 - Classes implementadas do pacote `rm.demographic`

Classe	Descrição
PARTY	Classe abstrata, base para todos os tipos de partes, incluindo entidades do mundo real e seus papéis, considerando-se neste contexto, que parte é qualquer entidade que pode participar de uma atividade.
PARTY_IDENTITY	Uma identidade de propriedade de uma PARTY, como um nome de pessoa, usado para identificar seu proprietário.

CONTACT	Descrição de uma forma de contato de uma parte.
ADDRESS	Endereço de contato, podendo ser eletrônico ou geográfico.
ACTOR	Classe base de todos os tipos do mundo real – os atores – incluindo pessoas e organizações, sendo qualquer entidade capaz de assumir um papel.
PERSON	Descrição genérica de pessoas.
ORGANISATION	Descrição genérica de organizações.
GROUP	Classe para representar grupos.
AGENT	Classe que representa o conceito genérico de qualquer tipo de agente, incluindo dispositivos, sistemas de <i>software</i> . Não são considerados como agentes, humanos e organizações.
ROLE	Descrição genérica de um papel desempenhado por um ator.
CAPABILITY	Aptidões de um papel.
PARTY_RELATIONSHIP	Descrição genérica de relacionamento entre partes.

Fonte: adaptado de Beale et al., 2008d.

#### 4.1.4 O IM Comum

O IM Comum, conforme descrito na especificação openEHR utilizada, compreende os pacotes que contêm conceitos abstratos e padrões de design utilizados em classes dos IM superiores, conforme apresentado anteriormente na Figura 8.

Este IM consiste nos pacotes *Archetyped*, *Generic*, *Directory*, *Change Control* e *Resource*.

##### 4.1.4.1 Pacote *Archetyped*

As classes que representam os princípios de design do modelo de dois níveis estão definidas no pacote *Archetyped* e são apresentadas no Quadro 11.

Quadro 11 - Classes implementadas do pacote *rm.common.archetyped*

Classe	Descrição
PATHABLE	Classe abstrata base para todas as classes as quais suas instâncias podem ser acessadas por caminhos e que podem localizar objetos filhos a partir de caminhos.
LOCATABLE	Classe base para todas as classes de modelo de

	informação que podem ser estruturados em arquétipos.
ARCHETYPED	Classe que representa informações de identificação relevantes de arquétipos.
LINK	Define um relacionamento lógico entre dois itens.
FEEDER_AUDIT	Define os dados de auditoria e outros metadados para sistemas na cadeia de alimentação de dados.
FEEDER_AUDIT_DETAILS	Define os detalhes de auditoria para qualquer sistema em uma cadeia de alimentação de sistemas.

Fonte: adaptado de Beale et al., 2008a.

#### 4.1.4.2 Pacote *Generic*

O pacote *Generic* contém as classes que formam padrões genéricos no domínio da saúde, tais como participações e atestações. Um dos usos comuns de algumas das classes definidas neste pacote e apresentadas no Quadro 12, é no referenciamento de entidades demográficas dentro de outros dados.

Quadro 12 - Classes implementadas do pacote `rm.common.generic`

Classe	Descrição
PARTY_PROXY	Conceito abstrato de uma descrição de <i>proxy</i> <sup>16</sup> para uma parte.
PARTY_SELF	Define o <i>proxy</i> de uma parte que representa o indivíduo do registro.
PARTY_IDENTIFIED	Define um <i>proxy</i> de dados para uma parte identificada diferente do indivíduo do registro.
PARTY_RELATED	Define um tipo de <i>proxy</i> para identificar uma parte e seu relacionamento com o indivíduo do registro.
PARTICIPATION	Modelo de uma participação de uma parte, que pode ser qualquer ator ou papel, em uma atividade.
AUDIT_DETAILS	Define o conjunto de atributos necessários para documentar o envio de um item de informação a um repositório.
ATTESTATION	Classe que define o registro de uma atestação por uma parte de itens do conteúdo de registro.
REVISION_HISTORY	Classe que define a noção de um histórico de revisão de itens de auditoria.
REVISION_HISTORY_ITEM	Define uma entrada em um histórico de revisão e

<sup>16</sup> Um objeto *proxy* provê um substituto ou marcador para outro objeto, para controlar o acesso a este (GAMMA et al., 1994).

corresponde a uma versão em um contêiner versionado.

Fonte: adaptado de Beale et al., 2008a.

#### 4.1.4.3 Pacote *Directory*

O pacote *Directory* provê uma abstração simples e reutilizável de estrutura de pastas versionadas. As classes apresentadas no Quadro 13 permitem a representação de pastas versionadas ao longo do tempo, úteis no RES, em serviços demográficos ou em outros locais onde pastas são utilizadas para agrupar informações.

Quadro 13 - Classes implementadas do pacote `rm.common.directory`

Classe	Descrição
VERSIONED_FOLDER	Uma hierarquia de pastas versionadas, simulando uma estrutura de diretório.
FOLDER	Define o conceito de uma pasta nomeada.

Fonte: adaptado de Beale et al., 2008a.

#### 4.1.4.4 Pacote *Change Control*

O pacote *Change Control* define a semântica genérica de alterações em um repositório lógico, como o RES, ao longo do tempo, em resposta a requisitos médico-legais de controle de versão dos registros de saúde. Este controle de versões permite o atendimento aos requisitos de consistência, indelebilidade, rastreabilidade e compartilhamento distribuído. O Quadro 14 descreve as classes deste pacote implementadas nesta pesquisa.

Quadro 14 - Classes implementadas do pacote `rm.common.change_control`

Classe	Descrição
VERSIONED_OBJECT	Define a abstração de controle de versão, definindo a semântica de versionamento de um objeto complexo.
VERSION	Classe que define o modelo abstrato de uma versão dentro de um contêiner de versão.

ORIGINAL_VERSION	Classe que define uma versão contendo conteúdo criado localmente e, opcionalmente, atestações.
IMPORTED_VERSION	Classe que define uma versão na qual o conteúdo original foi copiado de outro local.
CONTRIBUTION	Documenta uma contribuição de uma ou mais versões adicionadas a um repositório com alterações controladas.

Fonte: adaptado de Beale et al., 2008a.

#### 4.1.4.5 Pacote *Resource*

O pacote *Resource* define a estrutura e semântica da noção genérica de um recurso *online*, como documentos, criados por autores humanos que, por consequência, tem como fator o uso de linguagem natural. Esta estrutura, composta pelas classes apresentadas no Quadro 15, provê suporte a traduções em múltiplos idiomas, metadados descritivos e histórico de revisões.

Quadro 15 - Classes implementadas do pacote `rm.common.resource`

Classe	Descrição
AUTHORED_RESOURCE	Classe que define a ideia abstrata de um recurso <i>online</i> criado por um autor humano.
TRANSLATION_DETAILS	Classe que provê detalhes de uma tradução em linguagem natural.
RESOURCE_DESCRIPTION	Define os metadados descritivos de um recurso.
RESOURCE_DESCRIPTION_ITEM	Classe que define os detalhes de descrição de um recurso, em um idioma específico.

Fonte: adaptado de Beale et al., 2008a.

#### 4.1.5 O IM de Estruturas de Dados e o Pacote *Data Structures*

O pacote *Data Structures*, que define o IM de estruturas de dados, é composto por dois subpacotes: *Item Structure* e *History*, que descrevem estruturas de dados endereçáveis por caminhos e a noção genérica de histórico linear para o registro de eventos passados, respectivamente. Neste pacote está definida também, a classe `DATA_STRUCTURE`, descrita no Quadro 16, que é a classe base para todas as estruturas de dados do modelo openEHR.

Quadro 16 - Classe implementada do pacote `rm.data_structures`

Classe	Descrição
DATA_STRUCTURE	Classe abstrata que serve como base para todos os tipos de estruturas de dados do RM.

Fonte: adaptado de Beale et al., 2008b.

#### 4.1.5.1 Pacote *Item Structure*

Neste pacote estão contidas as classes que representam o formalismo necessário para estruturas de dados genéricas e que podem ser definidas em arquétipos, apresentadas no Quadro 17, para a representação de listas, itens simples, tabelas e árvores.

Quadro 17 - Classes implementadas do pacote `rm.data_structures.item_structure`

Classe	Descrição
ITEM_STRUCTURE	Classe abstrata, que serve como base para todos os tipos de dados espaciais.
ITEM_SINGLE	Classe que define a estrutura de dados de um único valor.
ITEM_LIST	Classe que define a estrutura de dados lista.
ITEM_TABLE	Classe que define uma estrutura de dados no estilo de tabela de banco de dados relacional, na qual as colunas são nomeadas e ordenadas.
ITEM_TREE	Classe que define a estrutura de dados árvore.

Fonte: adaptado de Beale et al., 2008b.

#### 4.1.5.2 Pacote *Representation*

Este pacote contém as classes necessárias para representar de forma hierárquica qualquer estrutura de dados. O Quadro 18 descreve as classes deste pacote implementadas nesta pesquisa.

Quadro 18 - Classes implementadas do pacote `rm.data_structures.representation`

Classe	Descrição
ITEM	Classe abstrata que serve como base para as classes de representação CLUSTER e ELEMENT.
CLUSTER	Classe que representa um agrupador, variante de ITEM, que pode

	conter outras instâncias de ITEM, em uma lista ordenada.
ELEMENT	Classe que representa um nó folha na hierarquia, variante de ITEM, na qual uma instância de DATA_VALUE é atribuída.

Fonte: adaptado de Beale et al., 2008b.

#### 4.1.5.3 Pacote *History*

O pacote *History* define as classes que formalizam o conceito de tempo linear passado, por meio do qual, dados históricos, de qualquer complexidade estrutural, podem ser registrados, suportando amostras de eventos instantâneas e definidas em intervalos, bem como eventos em séries periódicas e aperiódicas. As classes que compõem este pacote são apresentadas no Quadro 19.

Quadro 19 - Classes implementadas do pacote `rm.data_structures.history`

Classe	Descrição
HISTORY	Classe que representa o objeto raiz de um histórico linear.
EVENT	Classe que define a noção abstrata de um único evento em uma série.
POINT_EVENT	Classe que define um único evento pontual em uma série.
INTERVAL_EVENT	Classe que define um único evento de intervalo em uma série.

Fonte: adaptado de Beale et al., 2008b.

#### 4.1.6 O IM de Tipos de Dados

Este IM define os tipos de dados clínicos e científicos utilizados nos outros IM da especificação, de forma a harmonizar os tipos de dados utilizados entre os IM, pelos diferentes serviços em uma infraestrutura de informação em saúde. O IM de Estruturas de Dados é composto pelos pacotes *Basic*, *Text*, *Quantity*, *Date Time*, *Encapsulated* e *Uri*, descritos a seguir.

##### 4.1.6.1 Pacote *Basic*

Este pacote contém as classes que definem os conceitos de estado binário ou booleano, estado em uma máquina de estados e identificadores de entidades do

mundo real, apresentadas no Quadro 20. A classe que serve como base para as demais classes deste IM, `DATA_VALUE`, foi implementada como uma *interface* considerando que não possui métodos ou atributos, apenas define uma classe do IM comum às demais.

Quadro 20 - Classes implementadas do pacote `rm.data_types.basic`

Classe	Descrição
<code>DATA_VALUE</code>	Implementada como uma <i>interface</i> , serve como uma base comum para todos os tipos de dados nos modelos openEHR.
<code>DV_BOOLEAN</code>	Classe que define um valor booleano.
<code>DV_STATE</code>	Classe que define a representação de valores de estado, que podem obedecer a um estado de máquinas definido.
<code>DV_IDENTIFIER</code>	Classe para representar identificadores de entidades do mundo real.

Fonte: adaptado de Beale et al., 2008c.

#### 4.1.6.2 Pacote *Text*

As classes que representam valores textuais em registros de saúde, que incluem texto plano, termos codificados e textos narrativos, estão definidas neste pacote e são descritas no Quadro 21.

Quadro 21 - Classes implementadas do pacote `rm.data_types.text`

Classe	Descrição
<code>DV_TEXT</code>	Classe que define um item textual, podendo ser codificado pela adição de mapeamentos.
<code>TERM_MAPPING</code>	Classe que define o mapeamento de um termo codificado para um <code>DV_TEXT</code> .
<code>CODE_PHRASE</code>	Define um termo totalmente coordenado de um serviço de terminologia.
<code>DV_CODED_TEXT</code>	Classe que define um item textual no qual o valor deve ser a rubrica de uma terminologia controlada.
<code>DV_PARAGRAPH</code>	Classe que define um texto composto, consistindo em uma série de instâncias de <code>DV_TEXT</code> , que pode ser interpretado como um parágrafo.

Fonte: adaptado de Beale et al., 2008c.

#### 4.1.6.3 Pacote *Quantity*

Neste pacote estão definidas classes que representam valores quantificáveis como quantidades, valores ordenados, valores ordinais, contadores e proporções. As classes deste pacote, implementadas nesta pesquisa, estão apresentadas no Quadro 22.

Quadro 22 - Classes implementadas do pacote `rm.data_types.quantity`

Classe	Descrição
DV_ORDERED	Classe abstrata que define o conceito de valores ordenados.
DV_INTERVAL	Classe genérica que define um intervalo de um tipo comparável.
REFERENCE_RANGE	Classe que define um intervalo nomeado para ser associado a qualquer dado ordenado.
DV_ORDINAL	Modela classificações e escores, com quantidade finita de valores, com uma ordem estabelecida, porém sem a necessidade de distância constante entre os valores.
DV_QUANTIFIED	Classe abstrata que define o conceito de valores quantificados, que possuem uma magnitude precisa.
DV_AMOUNT	Classe abstrata que define o conceito de quantidades relativas quantificadas.
DV_QUANTITY	Classe que define um tipo quantificado para representar quantidades expressadas como uma magnitude e unidade de medida.
DV_COUNT	Classe que define o conceito de quantidades que podem ser contadas.
DV_PROPORTION	Classe que modela a razão de valores, expressada por um numerador e um denominador.
PROPORTION_KIND	Classe de enumeradores que definem o tipo de proporção para a classe DV_PROPORTION.
DV_ABSOLUTE_QUANTITY	Classe abstrata que define o conceito de entidades quantificáveis nas quais os valores são absolutos com relação a uma origem.

Fonte: adaptado de Beale et al., 2008c.

Neste pacote está definido também o subpacote *Date Time*, que contém as classes que representam dados temporais, apresentadas no Quadro 23.

Quadro 23 - Classes implementadas do pacote `rm.data_types.quantity.date_time`

Classe	Descrição
DV_TEMPORAL	Especialização e variante temporal de DV_ABSOLUTE_QUANTITY.
DV_DATE	Classe que representa um ponto absoluto no tempo, com detalhamento até o dia, apenas.
DV_TIME	Classe que representa um ponto no tempo, a partir de uma origem normalmente interpretada com o início do dia atual, com detalhamento até os segundos, apenas.
DV_DATE_TIME	Classe que representa um ponto absoluto no tempo, com detalhamento até os segundos.
DV_DURATION	Classe que representa um período de tempo, com relação a um ponto fictício e não especificado, no tempo.

Fonte: adaptado de Beale et al., 2008c.

#### 4.1.6.4 Pacote *Encapsulated*

Este pacote contém classes que representam valores de dados que possuem estrutura interna definida fora do RES, como imagens, séries de dados de sinais biológicos, conteúdo textual estruturado, como XML e conteúdos com sintaxe própria, entre outros.

Quadro 24 - Classes implementadas do pacote `rm.data_types.encapsulated`

Classe	Descrição
DV_ENCAPSULATED	Classe abstrata que define os metadados comuns a todos os tipos de dados encapsulados, ou dados com estrutura definida fora do RES.
DV_MULTIMEDIA	Classe especializada de DV_ENCAPSULATED para tipos audiovisuais e sinais biológicos.
DV_PARSEABLE	Classe que define dados encapsulados expressados como uma cadeia de caracteres analisável.

Fonte: adaptado de Beale et al., 2008c.

#### 4.1.6.5 Pacote *Uri*

O pacote *Uri* é composto por duas classes, apresentadas no Quadro 25, que permitem a criação de referência a recursos disponíveis na Internet ou no próprio RES: DV\_URI e sua especialização DV\_EHR\_URI.

Quadro 25 - Classes implementadas do pacote *rm.data\_types.uri*

Classe	Descrição
DV_URI	Classe que define uma referência a um objeto em conformidade com o padrão de identificador universal de recurso (URI – <i>Universal Resource Identifier</i> ).
DV_EHR_URI	Classe especializada de DV_URI, que possui “ehr” como o nome do esquema e que somente permite referenciar elementos em RES.

Fonte: adaptado de Beale et al., 2008c.

#### 4.1.7 O IM de Suporte

Este IM compreende os tipos utilizados por todos os IM openEHR e inclui tipos primitivos considerados existentes e definidos externamente ao modelo openEHR. Estão definidos neste IM, o pseudopacote *Assumed* e os pacotes *Identification*, *Terminology*, *Measurement* e *Definition*. A classe EXTERNAL\_ENVIRONMENT\_ACCESS definida no pacote com o mesmo nome do IM, *Support*, não foi implementada nesta pesquisa por não ter sido necessária para a execução das regras. Desta forma, o pacote *Support* não foi considerado e não será apresentado.

##### 4.1.7.1 Pacote *Assumed*

Este pacote é considerado pela especificação como um pseudopacote, uma vez que considera-se que os tipos definidos neste, estejam disponíveis em uma tecnologia de implementação, como em uma linguagem de programação, linguagem de esquema ou ambiente de banco de dados.

Parte dos tipos contidos neste pacote está disponível na linguagem de programação utilizada no desenvolvimento desta pesquisa, como *Integer*, *Boolean*,

*Double*, *String*, *Array*, *List* e *Set*. Foi necessário implementar apenas o tipo *Interval*, apresentado no Quadro 26.

Quadro 26 - Classe implementada do pacote *rm.support.assumed*

Classe	Descrição
Interval	Classe que define o conceito de intervalo de itens ordenados.

Fonte: adaptado de Beale et al., 2008f.

#### 4.1.7.2 Pacote *Identification*

Neste pacote estão definidas as classes que descrevem um modelo de referências e identificadores para entidades de informação. Instâncias das classes que representam identificadores podem ser utilizadas para identificar entidades do mundo real, como o Cadastro Nacional de Saúde (CNS), entidades que devem possuir identificadores únicos, versões ou alterações a entidades informacionais associadas a entidades do mundo real, entre outras. Já as instâncias das classes que representam referências, podem ser utilizadas para conectar ou associar objetos em um sistema de informação distribuído, permitindo ao RES criar referências entre eles.

As classes deste pacote, implementadas nesta pesquisa são apresentadas no Quadro 27.

Quadro 27 - Classes implementadas do pacote *rm.support.identification*

Classe	Descrição
UID	Classe abstrata que serve como base para as classes que representam identificadores únicos que identificam entidades de informação em uma forma durável.
ISO_OID	Classe que modela identificadores de objeto da ISO.
UUID	Classe que modela identificadores universais únicos do Ambiente de Computação Distribuído (DCE – <i>Distributed Computing Environment</i> ).
INTERNET_ID	Classe que modela o conceito de domínio de internet reverso.
OBJECT_ID	Classe base de identificadores de objetos informacionais, podendo ou não possuir algum significado.
UID_BASED_ID	Classe abstrata que modela o conceito de identificadores baseados em UID que consistem em uma parte raiz e uma extensão opcional.

HIER_OBJECT_ID	Uma classe concreta que corresponde a identificadores hierárquicos na forma definida por UID_BASED_ID.
OBJECT_VERSION_ID	Classe que modela identificadores únicos globais para uma versão de um objeto versionado.
VERSION_TREE_ID	Classe que define o identificador de árvore de versão, para uma versão.
ARCHETYPE_ID	Classe que modela o identificador estrutural multiaxial para arquétipos.
TEMPLATE_ID	Classe que modela o identificador para modelos, ainda sem um formato especificado.
TERMINOLOGY_ID	Classe que modela identificadores para terminologias acessadas por meio de serviços de consulta a terminologias.
GENERIC_ID	Classe para identificadores genéricos que possuem formato desconhecido para o modelo openEHR.
OBJECT_REF	Classe que descreve uma referência para outro objeto que pode existir localmente ou ser mantido externamente ao RES.
ACCESS_GROUP_REF	Classe que descreve a referência a grupo de acesso em um serviço de controle de acesso.
PARTY_REF	Classe que define identificadores para partes em um serviço demográfico ou de identidades.
LOCATABLE_REF	Classe que define uma referência a uma instância de LOCATABLE contida na estrutura de conteúdo em um nível superior dentro de uma instância de VERSION.

Fonte: adaptado de Beale et al., 2008f.

#### 4.1.7.3 Pacote *Terminology*

Este pacote contém as classes para acesso a serviços de terminologia e de conjuntos de códigos, apresentadas no Quadro 28.

Instâncias das classes definidas neste pacote não são necessárias na execução das regras, porém são referenciadas em classes utilizadas no processo de inferência. Desta forma, as classes de acesso e de serviço foram implementadas apenas como *interfaces* e as classes de identificadores foram implementadas como enumeradores, contendo as constantes definidas na especificação.

Quadro 28 - Classes implementadas do pacote `rm.support.terminology`

Classe	Descrição
TERMINOLOGY_SERVICE	Implementada como uma <i>interface</i> , define um objeto que provê um <i>proxy</i> de acesso a um serviço de terminologia.
TERMINOLOGY_ACCESS	Implementada como uma <i>interface</i> , define um objeto que provê um <i>proxy</i> de acesso a uma terminologia.
CODE_SET_ACCESS	Implementada como uma <i>interface</i> , define um objeto que provê um <i>proxy</i> de acesso a um conjunto de códigos.
OPENEHR_TERMINOLOGY_GROUP_IDENTIFIERS	Implementada como um enumerador, define a lista de identificadores para grupos na terminologia openEHR.
OPENEHR_CODE_SET_IDENTIFIERS	Implementada como um enumerador, define a lista de identificadores para conjuntos de código na terminologia openEHR.

Fonte: adaptado de Beale et al., 2008f.

#### 4.1.7.4 Pacote *Measurement*

Este pacote define uma semântica mínima relacionada à medição quantitativa, de unidade e conversões, permitindo expressar corretamente o pacote *Quantity* do IM de Tipos de Dados. Contudo, assim como o pacote *Terminology*, este pacote não possui classes utilizadas na execução das regras, porém necessitou ser implementado, pois a classe definida neste, apresentada no Quadro 29, é referenciada em outras classes necessárias.

Quadro 29 - Classe implementada do pacote `rm.support.measurement`

Classe	Descrição
MEASUREMENT_SERVICE	Implementada como uma <i>interface</i> , define um objeto que provê um <i>proxy</i> de acesso a um serviço de informações de medições.

Fonte: adaptado de Beale et al., 2008f.

#### 4.1.7.5 Pacote *Definition*

Este pacote compreende apenas duas classes, apresentadas no Quadro 30, que possuem definições simbólicas usadas pelos IM openEHR. Atualmente, a especificação define somente duas constantes na classe BASIC\_DEFINITIONS: o caractere de retorno de carro (CR – *Carriage Return*) e o caractere de avanço de linha (LF – *Line Feed*).

Quadro 30 - Classes implementadas do pacote rm.support.definition

Classe	Descrição
OPENEHR_DEFINITIONS	Implementada como uma <i>interface</i> , define a base para prover acesso a constantes definidas em outros pacotes.
BASIC_DEFINITIONS	Implementada como uma <i>interface</i> , define valores de constantes utilizadas globalmente.

Fonte: adaptado de Beale et al., 2008f.

#### 4.1.80 IM de Integração e o Pacote *Integration*

O IM de Integração é composto pelo pacote *Integration*, que por sua vez compreende a classe GENERIC\_ENTRY, utilizada para representar dados provenientes de fontes externas e que não estão em conformidade com o modelo openEHR, em estrutura definida em arquétipos, permitindo uma separação entre as transformações sintática e semântica nos dados.

A transformação sintática consiste na conversão dos dados da fonte externa, de seu formato original ou qualquer outro formato intermediário, para um formato que obedeça a uma classe especial no RM openEHR, porém com estrutura lógica e semântica controladas por arquétipos de integração, levando os dados de origem ao contexto computacional do modelo openEHR. A transformação semântica compreende a transformação deste dado em formato intermediário em conformidade com o modelo openEHR, em instâncias do RM openEHR principal, obedecendo os arquétipos clínicos modelados.

A classe implementada que permite esta integração de dados de fontes externas é apresentada no Quadro 31.

Quadro 31 - Classe implementada do pacote rm.integration

Classe	Descrição
GENERIC_ENTRY	Classe utilizada para criar representações intermediárias de dados provenientes de fontes que não estão em conformidade com as classes do RM openEHR, como mensagens HL7, bancos de dados relacionais entre outras.

Fonte: adaptado de Beale et al., 2006.

#### 4.1.9O IM de Segurança e o Pacote *Security*

Este IM e a classe definida no pacote *Security* não possuem documento ou especificação disponível, porém a classe é referenciada na classe EHR\_ACCESS do pacote EHR, que por sua vez é referenciada na classe EHR do mesmo pacote, do IM de RES. Desta forma, a classe apresentada no Quadro 32 foi implementada como uma classe abstrata, sem métodos ou atributos, apenas para manter a implementação com o mínimo de alterações relativas à especificação possível.

Quadro 32 - Classe implementada do pacote rm.security

Classe	Descrição
ACCESS_CONTROL_SETTINGS	Classe abstrata, que representa configurações de controle de acesso. Não possui especificação disponível.

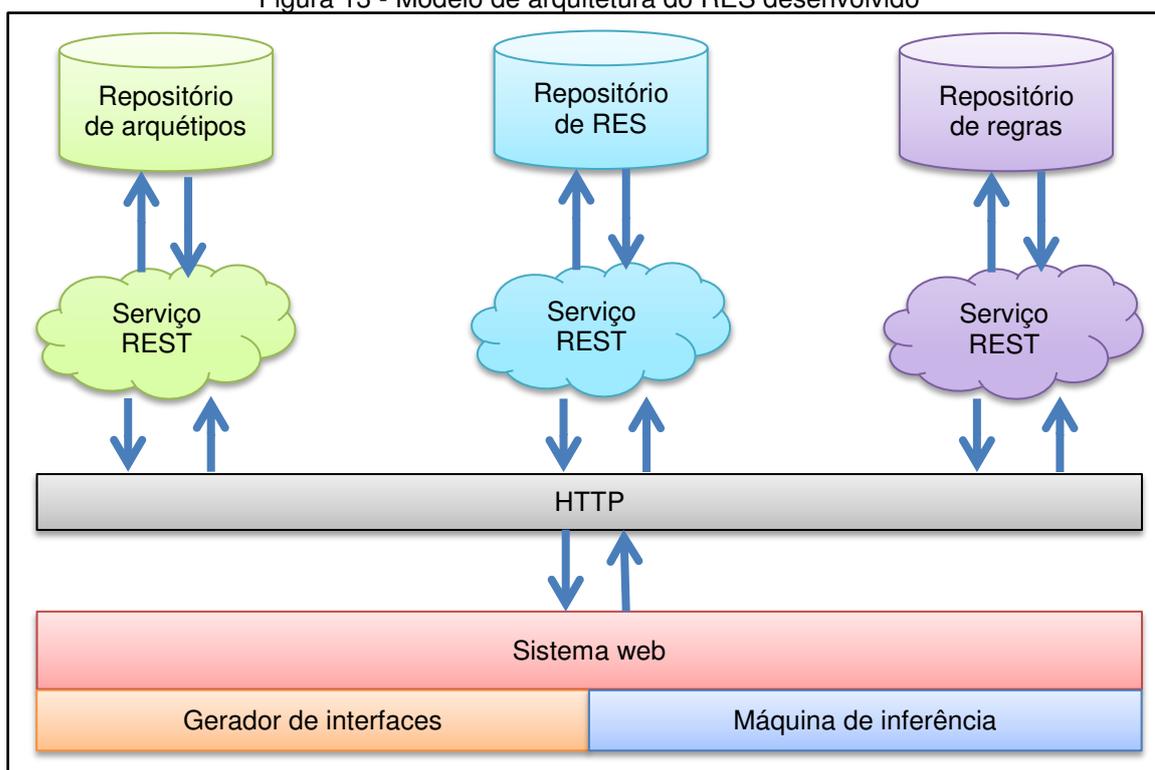
Fonte: o autor, 2014.

## 4.2 MODELO PARA O DESENVOLVIMENTO DE UM RES BASEADO EM ARQUÉTIPOS

Além das implementações realizadas, parte prática do projeto, esta pesquisa obteve como resultado a concepção de um modelo de arquitetura para o desenvolvimento de um RES baseado em arquétipos, capaz de prover suporte a decisão. Este modelo, apresentado na Figura 13, utiliza o conceito de repositórios, onde são armazenados os recursos necessários e serviços, que manipulam os dados dos repositórios.

Neste modelo, o sistema comunica-se com os repositórios unicamente por meio do respectivo serviço, sem qualquer conexão direta com o repositório. Com esta abordagem, o serviço responsável pela manipulação dos dados do repositório pode utilizar uma ou mais fontes de dados e até mesmo substituir repositórios indisponíveis, sendo transparente ao sistema qual a origem e forma de obtenção destes recursos.

Figura 13 - Modelo de arquitetura do RES desenvolvido



Fonte: o autor, 2014.

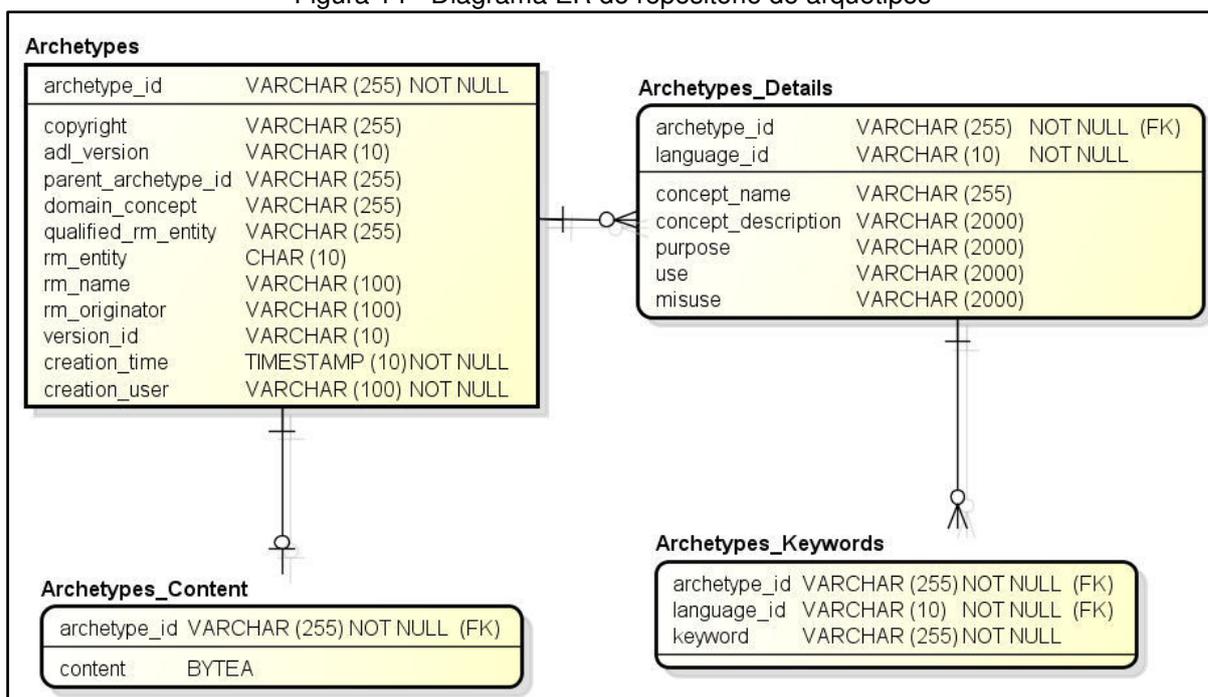
A abordagem de serviços RESTful para o desenvolvimento dos serviços, permite aproveitar os recursos providos pelo protocolo HTTP, reduzindo o acoplamento entre o sistema e estes serviços, considerando que toda a comunicação com estes é feita de forma padronizada por meio de requisições HTTP. Desta forma, outras implementações dos serviços podem ser disponibilizadas, sendo necessário apenas que o sistema conheça o URL para cada operação existente, o método de requisição e os parâmetros necessários. Neste caso, não é necessário inclusive que o sistema implemente interfaces para cada serviço ou utilize a mesma tecnologia de desenvolvimento empregada neste.

Nesta pesquisa, considerando que são 3 os tipos de recursos manipulados, foram desenvolvidos 3 repositórios, implementados por meio do SGBD PostgreSQL: ArchRepo, EHRRepo e RulesRepo. Também foram implementados 3 serviços RESTful, para acessar cada um destes repositórios: ArchMan, EHRMan e RulesMan. O Apêndice B apresenta uma listagem dos códigos HTTP retornados pelos serviços, com uma breve descrição de cada um destes.

#### 4.2.1 Repositório ArchRepo

De forma a armazenar os arquétipos que serão disponibilizados para o sistema por meio do serviço ArchMan, foi implementado o repositório utilizando o PostgreSQL, pois este permite a busca de dados por meio de expressões regulares, que são utilizadas na restrição de arquétipos possíveis em *slots*. A Figura 14 apresenta o diagrama Entidade-Relacionamento (ER) do repositório de arquétipos.

Figura 14 - Diagrama ER do repositório de arquétipos



Fonte: o autor, 2014.

Como alternativa, poderiam ser utilizados outros SGBD como Oracle Database<sup>17</sup> e MySQL<sup>18</sup>, que também possuem a funcionalidade de busca por expressões regulares, bem como o armazenamento dos arquétipos em um sistema de arquivos. Porém, neste último caso, não seria possível realizar a indexação de metadados, como palavras-chave, conceito, entre outros, sem a utilização de recursos adicionais.

<sup>17</sup> <https://www.oracle.com/database/index.html>

<sup>18</sup> <http://www.mysql.com/>

#### 4.2.2 Repositório EHRRepo

Os dados coletados de forma estruturada em arquétipos são armazenados em um repositório próprio, implementado por meio da utilização do SGBD PostgreSQL.

Este repositório armazena, além dos dados clínicos, os dados coletados por meio de arquétipos demográficos. Apesar de ser recomendada a separação das informações demográficas de forma a dificultar a associação de informações clínicas aos respectivos indivíduos em caso de acesso não autorizado, esta pesquisa não utilizou tal abordagem para reduzir a complexidade da implementação, uma vez que a execução das regras não sofre interferência desta separação de dados, apenas necessita da disponibilidade destes. De forma a prover esta separação de dados, seria necessário um SGBD separado para conter apenas os dados demográficos e identificadores, sendo que o serviço de manipulação de dados de RES seria também o responsável por solicitar os dados demográficos a partir do identificador e realizar a associação destes com os dados clínicos.

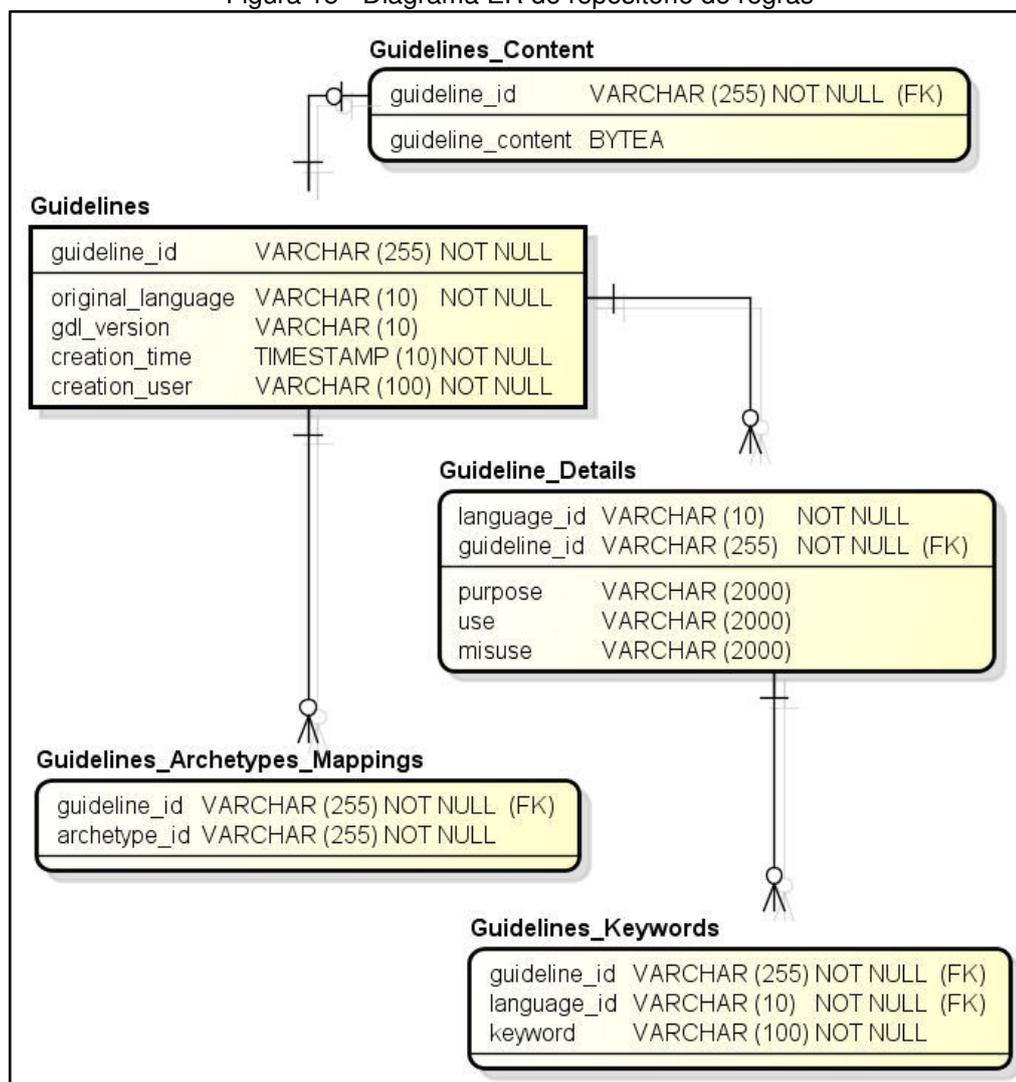
Considerando-se que a estrutura de armazenamento dos dados estruturados em arquétipos constitui um dos resultados desta pesquisa, esta é apresentada no item 4.4.

#### 4.2.3 Repositório RulesRepo

De forma a armazenar as regras que serão disponibilizadas para o motor de inferência integrado ao sistema por meio do serviço RulesMan, foi implementado o repositório utilizando o PostgreSQL. A Figura 15 apresenta o diagrama ER do repositório de regras.

O conteúdo das regras em GDL é armazenado em uma coluna do tipo `bytea`, um tipo de dados do SGBD PostgreSQL para o armazenamento de dados binários com tamanho variável, semelhante ao tipo objetos binários grandes (BLOB – *Binary Large Object*) utilizado em outros SGBD. Metadados como idioma, versão e palavras-chave são armazenados em campos de texto simples, porém podem ser recuperados a partir do conteúdo da diretriz.

Figura 15 - Diagrama ER do repositório de regras



Fonte: o autor, 2014.

Assim como os repositórios descritos anteriormente, o RulesRepo poderia ser implementado utilizando outros SGBD, considerando que este componente não utilizou nenhum recurso específico do SGBD utilizado. Como o PostgreSQL foi utilizado nos demais componentes, optou-se por utilizar o mesmo SGBD neste.

As operações de inserção e recuperação das regras no RulesRepo são realizadas exclusivamente por meio do serviço RulesMan.

#### 4.2.4 Serviço ArchMan

O acesso ao repositório de arquétipos é realizado exclusivamente por meio do serviço ArchMan, um serviço RESTful que disponibiliza recursos de criação e recuperação de dados por meio de requisições HTTP. Estes recursos são

acessados por meio de um caminho, especificando-se o método de requisição e enviando ao serviço, parâmetros no URL ou no corpo da requisição.

O Quadro 33 apresenta as operações disponibilizadas pelo ArchMan, com o caminho pelo qual são acessadas, o método de requisição e uma breve descrição da operação. Neste serviço, o contexto para todas as operações é /archetypes. Todos os caminhos apresentados são relativos ao URL do servidor onde o serviço está sendo executado e ao contexto padrão. Por exemplo, para o caminho /operacao do serviço disponibilizado em <http://www.servidor.com.br>, o URL completo da operação seria <http://www.servidor.com.br/archetypes/operacao>.

Quadro 33 - Operações disponibilizadas pelo serviço ArchMan

Operação	Caminho	Método	Descrição
<b>Adicionar</b>	/create	POST	Adiciona o conteúdo de um arquétipo enviado no corpo da requisição no repositório.
<b>Substituir</b>	/replace	POST	Substitui o conteúdo de um arquétipo previamente adicionado, pelo conteúdo enviado no corpo da requisição.
<b>Buscar arquétipo</b>	/{id}	GET	Envia o conteúdo do arquétipo com identificador igual a {id}, caso seja encontrado.
<b>Buscar lista do nome do RM dos arquétipos</b>	/rmnamelist	GET	Retorna uma lista contendo os nomes dos RM dos arquétipos do repositório, exemplo, "rim", "ehr_rm", "en13606".
<b>Buscar lista de entidades do RM a partir do nome do RM</b>	/entitylist/{rmName}	GET	Retorna uma lista contendo o nome das entidades do RM para todos os arquétipos do repositório com nome do RM igual a {rmName}, exemplo "folder", "composition", "entry".
<b>Buscar lista de arquétipos de uma entidade e RM específicos</b>	/archetypelist/{rmName}/{rmEntity}	GET	Retorna uma lista contendo o identificador dos arquétipos com a entidade {rmEntity} e RM {rmName}.

Fonte: o autor, 2014.

A partir dos URL disponibilizados pelo ArchMan, o sistema web é capaz de acessar os arquétipos e metadados armazenados no ArchRepo para utilização no componente de geração de interfaces e motor de inferência, por meio de requisições HTTP, de acordo com os parâmetros e métodos apresentados no Quadro 33.

A operação Adicionar permite que seja incluído no ArchRepo um novo arquétipo, utilizando o identificador deste como forma de identificar univocamente

estes recursos, retornando o código HTTP 200, para indicar que a operação foi realizada com sucesso. Caso o identificador já esteja associado a um arquétipo do repositório, este método retorna o código HTTP 409, informando o sistema que a requisição resultou em um conflito com um recurso já existente. Na ocorrência de algum erro, decorrente do processamento do arquétipo, é retornado o código HTTP 406 que indica que o recurso enviado na requisição não é aceitável para esta operação, incluindo na resposta, o erro gerado.

Para substituir um arquétipo já existente no ArchRepo, o serviço ArchMan disponibiliza a operação Substituir que, como o próprio nome do método indica, substitui o conteúdo de um recurso do repositório, retornando o código HTTP 200, indicando o sucesso da operação. Caso seja enviado um arquétipo ainda não disponibilizado no repositório, esta operação retorna o código HTTP 404, indicando que o recurso não foi encontrado. No caso de erros durante o processamento do arquétipo, é retornado o código HTTP 406, contendo também a mensagem de erro retornada, informando ao sistema que o conteúdo enviado não é aceitável para a operação.

Os arquétipos contidos no repositório podem ser recuperados pelo sistema, por meio da operação Buscar arquétipo do ArchMan, que recebe o identificador do arquétipo desejado e retorna o código HTTP 200, indicando o sucesso da operação, juntamente com o conteúdo do arquétipo no corpo da resposta, em formato JSON. Se o identificador informado não estiver associado a algum arquétipo do repositório, é retornado o código HTTP 404, informando que o recurso não foi encontrado. Este método permite que seja informada como identificador, uma expressão regular onde o primeiro arquétipo que atenda à expressão informada é retornado.

O serviço ArchMan permite também que sejam recuperados metadados dos arquétipos contidos no repositório ArchRepo. Para isso são disponibilizadas as seguintes operações:

- Buscar lista do nome do RM dos arquétipos: esta operação retorna o código HTTP 200, indicando o sucesso da operação, juntamente com um JSON contendo a lista dos nomes de RM de todos os arquétipos do ArchRepo. Caso o repositório não possua arquétipos armazenados, é retornado o código HTTP 404, informando que o recurso não foi encontrado;

- Buscar lista de entidades do RM a partir do nome do RM: esta operação recebe como parâmetro um nome de RM e retorna o código HTTP 200, indicando o sucesso da operação, juntamente com um JSON contendo a lista de todas as entidades pertencentes ao RM informado, dos arquétipos disponibilizados no ArchRepo, por exemplo, *folder*, *composition* e *entry*. Caso não sejam encontrados arquétipos para o RM informado, é retornado o código HTTP 404; e
- Buscar lista de arquétipos de uma entidade e RM específicos: esta operação recebe como parâmetro o nome do RM e a entidade a ser buscada e retorna o código HTTP 200, indicando o sucesso da operação juntamente com o JSON contendo uma lista com o identificador de todos os arquétipos disponibilizados no ArchRepo pertencentes ao RM informado e que possuam a entidade especificada no parâmetro. Caso não sejam encontrados arquétipos com os critérios definidos, é retornado o código HTTP 404.

#### **4.2.5 Serviço EHRMan**

O acesso ao repositório de dados de RES é realizado exclusivamente por meio do serviço EHRMan, um serviço RESTful, dividido logicamente em dois serviços: serviço de Pacientes e serviço de RES. Ambos os serviços permitem a criação e recuperação de dados do repositório por meio de requisições HTTP. Estes recursos são acessados por meio de um caminho, especificando-se o método de requisição e enviando ao serviço, parâmetros no URL ou no corpo da requisição.

O Quadro 34 apresenta as operações disponibilizadas pelo EHRMan para o serviço de Pacientes, com o caminho pelo qual são acessadas, o método de requisição e uma breve descrição da operação. Neste serviço, o contexto para todas as operações é /patients. Todos os caminhos apresentados são relativos ao URL do servidor onde o serviço está sendo executado e ao contexto padrão. Por exemplo, para o caminho /operacao do serviço disponibilizado em <http://www.servidor.com.br>, o URL completo da operação seria <http://www.servidor.com.br/patients/operacao>.

Quadro 34 - Operações disponibilizadas pelo EHRMan para o serviço de Pacientes

Operação	Caminho	Método	Descrição
<b>Adicionar</b>	/create	POST	Armazena no EHRRepo os dados do paciente enviados no corpo da requisição.
<b>Atualizar paciente</b>	/update	POST	Atualiza os dados de um paciente já armazenado no repositório, a partir do conteúdo enviado no corpo da requisição.
<b>Buscar paciente</b>	/{id}	GET	Busca no EHRRepo um paciente com o identificador informado, retornando o registro estruturado em arquétipo.
<b>Listar pacientes</b>	/list/{name}	GET	Retorna uma lista de identificadores dos pacientes que possuem o nome que corresponda ao parâmetro informado, podendo ser parcialmente ou completo.

Fonte: o autor, 2014.

A operação Adicionar permite que seja incluído no EHRRepo um novo registro de paciente. O serviço de Pacientes atribui como identificação, um Identificador Universal Único (UUID – *Universally Unique Identifier*) gerado de forma aleatória e processa os mapeamentos existentes para a indexação de dados específicos, como nome, data de nascimento entre outros, descrito no item 4.4. Em seguida, os dados são armazenados no EHRRepo e é retornado o código HTTP 201, indicando que o recurso enviado foi criado com sucesso, juntamente com os dados atualizados em formato JSON, incluindo o identificador gerado, no corpo da resposta.

Para atualizar os dados de um paciente já armazenado no EHRRepo, é disponibilizada a operação Atualizar paciente, que recebe os dados do paciente estruturados em arquétipo, que deve conter o identificador já atribuído. O serviço envia para o EHRRepo uma nova versão do registro, retornando o código HTTP 200, indicando o sucesso da operação. Caso o identificador do paciente não esteja atribuído é retornado o código HTTP 406, indicando que o conteúdo enviado não é aceitável para a operação.

A operação Buscar paciente permite ao sistema, por meio do serviço, recuperar os dados de um paciente previamente armazenado no EHRRepo. Utilizando como critério de busca o identificador enviado por parâmetro, o EHRMan retorna o código HTTP 200, indicando que a operação foi realizada com sucesso, juntamente com os dados do paciente no corpo da resposta, estruturado em arquétipo, no formato JSON. Nos casos em que o EHRRepo não possua nenhum paciente com o identificador informado, o código HTTP 404 é enviado, de forma a informar ao sistema que o recurso não foi encontrado.

O EHRMan permite também a listagem de pacientes, por meio da operação Listar pacientes que recebe como parâmetro, parte do nome para a busca. O identificador de cada paciente que possui o texto enviado no parâmetro no nome é inserido em um JSON e enviado no corpo da resposta, que inclui o código HTTP 200 para indicar o sucesso da operação. Se nenhum registro de paciente corresponder ao critério de busca, o código HTTP 404 é retornando informando ao sistema que a busca não retornou resultados.

Além dos dados de pacientes, o EHRMan permite também o acesso aos dados de RES armazenados no EHRRepo, bem como o envio de dados para o repositório. O Quadro 35 apresenta as operações disponibilizadas pelo EHRMan para o serviço de dados do RES, com o caminho pelo qual são acessadas, o método de requisição e uma breve descrição da operação. Neste serviço, o contexto para todas as operações é /ehr. Todos os caminhos apresentados são relativos ao URL do servidor onde o serviço está sendo executado e ao contexto padrão. Por exemplo, para o caminho /operacao do serviço disponibilizado em <http://www.servidor.com.br>, o URL completo da operação seria <http://www.servidor.com.br/ehr/operacao>.

Quadro 35 - Operações disponibilizadas pelo EHRMan para o serviço de dados do RES

Operação	Caminho	Método	Descrição
<b>Adicionar</b>	/create	POST	Armazena no EHRRepo os dados do RES enviados no corpo da requisição.
<b>Atualizar registro</b>	/update	POST	Armazena uma nova versão dos dados de RES já armazenados no repositório, a partir do conteúdo enviado no corpo da requisição.
<b>Buscar registro</b>	{id}	GET	Busca no EHRRepo um registro de RES com o identificador informado, retornando o registro estruturado em arquétipo.
<b>Listar registros de um paciente</b>	/list/patient/{id}	GET	Retorna uma lista de identificadores dos RES associados ao paciente com o identificador informado.
<b>Listar registros de um paciente para um arquétipo específico</b>	/list/patient/{id}/{archetypeId}	GET	Retorna uma lista de identificadores dos RES associados ao paciente com o identificador informado, coletados a partir do arquétipo com o identificador informado.
<b>Listar registros de um arquétipo específico</b>	/list/archetype/{id}	GET	Retorna uma lista de identificadores dos RES coletados a partir do arquétipo com o identificador informado.

Fonte: o autor, 2014.

A operação Adicionar permite que seja incluído no EHRRepo um novo registro de RES, associado a um paciente. O serviço de dados de RES atribui como identificação ao registro, um UUID gerado de forma aleatória e processa os mapeamentos existentes para a indexação de dados específicos, caso existam. Em seguida, os dados são armazenados no EHRRepo e é retornado o código HTTP 201, indicando que o recurso enviado foi criado com sucesso, juntamente com os dados atualizados, incluindo o identificador gerado, no corpo da resposta, em formato JSON. A estrutura do repositório e a forma como os dados são armazenados neste é descrita em detalhes no item 4.4. Caso a requisição não contenha os dados informados no corpo desta, o código HTTP 400 é retornado, indicando que a requisição não pôde ser compreendida e que não deve ser repetida sem que a requisição seja modificada.

Para atualizar dados de RES já armazenado no EHRRepo, é disponibilizada a operação Atualizar registro, que recebe os dados do RES estruturados em arquétipo, que deve conter o identificador já atribuído. O serviço envia para o EHRRepo uma nova versão do registro, retornando o código HTTP 200, indicando o sucesso da operação. Caso o identificador do RES não esteja atribuído é retornado o código HTTP 406, indicando que o conteúdo enviado não é aceitável para a operação.

Os registros de RES específicos podem ser recuperados por meio da operação Buscar registro, que recebe por parâmetro o identificador deste. O EHRMan retorna o código HTTP 200, indicando o sucesso da busca, juntamente com os dados do registro em formato JSON, no corpo da resposta. Caso nenhum registro corresponda ao identificado informado, é retornado o código HTTP 404 para indicar que o recurso não foi encontrado no repositório.

O EHRMan possui também, operações para a busca do identificador de registros de RES de acordo com critérios específicos. As operações para esta finalidade permitem que seja especificado o número máximo de registros retornados, enviando-se no cabeçalho da requisição, o atributo max-results com a quantidade máxima de resultados que podem ser retornadas. As operações de busca são:

- Listar registros de um paciente: recebe o identificador do paciente e retorna um JSON contendo o identificador dos registros de RES armazenados no repositório, por ordem cronológica do mais atual para o mais antigo, retornando também, o código HTTP 200 para indicar o

sucesso da operação ou o código HTTP 404 para indicar que nenhum registro de RES foi encontrado para o paciente;

- Listar registros de um paciente para um arquétipo específico: recebe o identificador do paciente e do arquétipo e retorna um JSON contendo o identificador dos registros de RES armazenados no repositório, associados ao paciente e coletados para o arquétipo especificado, por ordem cronológica do mais atual para o mais antigo, retornando também, o código HTTP 200 para indicar o sucesso da operação ou o código HTTP 404 para indicar que nenhum registro de RES foi encontrado para o paciente e arquétipo informados; e
- Listar registros de um arquétipo específico: recebe o identificador do arquétipo e retorna um JSON contendo o identificador dos registros de RES armazenados no repositório para o arquétipo informado, por ordem cronológica do mais atual para o mais antigo, retornando também, o código HTTP 200 para indicar o sucesso da operação ou o código HTTP 404 para indicar que nenhum registro de RES foi encontrado para o arquétipo.

#### **4.2.6 Serviço RulesMan**

Seguindo a abordagem dos serviços já descritos, o acesso ao repositório de regras é realizado exclusivamente por meio do serviço RulesMan, um serviço RESTful que permite a criação e recuperação de diretrizes e metadados do repositório por meio de requisições HTTP. Estes recursos são acessados por meio de um caminho, especificando-se o método de requisição e enviando ao serviço, parâmetros na URL ou no corpo da requisição.

O Quadro 36 apresenta as operações disponibilizadas pelo RulesMan com o caminho pelo qual são acessadas, o método de requisição e uma breve descrição da operação. Neste serviço, o contexto para todas as operações é /rules. Todos os caminhos apresentados são relativos ao URL do servidor onde o serviço está sendo executado e ao contexto padrão. Por exemplo, para o caminho /operacao do serviço disponibilizado em <http://www.servidor.com.br>, o URL completo da operação seria <http://www.servidor.com.br/rules/operacao>.

Quadro 36 - Operações disponibilizadas pelo serviço RulesMan

Operação	Caminho	Método	Descrição
<b>Adicionar</b>	/create	POST	Armazena no RulesRepo o conteúdo da diretriz enviado no corpo da requisição.
<b>Atualizar</b>	/update	POST	Atualiza no RulesRepo, a diretriz com o conteúdo enviado no corpo da requisição.
<b>Buscar diretriz</b>	/guideline/{id}	GET	Retorna o conteúdo da diretriz com o identificador informado, armazenado no RulesRepo.
<b>Listar diretrizes com arquétipo mapeado</b>	/guidelines/by-archetype/{id}	GET	Retorna uma lista com o identificador das diretrizes que possuem mapeado, o arquétipo especificado.
<b>Listar arquétipos mapeados para diretriz</b>	/archetypes/by-guideline/{id}	GET	Retorna uma lista contendo o identificador de todos os arquétipos mapeados para a diretriz com o identificador informado.

Fonte: o autor, 2014.

A operação Adicionar permite que seja incluído no RulesRepo o conteúdo de uma diretriz, que deve possuir um identificador que ainda não esteja associado a nenhuma diretriz no repositório. Além do conteúdo, são armazenados metadados, já descritos no item 4.2.3. Em seguida, o código HTTP 201 é retornado, indicando que o recurso enviado foi criado com sucesso, juntamente com os dados sem modificações no corpo da resposta. Caso o corpo da requisição não inclua o conteúdo da diretriz, o código HTTP 400 é enviado, informado ao sistema que a requisição não deve ser repetida sem a correção das informações enviadas. No caso de o identificador da diretriz já estar associado a outro registro do repositório, o código HTTP 409 é retornado, informando um conflito com outro recurso do repositório. Se o processo resultar em algum erro, o código HTTP 406 é retornado juntamente com a mensagem do erro gerado.

Para atualizar o conteúdo e os metadados de uma diretriz já existente no RulesRepo, o serviço RulesMan disponibiliza a operação Atualizar que permite a atualização do conteúdo de um recurso do repositório, bem como dos metadados associados a este, retornando o código HTTP 200, indicando o sucesso da operação. Caso seja enviada uma diretriz ainda não disponibilizada no repositório, esta operação retorna o código HTTP 404, indicando que o recurso não foi encontrado. No caso de erros durante o processamento da diretriz, é retornado o

código HTTP 406, contendo também a mensagem de erro retornada, informando ao sistema que o conteúdo enviado não é aceitável para a operação.

As diretrizes disponibilizadas no repositório podem ser recuperadas pelo sistema, por meio da operação Buscar diretriz do RulesMan, que recebe o identificador do recurso desejado e retorna o código HTTP 200, indicando o sucesso da operação, juntamente com o conteúdo da diretriz no corpo da resposta. Se o identificador informado não estiver associado a alguma diretriz do repositório, é retornado o código HTTP 404, informando que o recurso não foi encontrado.

Para facilitar a busca das diretrizes que podem ser executadas em dados de um determinado arquétipo, o RulesMan disponibiliza a operação Listar diretrizes com arquétipo mapeado que recebe como parâmetro, o identificador do arquétipo e retorna uma lista com o identificador de todas as diretrizes do RulesRepo, que possuem o arquétipo mapeado, em formato JSON, juntamente com o código HTTP 200. Caso nenhuma diretriz possua o arquétipo mapeado, é retornado o código HTTP 404, indicando que nenhum recurso foi encontrado.

O RulesMan disponibiliza também, a operação Listar arquétipos mapeados para diretriz que recebe como parâmetro o identificador de uma diretriz e retorna uma lista, em formato JSON, contendo o identificador de todos os arquétipos mapeados para a diretriz especificada, juntamente com o código HTTP 200. O código HTTP 404 é retornado caso a diretriz não esteja armazenada no RulesRepo ou nenhum arquétipo esteja associado à diretriz.

#### 4.3 GERAÇÃO AUTOMÁTICA DE INTERFACE PARA A COLETA DE DADOS ESTRUTURADOS EM ARQUÉTIPOS

Para a coleta dos dados estruturados em arquétipos, foi necessário desenvolver um componente do sistema capaz de gerar uma interface para a entrada de dados, com componentes adequados aos tipos de dados das definições dos arquétipos. Utilizando o RM implementado nesta pesquisa, são geradas telas utilizando a tecnologia JSF 2, utilizando o framework de componentes PrimeFaces versão 5.0<sup>19</sup>.

---

<sup>19</sup> <http://www.primefaces.org/>

O sistema disponibiliza uma listagem dos arquétipos disponíveis no repositório para que seja gerada a interface para a coleta dos dados deste, conforme apresentado na Figura 16.

Figura 16 - Listagem dos arquétipos disponíveis

The screenshot shows the Smart archEHR Web interface. On the left, a navigation menu lists various archetype categories: DEMOGRAPHIC, EN13606, and EHR. Under EHR, the 'blood\_pressure (v1)' archetype is highlighted. On the right, the details for 'Pressão Arterial' are shown, including a description and a 'Novo registro' button. Below this is a table of recent records with columns for 'Paciente' and 'Data/hora do registro'.

Paciente	Data/hora do registro	
Diego Garcia	20/10/2014 19:25	Ver
Diego Garcia	20/10/2014 19:20	Ver
Diego Garcia	20/10/2014 14:47	Ver
Claudia Maria Cabral Moro	19/10/2014 15:07	Ver
Alex de Oliveira	20/10/2014 14:53	Ver
Diego Garcia	19/10/2014 14:13	Ver
Lilian Mie Mukai Cintho	19/10/2014 13:33	Ver
Diego Garcia	19/10/2014 12:37	Ver
Lilian Mie Mukai Cintho	01/10/2014 17:37	Ver
Diego Garcia	01/09/2014 18:34	Ver

Fonte: o autor, 2014.

Ao selecionar um arquétipo da listagem, são apresentadas informações deste, no idioma selecionado pelo sistema, bem como os últimos registros armazenados no EHRRepo para este arquétipo e a opção para a geração de interface para a coleta de dados no painel direito da tela.

A Figura 17 e a Figura 18 apresentam a tela gerada a partir do arquétipo openEHR-DEMOGRAPHIC-PERSON.person-patient.v1. As restrições definidas como *slots* também são geradas na interface, por meio da recuperação dos arquétipos especificados. Nesta tela, o arquétipo principal, openEHR-

DEMOGRAPHIC-PERSON.person-patient.v1, possui referência aos seguintes arquétipos:

- openEHR-DEMOGRAPHIC-ITEM\_TREE.person\_details.v1;
- openEHR-DEMOGRAPHIC-CLUSTER.birth\_data\_additional\_detail\_br.v1;
- openEHR-DEMOGRAPHIC-CLUSTER.person\_identifier.v1;
- openEHR-DEMOGRAPHIC-PARTY\_IDENTITY.person\_name.v1; e
- openEHR-DEMOGRAPHIC-ADDRESS.address.v1.

Figura 17 - Interface gerada para o arquétipo openEHR-DEMOGRAPHIC-PERSON.person-patient.v1

The screenshot displays a web browser window titled "Smart archEHR Web" with the URL "smartarchehr.garciad.com/smartehr/archtypedata.ehr?archetypeId=openEHR-EHR-OBSERVATION.L". The main content area is titled "Dados do paciente" and contains several sections for data entry:

- Dados da pessoa:** Includes fields for "Gênero" (set to Masculino), "Sobrenome da Mãe" (Ferrari), "Estado Civil" (Solteiro), "Etnia" (Etnia), and "Comentários de identificação" (Comentários de identificação).
- Dados demográficos do nascimento:** Includes fields for "Data do nascimento" (23/01/1988), "Indicador de seguimento da data de nascimento" (checkbox), "país de nascimento" (Brasil), "Estado/território/provincia" (Santa Catarina), "Cidade/localidade" (Criciúma), "Nascimento múltiplo" (Nascimento múltiplo), "Ordem de Nascimento" (Ordem de Nascimento), "sexo" (Masculino), and "Cartório" (Cartório).
- Outros dados da certidão de nascimento brasileira:** A collapsed section.
- Dados demográficos do óbito:** A collapsed section.
- Identificador biométrico de uma pessoa:** A collapsed section.
- Identificador para uma pessoa:** A collapsed section.
- Componentes do nome:** Includes fields for "Nome atribuído" (Diego), "Sobrenome" (Garcia), "Título do nome" (Título do nome), and "Sufixo do nome" (Sufixo do nome). A "Nome preferencial" checkbox is checked.

A small icon of a laptop with a document is visible in the bottom right corner of the form area.

Fonte: o autor, 2014.

O texto utilizado como rótulo dos campos e dica dentro dos próprios campos é recuperado a partir dos metadados do arquétipo, de acordo com o idioma selecionado. Na Figura 17 foi selecionado o idioma Português do Brasil, desta forma todas as definições que possuem tradução para este idioma são apresentadas em Português. Caso alguma definição não possua tradução no idioma selecionado, é apresentado o texto no idioma original.

Figura 18 - Interface gerada para o arquétipo openEHR-DEMOGRAPHIC-PERSON.person-patient.v1

The screenshot shows a web browser window with the URL `smartarchehr.garciad.com/smartehr/archtypedata.ehr?archetypeld=openEHR-EHR-OBSERVATION.E`. The page title is "Smart archEHR Web".

The form contains the following sections and fields:

- Outros dados da certidão de nascimento brasileira** (collapsible section)
- Dados demográficos do óbito** (collapsible section)
- Identificador biométrico de uma pessoa** (collapsible section)
- Identificador para uma pessoa.** (collapsible section)
- Componentes do nome** (expanded section):
  - Nome atribuído:
  - Sobrenome:
  - Título do nome:
  - Sufixo do nome:
  - Nome preferencial:
  - Indicador de acurácia/segurança:
  - Nome completo:
- Intervalo de utilização** (collapsible section):
  - Data inicial:
  - Data inicial inclusiva:
  - Sem limite para data inicial:
  - Data final:
  - Data final inclusiva:
  - Sem limite para data final:
- Identificador usado** (collapsible section):
  - Identificador:
  - Tipo:
  - Atribuidor:
  - Emitente:
- Grupo que descreve uma forma alternativa de representar o nome de uma pessoa.** (collapsible section)
- Componentes** (collapsible section)

At the bottom of the form, there are three buttons: **Salvar** (blue), **Executar regras** (grey), and **Limpar** (orange). A small icon of a person's ID card is visible in the bottom right corner.

A interface apresentada na Figura 19 foi gerada a partir do arquétipo openEHR-EHR-OBSERVATION.blood\_pressure.v1. Na parte superior desta é apresentado o campo para seleccionar o indivíduo que será associado ao registro de RES que será enviado para o EHRMan.

Figura 19 - Interface gerada para o arquétipo openEHR-EHR-OBSERVATION.blood\_pressure.v1

The screenshot displays the Smart archEHR Web interface for recording blood pressure. The browser address bar shows the URL: `smartarchehr.garciad.com/smartehr/archtypedata.ehr?archetypeId=openEHR-EHR-OBSERVATION.t`.

**Paciente**

Paciente:

Data de nascimento:

Nome da mãe:

Naturalidade:

[Detalhes](#)

**Informação**

Existem regras que podem ser executadas para o arquétipo.

Existem dados adicionais pendentes. Verifique no canto inferior direito da tela.

**Pressão Arterial**

Data/hora:

**\*blood pressure(en)**

Sistólica:

Diastólica:

Pressão arterial Média:

Pressão de Pulso:

Comentários:

**\*state structure(en)**

Posição:

Fatores confundidores:

Status do sono:

Nível de inclinação:

[Record information about level of exertion.](#)

[estrução de lista](#)

[Salvar](#) [Executar regras](#) [Limpar](#)

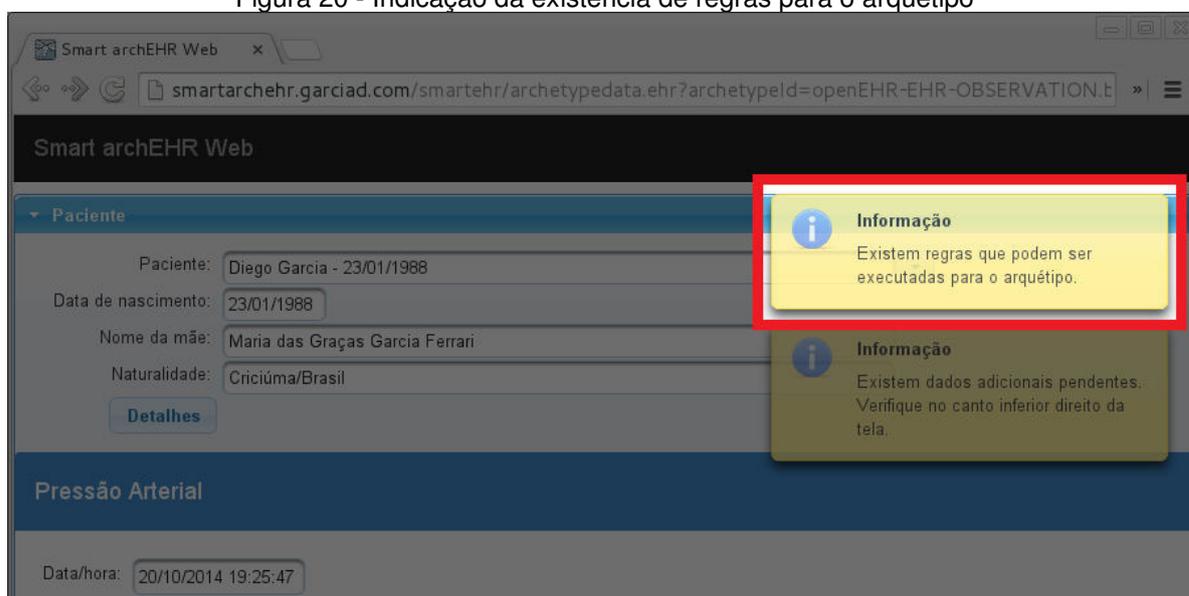
Clique no ícone para ver os dados

Clique no ícone para ver os dados

Fonte: o autor, 2014.

Nas interfaces apresentadas é possível visualizar a indicação da existência de regras definidas a partir de uma diretriz que podem ser executadas nos dados do arquétipo selecionado. A Figura 20 apresenta em destaque esta indicação.

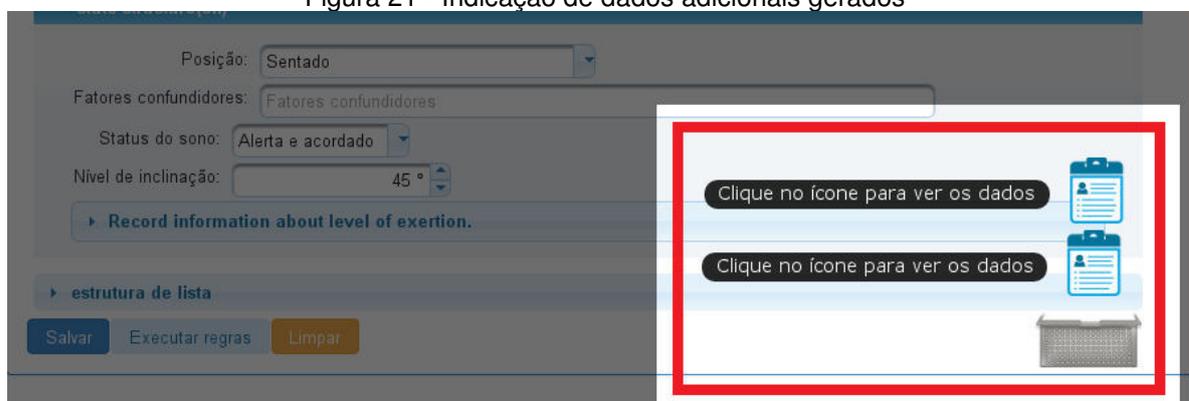
Figura 20 - Indicação da existência de regras para o arquétipo



Fonte: o autor, 2014.

Após a execução das regras, caso sejam criados dados para arquétipos adicionais, além do utilizado na geração da interface, é indicado na interface que existem dados adicionais que serão enviados ao EHRMan juntamente com os dados da tela, conforme Figura 21.

Figura 21 - Indicação de dados adicionais gerados



Fonte: o autor, 2014.

Após a coleta dos dados e execução das regras, estes são enviados para o EHRMan que irá armazená-los no repositório apropriado.

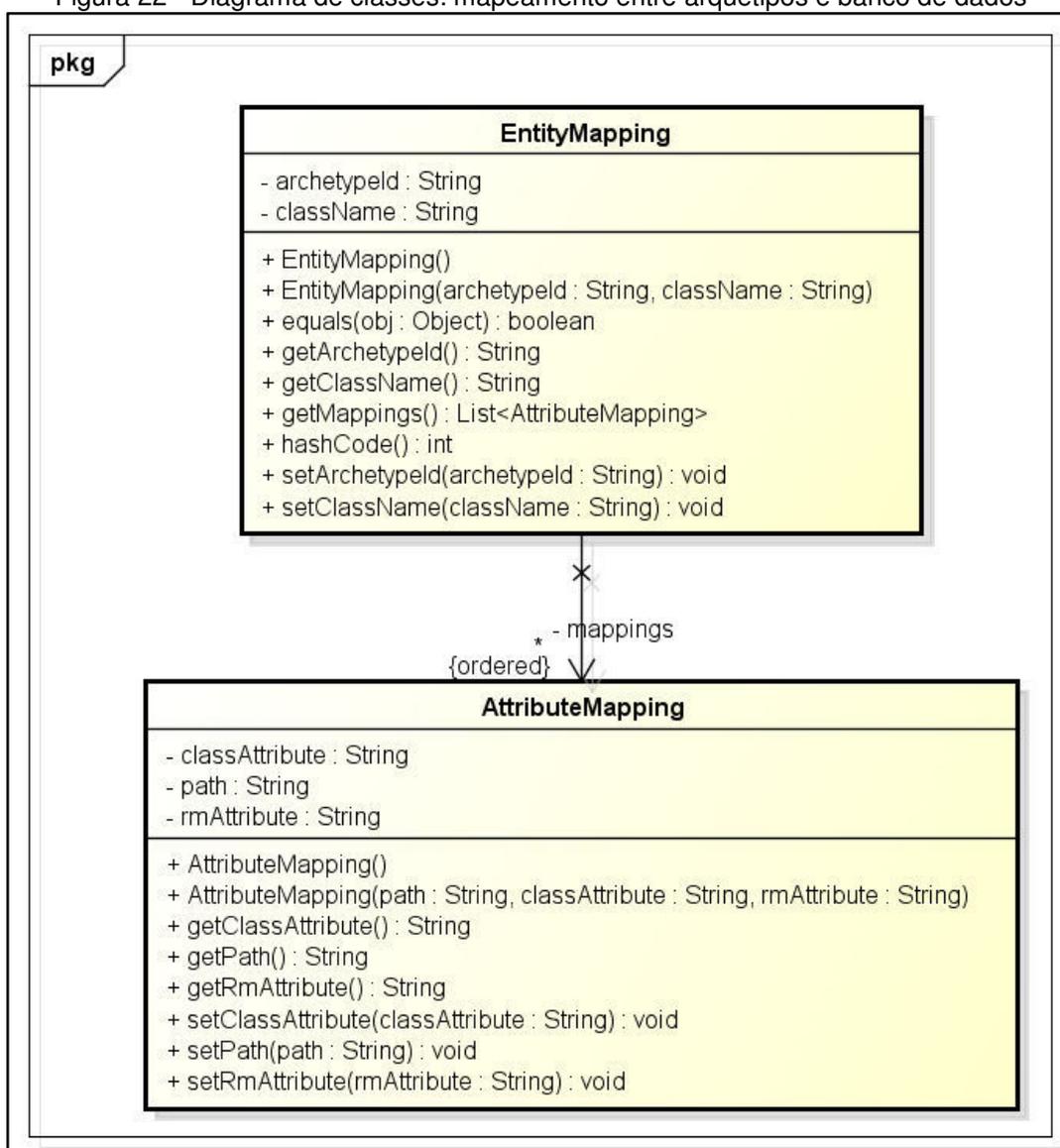
#### 4.4 FORMA DE ARMAZENAMENTO DE DADOS ESTRUTURADOS EM ARQUÉTIPOS

Considerando-se que todos os registros de RES possuem um identificador único, o armazenamento de dados estruturados em arquétipos baseia-se neste princípio onde os registros são armazenados em formato JSON, associados ao identificador do registro de RES e, opcionalmente, indexados por meio de mapeamentos entre arquétipos e entidades do banco de dados.

Este mapeamento permite que determinados dados sejam indexados em tabelas específicas, disponibilizando, por exemplo, uma forma mais eficiente de busca de dados de pacientes a partir do nome deste, sem a necessidade de processar os registros armazenados em JSON. A Figura 22 apresenta as duas classes desenvolvidas para representar mapeamentos entre arquétipos e entidades do BD.

O serviço EHRMan desenvolvido possui um mapa que associa o identificador dos arquétipos a uma lista de mapeamentos, permitindo assim, que os dados de um único arquétipo sejam indexados em mais de uma entidade do BD. Para cada mapeamento encontrado, são buscados os atributos dentro da estrutura hierárquica do arquétipo e atribuído ao respectivo atributo em um objeto que representa a entidade do BD. Em seguida, estes objetos são persistidos por meio do framework de persistência, ficando disponíveis para recuperação.

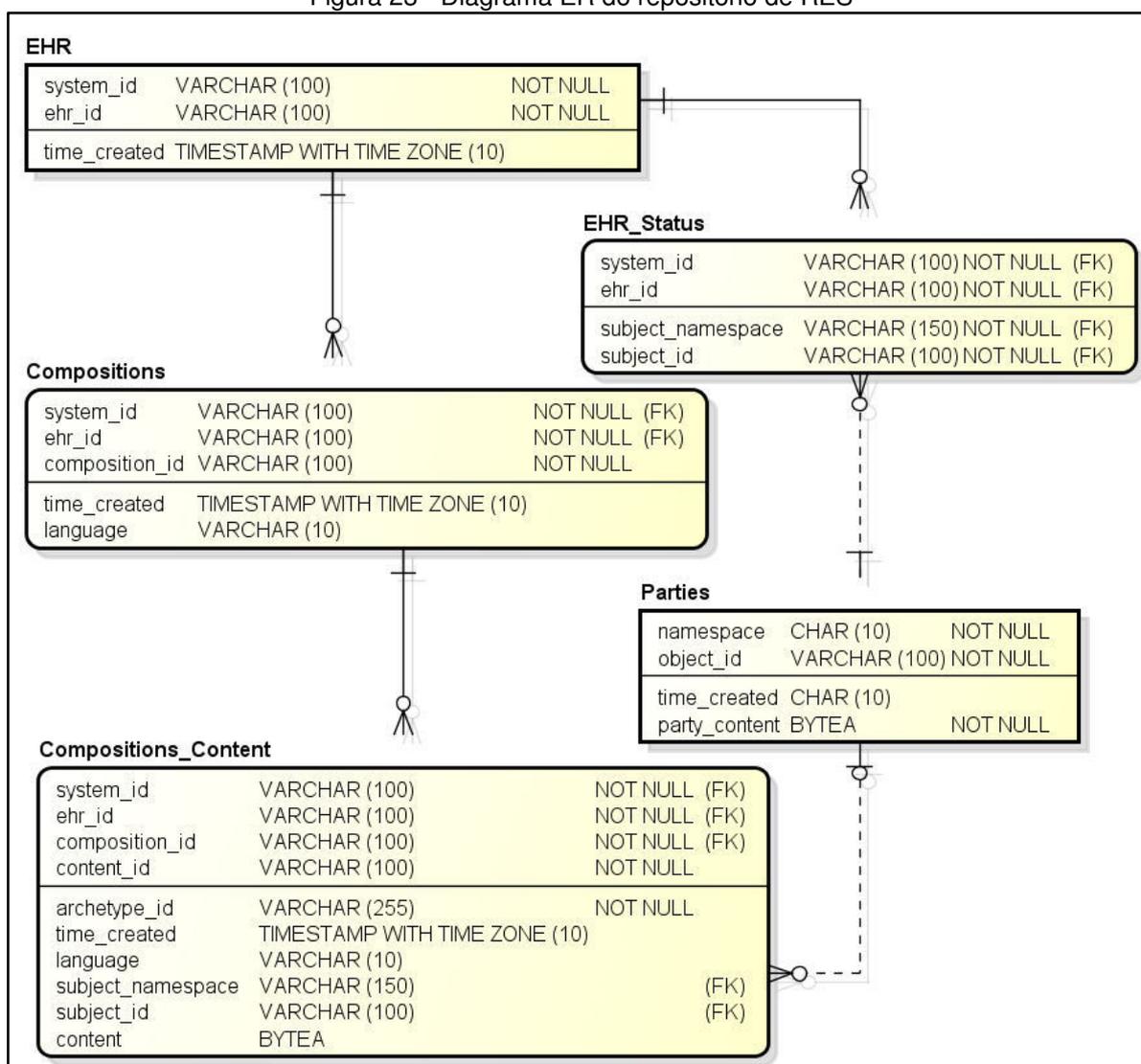
Figura 22 - Diagrama de classes: mapeamento entre arquétipos e banco de dados



Fonte: o autor, 2014.

Além do mapeamento descrito, os dados estruturados em arquétipos são armazenados em formato JSON, mantendo-se a hierarquia definida nos arquétipos. O Apêndice C apresenta um exemplo de dados coletados a partir do arquétipo `openEHR-EHR-OBSERVATION.blood_pressure.v1`, com os dados adicionais do arquétipo `openEHR-EHR-OBSERVATION.res_drc_prevrim_a.v2` contidos em uma *composition*, em formato JSON que foi armazenado no EHRRepo. A estrutura do repositório é apresentada na Figura 23.

Figura 23 - Diagrama ER do repositório de RES



Fonte: o autor, 2014.

Consistindo em 5 entidades, este repositório armazena os dados do RES bem como metadados utilizados na recuperação dos registros armazenados.

Na tabela Parties são armazenados os registros que representam as partes de um RES, podendo ser o indivíduo, participantes de um evento clínico ou organizações, por exemplo. Dados de arquétipos demográficos coletados no sistema são armazenados nesta tabela. A coluna *namespace* armazena o contexto, normalmente o identificador do sistema, que provê os dados do registro, podendo ser especificado o valor *local* para identificar que trata-se de um recurso local. O identificador do registro é armazenado na coluna *object\_id*, que deve ser único dentro de um mesmo contexto. Os dados deste registro, estruturado em arquétipos e representado em JSON é armazenado na coluna *party\_content*.

A tabela EHR armazena os RES dos indivíduos, representando o contêiner principal dos dados. Considerando que o RES deve ser único, cada paciente possui apenas um registro armazenado nesta tabela. A coluna *system\_id* armazena o identificador do sistema que gerou o registro, sendo que nesta pesquisa, é o identificador do próprio sistema desenvolvido. Nesta tabela é armazenado também a data e hora nas quais o RES foi criado.

O paciente ao qual o RES da tabela EHR pertence, é armazenado na tabela EHR\_Status, que contém, além do identificador do sistema de origem e do RES, o contexto da parte relacionada e o identificador desta, de forma a possibilitar a recuperação dos dados do paciente a partir destes parâmetros.

Considerando que um RES possui uma ou mais *Compositions*, a tabela Compositions armazena os metadados destas, relacionado a um RES. De forma a permitir a identificação do idioma utilizado para o registro das *Compositions* esta tabela possui a coluna *language* que armazena o código do idioma, o mesmo utilizado nos arquétipos. Sendo assim, em um mesmo RES podem estar disponíveis *Compositions* que utilizam diferentes idiomas, mantendo a fidelidade aos dados coletados.

Os dados coletados a partir da interface gerada pelo sistema são armazenados na tabela *Compositions\_Content*, que contém além do JSON com os dados estruturados, o identificador único do conteúdo, o identificador do arquétipo que gerou os dados, o idioma utilizado na coleta e o indivíduo relacionado ao registro que, caso seja o próprio paciente, possuirá o mesmo identificador presente no registro da tabela EHR\_Status, bem como o identificador do EHR e Composition aos quais pertence e a data e hora de criação.

Esta estrutura mesmo sendo genérica e pouco detalhada com relação aos dados, foi suficiente para o armazenamento e recuperação dos dados necessários para a execução de regras de decisão baseadas em diretrizes.

#### 4.5 EXECUÇÃO DE REGRAS DE DECISÃO BASEADAS EM DIRETRIZES

Com a utilização da GDL na elaboração das regras a partir das informações extraídas das diretrizes, a execução das regras no Drools Expert foi facilitada, pois tanto a GDL quanto o RM utilizado para a coleta de dados foram desenvolvidos com base na especificação do RM da openEHR. Desta forma, foi necessário apenas o

mapeamento e transformação dos dados coletados diretamente para os objetos definidos na GDL para que as regras fossem executadas.

Considerando que as regras são executadas apenas em instâncias de `DATA_VALUE`, foi necessário transformar os objetos das seguintes classes do RM desenvolvido para as classes da GDL:

- `DV_BOOLEAN`;
- `DV_CODED_TEXT`;
- `DV_COUNT`;
- `DV_DATE`;
- `DV_ORDINAL`;
- `DV_PARAGRAPH`;
- `DV_PROPORTION`;
- `DV_QUANTITY`; e
- `DV_TEXT`.

A transformação das instâncias entre classes de implementações diferentes consiste apenas na cópia das informações de um objeto para outro. Algumas classes como `DV_TIME` e `DV_STATE` não foram transformadas, pois não estavam sendo utilizadas em diretrizes disponíveis no `RulesRepo`. Classes que representam tipos abstratos, como por exemplo, `DV_QUANTIFIED`, `DV_AMOUNT` e a própria classe `DATA_VALUE`, não foram consideradas na transformação, pois não são instanciadas diretamente.

Para a execução das regras, foram criadas as bases de conhecimento a partir das diretrizes, conforme apresentado no método desta pesquisa e submetidas cópias dos objetos transformados ao `Drools Expert`. Após a execução, os dados inseridos no motor de inferência são comparados com os objetos originais e, caso tenham sofrido alterações, são atribuídos aos respectivos nós do objeto que representa o arquétipo do qual foi gerada a interface.

Dados de outros arquétipos, que tenham sido criados em decorrência da execução das regras são disponibilizados também na interface gerada, porém de forma separada, conforme apresentado na Figura 24, para que o usuário do sistema verifique as informações e envie ou não para o `EHRMan` armazenar no repositório de `RES`.

Figura 24 - Dados de arquétipos adicionais

The screenshot shows a web browser window with the URL `smartarchehr.garciad.com/smartehr/archtypedata.ehr?archtypeld=openEHR-EHR-OBSERVATION.E`. The main page displays patient information for Diego Garcia, born on 23/01/1988, with mother Maria das G. and naturalized in Curitiba/Brazil. A 'Pressão Arterial' section is visible with a date of 20/11/2014 19:25:47. A modal window titled 'RES\_DRC\_PrevRim' is open, showing a date of 20/10/2014 19:25:47 and a list of risk factors for DRC:

- Diabetes Mellitus:
- Hipertenso:
- Idoso:
- Obeso:
- História de acidente vascular cerebral:
- História de doença vascular periférica:
- História de insuficiência cardíaca:
- História pessoal de doença arterial coronariana:
- História de DRC na família:
- Fumante:
- Uso de agentes nefrotóxicos:
- Uso de benzilpenicilina:
- Uso de antibióticos aminoglicosídeos:
- Uso de drogas antifúngicas:
- Possui fatores de risco para DRC:

Buttons for 'Salvar' and 'Descartar dados' are at the bottom of the modal.

Fonte: o autor, 2014.

#### 4.6 INTEGRAÇÃO DE DIRETRIZES CLÍNICAS EM UM RES BASEADO EM ARQUÉTIPOS

Como resultado final desta pesquisa, obteve-se a integração de diretrizes clínicas, especificadas em regras de decisão, em um RES baseado em arquétipos, permitindo que a execução das diretrizes seja realizada diretamente no sistema utilizado para a coleta e visualização de dados estruturados em arquétipos.

Para prover a integração de forma desacoplada, o modelo inicialmente proposto apresentado na Figura 5, foi expandido detalhando os componentes que representam o nível de informação. A partir deste modelo, foram especificados serviços e repositórios bem como a forma de comunicação entre eles e o sistema.

O sistema desenvolvido utiliza arquétipos disponíveis em um repositório próprio, ArchRepo, acessado por meio de um serviço RESTful, ArchMan, para a geração de interfaces a partir da definição do arquétipo. Os dados coletados são

então enviados ao serviço EHRMan para serem armazenados no repositório de RES, o EHRRepo. A estrutura deste repositório permite uma representação simples dos dados coletados, em forma de RES associados ao paciente, bem como a identificação do arquétipo utilizado na coleta dos dados.

O serviço de regras – RulesMan – manipula as regras em formato GDL. Desta forma, é necessário que as diretrizes a serem incorporadas ao sistema sejam analisadas, sejam extraídas as regras de decisão e seu conteúdo seja especificado em GDL. Além da estruturação das regras em GDL, nesta etapa também são buscados os arquétipos já existentes em repositórios disponíveis e novos arquétipos, quando necessário, são modelados.

A Figura 25 apresenta um trecho extraído das Diretrizes Clínicas para o cuidado ao paciente com Doença Renal Crônica – DRC no Sistema Único de Saúde (BRASIL, 2014) onde foi aplicado o método descrito por Garcia et al. (2013) e extraídas as informações da diretriz (GARCIA, CINTHO e MORO, 2014a).

Figura 25 - Trecho retirado das Diretrizes Clínicas para o cuidado ao paciente com Doença Renal Crônica - DRC no Sistema Único de Saúde

Os indivíduos sob o risco de desenvolver DRC são:

Pessoas com **diabetes** (quer seja do tipo 1 ou do tipo 2): o diagnóstico do diabetes deve ser realizado de acordo com o nível sérico da **glicemia de jejum acima de 126 mg/dL, ou acima de 200 mg/dL 2 horas após a ingestão de 75g de glicose, ou qualquer valor de hiperglicemia**, na presença de **sintomas clássicos**, como **poliúria, polidipsia** ou **polifagia**.

Fonte: adaptado de Cintho, 2014.

A partir das informações identificadas na diretriz apresentada, foram extraídos termos, tipo de termo, formas de identificação, condição entre outros dados. O Quadro 37 apresenta as informações extraídas a partir do trecho da diretriz destacadas na Figura 25.

Quadro 37 - Informações extraídas a partir do trecho da diretriz

Termo	Tipo termo	Terminologia SNOMED-CT	Forma de identificação	Condição
Diabetes	Doença	Diabetes Mellitus	Exame laboratorial de glicemia de jejum	Acima de 126mg/dL
			Exame laboratorial de glicemia, 2 horas após a ingestão de 75g de glicose	Acima de 200mg/dL
			Diabetes Mellitus	Verdadeiro ou Falso

Fonte: adaptado de Cintho, 2014.

A partir das informações extraídas, foram identificados em repositórios disponíveis ou modelados os arquétipos necessários e as regras contidas na diretriz especificadas em GDL, permitindo assim sua inclusão no repositório de regras. A Figura 26 demonstra a ferramenta visual GDL Tools utilizada na elaboração da regra para o trecho apresentado na Figura 25.

Figura 26 - Regra para diagnóstico de diabetes na ferramenta GDL Tools

The screenshot displays the GDL Editor - Guide interface. The main workspace shows a rule structure under 'Conditions & Actions'. The rule is defined as follows:

```

((
  Element Blood glucose >= 126 mg/dl
  and
  Element Intake == Fasting
) or (
  ((
    Element Blood glucose >= 200 mg/dl
    and
    Element Intake == Random
    and
    Element Symptoms == Polyuria
  ) or (
    Element Blood glucose >= 200 mg/dl
    and
    Element Intake == Random
    and
    Element Symptoms == Excessive thirst
  ))
) or (
  Element Blood glucose >= 200 mg/dl
  and
  Element Intake == Random
  and
  Element Symptoms == Excessive thirst
)

```

The 'Actions' panel contains the following actions:

- Set element **Has Risk Factors for CKD** to **true**
- Set element **Diabetes Mellitus** to **true**

The 'Conditions' legend on the right lists the following operators and actions:

- Compare (DataValue)
- Compare (NullValue)
- Compare (Element)
- Compare (Attribute)
- Element exists
- Or operator

The 'Actions' legend on the right lists the following actions:

- Set (DataValue)
- Set (NullValue)
- Set (Element)
- Set (Attribute)

Fonte: o autor, 2014.

As regras elaboradas em GDL utilizam as estruturas dos arquétipos em sua estruturação. Desta forma, o serviço de regras, RulesMan, extrai os mapeamentos realizados com os arquétipos associados de forma a permitir ao sistema que utiliza suas regras, recuperar os arquétipos necessários. Este mapeamento pode ser observado na Figura 27. No canto superior direito é apresentado o arquétipo no editor de arquétipos e tela à esquerda apresenta a regra no GDL Editor. A seta indica o mapeamento entre um componente da estrutura do arquétipo e um elemento da regra.

No quadro em vermelho na parte inferior à esquerda da Figura 27 estão destacados em azul, o identificador do arquétipo (openEHR-EHR-OBSERVATION.res\_drc\_prevrim\_a.v1), o identificador atribuído ao componente do arquétipo utilizado na regra (gt0026) e o caminho deste (/data[at0001]/events[at0002]/data[at0003]/items[at0004]). Em seguida, é possível visualizar este componente sendo utilizado na regra, por meio do identificador atribuído.

Os identificadores de cada componente do arquétipo utilizado na regra podem ser visualizados em destaque na cor vermelha, no quadro em azul, na parte inferior à direita, também na Figura 27. Além do identificador, estão destacados também, os atributos que compõem o caminho do elemento dentro da estrutura do arquétipo.

O conteúdo destas regras, em GDL é então, armazenado no repositório de regras, o RulesRepo, por meio do RulesMan que também é o responsável pela disponibilização destas regras ao RES, onde serão executadas.

Figura 27 - Mapeamento entre regras em GDL e arquétipos

The image shows two software windows side-by-side. The left window is the 'GDL Editor' for 'RES\_DRC\_PrevRim'. It displays a 'Conditions & Actions' panel with a complex logical expression involving elements like 'Diabetes Mellitus', 'Hypertension', 'Elderly', 'Obesity', and 'Coronary arteriosclerosis'. A red arrow points from the 'Diabetes Mellitus' condition in the GDL editor to the 'Diabetes Mellitus' element in the archetype editor's tree view. The right window is the 'Archetype Editor' for 'openEHR-EHR-OBSERVATION.res\_drc\_prevrim\_a.v1', showing a tree structure with elements 'Diabetes Mellitus', 'Hypertension', 'Elderly', and 'Obesity'. Below the archetype editor, a light blue box contains a JSON-like structure representing the archetype's data and events. Below the GDL editor, a light red box contains the GDL rule definition for the archetype.

**GDL Rule Definition (Red Box):**

```
[6] = (ARCHETYPE_BINDING) <
  archetype_id = <"openEHR-EHR-OBSERVATION.res_drc_prevrim_a.v1">
  domain = <"CDS">
  elements = <
    ["gt0026"] = (ELEMENT_BINDING) <
      path = <"/data[at0001]/events[at0002]/data[at0003]/items[at0004]">
    >
  [...]
  ["gt0071"] = (RULE) <
    when = <"($gt0026==true)||(($gt0028==true)||(($gt0029==true)||[...]">
```

**Archetype Structure (Blue Box):**

```
OBSERVATION[at0000] matches { -- RES_DRC_PrevRim
  data matches {
    HISTORY[at0001] matches { -- Event Series
      events cardinality matches {1..*; unordered} matches {
        EVENT[at0002] occurrences matches {0..1} matches { -- Any event
          data matches {
            ITEM_TREE[at0003] matches { -- Tree(pt)
              items cardinality matches {0..*; unordered} matches {
                ELEMENT[at0004] occurrences matches {0..1} matches { -- Diabetes Mellitus
                  value matches {
                    DV_BOOLEAN matches {
                      value matches {True, False}
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Fonte: adaptado de Cintho, 2014.

Considerando que ainda não existe RES baseado em arquétipos disponível para a integração das regras, foi desenvolvido o RES utilizando a plataforma JavaEE e o framework JSF, resultando em um sistema Web, para a coleta dos dados e execução das regras. Este sistema comunica-se com o serviço ArchMan para buscar os arquétipos que podem ser utilizados e gera, de forma totalmente automática, a interface para a coleta e visualização de dados baseados em estruturas definidas por arquétipos. O RES comunica-se também com o serviço EHRMan para o armazenamento e recuperação dos dados coletados, necessários para a execução das regras. Alguns exemplos de interfaces geradas para os arquétipos são demonstrados na Figura 17, Figura 18 e Figura 19.

De forma a prover o suporte a decisão necessário, o sistema comunica-se com o serviço de regras, RulesMan, que manipula as regras baseadas em diretrizes armazenadas no repositório RulesRepo, descrito anteriormente e, utilizando o motor de inferência Drools Expert executa estas regras. A Figura 28 exemplifica a execução de regras pelo motor de inferência apresentado na Figura 11.

No exemplo da figura, foram consideradas as regras simplificadas apresentadas no Quadro 38. Estas regras não incluem necessariamente todas as condições necessárias, apenas servem para demonstrar a execução das regras.

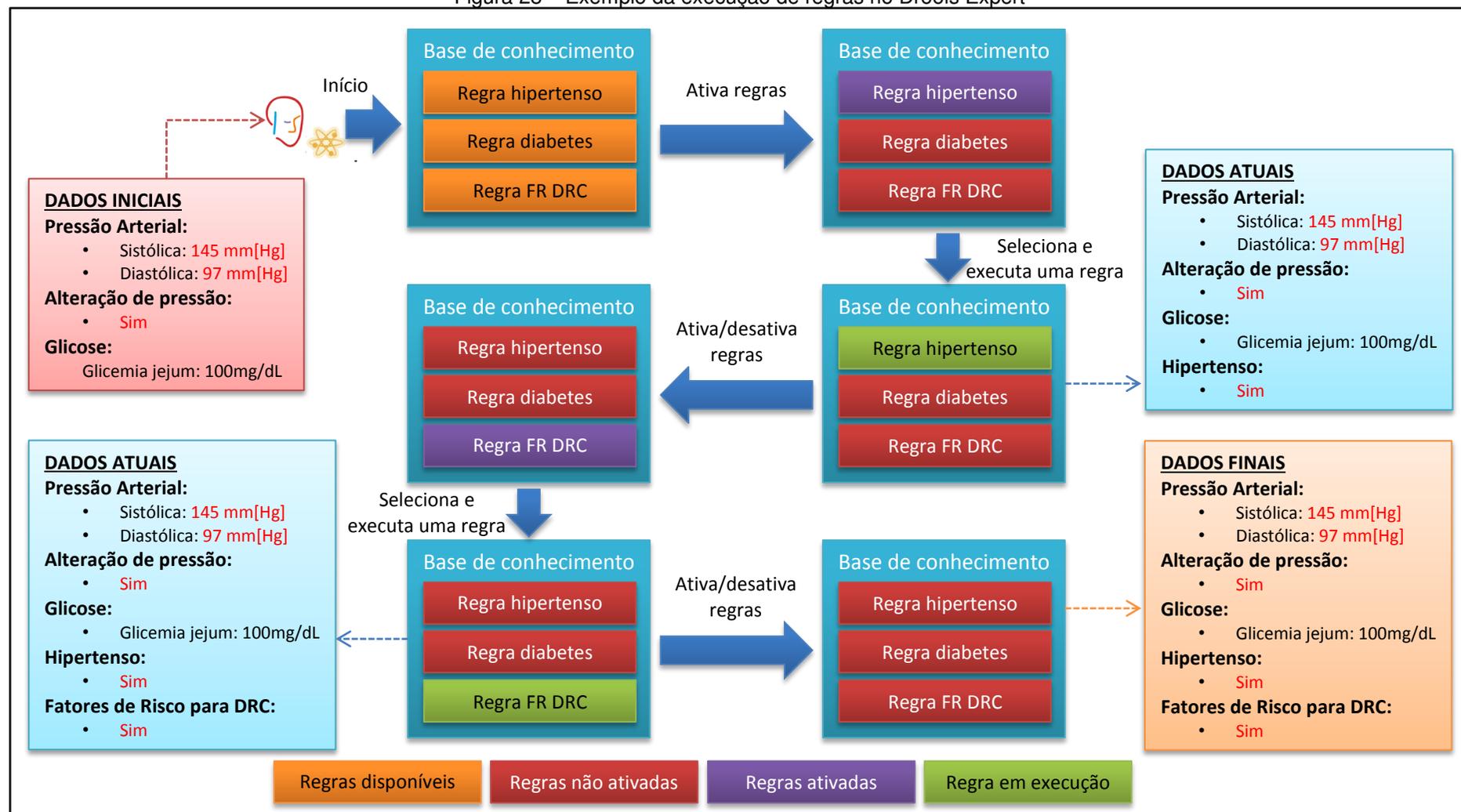
Quadro 38 - Regras utilizadas na exemplificação do funcionamento do Drools Expert

Regra hipertenso	Alteração de pressão: sim E Pressão sistólica $\geq$ 140mm[Hg] E Pressão diastólica $\geq$ 90mm[Hg]
Regra diabetes	Glicemia de jejum > 126mg/dL
Regra FR DRC	Hipertenso = sim OU Idoso = sim OU Diabetes = sim

Fonte: o autor, 2014.

A execução de regras pelo Drools Expert, exemplificada na Figura 28, inicia a partir da construção de uma base de conhecimento, contendo as regras apresentadas no Quadro 38. Estas regras, definidas em DRL ficam, então, disponíveis para execução de acordo com os dados, chamados de fatos, que são inseridos na memória de trabalho do motor de inferência. Estes fatos são os dados de RES coletados a partir da estrutura dos arquétipos. Os dados iniciais, submetidos ao motor de inferência são apresentados no retângulo em vermelho da Figura 28.

Figura 28 – Exemplo da execução de regras no Drools Expert



Fonte: o autor, 2014.

Em seguida, as regras são avaliadas e ativadas para execução, de acordo com os dados disponíveis. Uma das regras é selecionada e executada, podendo atualizar os fatos inseridos ou criar novos fatos, conforme apresentado no retângulo em azul à direita (Dados atuais), na Figura 28, onde foi gerado um novo dado indicando o diagnóstico de hipertensão (Hipertenso = sim). As regras são então reavaliadas e ativadas ou desativadas de acordo com os dados atualizados e/ou criados. Este processo se repete até que nenhuma regra seja ativada para execução.

Com a integração do Drools Expert ao RES desenvolvido, os dados presentes na memória de trabalho do motor de inferência, ao final da execução (retângulo em laranja, na Figura 28) são então devolvidos ao RES, onde são verificados os dados que foram criados e os que foram atualizados, disponibilizando-os ao usuário para que este verifique as informações e decida de devem ou não ser enviadas ao EHRMan para serem persistidas no repositório, conforme exemplificado na Figura 24.

Com os passos descritos, foi possível desenvolver um RES que atenda à necessidade de coleta e armazenamento de dados estruturados em arquétipos com suporte a decisão baseado em diretrizes clínicas integradas ao RES, permitindo aos usuários utilizarem um único sistema para estas duas tarefas, evitando retrabalho e facilitando a utilização das diretrizes clínicas no cuidado aos pacientes.

## 5 DISCUSSÃO

Esta pesquisa tem como diferencial, a preocupação na integração de diretrizes ao RES utilizando padrões de interoperabilidade para RES, os mesmos utilizados no próprio sistema. Geralmente os trabalhos que relatam a utilização de diretrizes em sistemas computacionais tais como o de Gonzalez-Ferrer et al. (2014), utilizam padrões específicos para diretrizes, não resolvendo, de fato, o problema de integração de RES e diretrizes clínicas, uma vez que ainda é necessário um formato para integrar a estrutura dos dados à estrutura das diretrizes.

Estudos realizados no Brasil que apresentam a utilização de arquétipos focam essencialmente na representação dos dados, como os de Dias, Cook e Freire (2011) com o padrão de Troca de Informações na Saúde Suplementar (TISS) e Santos, Bax e Kalra (2012) para o Departamento de Saúde do Estado de Minas Gerais, porém não descrevem e não aplicam o processamento dos mesmos no RES.

Apesar de estar disponível uma implementação do RM openEHR (OPENEHR, 2009), não foram encontrados trabalhos que descrevem a utilização da implementação de referência disponibilizada pela openEHR no desenvolvimento de RES, ou de sua adaptação. Assim foi necessário construir uma implementação parcial, adaptada aos componentes necessários para o RES desta pesquisa. Considerando-se que, além de não utilizar as convenções estabelecidas na linguagem de programação Java, por exemplo, em métodos de acesso a atributos da classe, a implementação de referência disponibilizada exige que os dados a serem informados nas estruturas do RM sejam conhecidos no momento da instanciação das classes. Como em um sistema os dados ainda não são conhecidos, era necessário que a implementação do RM utilizada permitisse que estes dados fossem atribuídos a suas respectivas estruturas a qualquer momento.

O relato de desenvolvimento e da utilização de uma implementação do RM possibilitará que estudos futuros focados na aplicação de arquétipos em RES possam se basear nesta dissertação, bem como a solução proposta pode ser utilizada na prática para implementação destes sistemas. É importante destacar que esta é uma necessidade no Brasil, uma vez que a portaria 2.073 do Ministério da Saúde, de 31 de agosto de 2011 define que para o RES deve ser utilizada RM da openEHR (BRASIL, 2011).

A geração automática da interface para a coleta e visualização dos dados que constitui um dos resultados desta pesquisa é um assunto pouco relatado na literatura e geralmente possui pouco ou nenhum detalhamento, de forma que não é possível basear-se nesta literatura para o desenvolvimento deste componente do RES. No trabalho de Duftchmid, Chaloupka e Rinner (2013), por exemplo, foi proposta uma abordagem para gerar automaticamente formulários em RES legados a partir de arquétipos, permitindo a exportação dos dados mapeados. Contudo não é detalhado todo o processo de geração destes formulários, apenas parte deste e também não é descrito como é feita a associação entre os componentes dos formulários e seus correspondentes no RM openEHR.

Desta forma, considerando a inexistência de sistemas com geração automática de interfaces adequados a esta pesquisa, fez-se necessário o desenvolvimento deste componente. Sabe-se que a usabilidade é um fator importante e que pode levar a uma resistência dos profissionais em utilizar o RES. Porém, considerando que as interfaces para interação com os usuários não existiam e precisaram ser desenvolvidas, os estudos relacionados à usabilidade do RES desenvolvidos não puderam ser realizados, uma vez que não foi possível utilizá-lo na prática, com usuários reais. Desta forma, sugere-se como trabalho futuro, a aplicação do RES na prática clínica, permitindo assim avaliar o RES quanto a questões relacionadas à interação humano-computador.

Complementando a contribuição científica apresentada, a inserção social deste trabalho se dá à medida que pode ser diretamente utilizada em sistemas que tendem a facilitar o acesso à informação, permitindo melhor qualidade nos atendimentos de saúde.

Além da questão de interoperabilidade, o modelo de desenvolvimento proposto foi implementado e aplicado em um cenário real da saúde, com regras de decisão baseadas em uma diretriz clínica elaborada por uma sociedade de especialistas e adotada nos atendimentos no Brasil (CINTHO, 2014). Foi priorizada a utilização de arquétipos disponíveis no CKM da Fundação openEHR, que é uma das vantagens desta arquitetura e que facilita a interoperabilidade, evitando a duplicação de conceitos existentes e a tendência do desenvolvimento muito específico, em um nível regional.

Para a especificação das diretrizes integradas ao RES foi utilizada a GDL, linguagem esta que até o término desta pesquisa foi citada e utilizada apenas por

Chen et al. (2013), autor da mesma, e no estudo de Anani et al. (2014), que Rong também colabora, e pelo projeto na qual a presente pesquisa está inserida.

Em relação à validação do processamento de diferentes arquétipos pelo RES desenvolvido, os que utilizam o RM da openEHR foram processados corretamente e puderam ser utilizados para a coleta de dados e execução de regras de decisão. Contudo, os arquétipos elaborados a partir do RM da ISO/EN 13606, obtidos do Portal Público da Base de Registro Eletrônico em Saúde de Minas Gerais não puderam ser processados pelo RES. Esta restrição se deu pelo fato de que o modelo openEHR é mais detalhado e possui uma especialização maior das classes que representam o conteúdo clínico.

Considerando que a pesquisa foi baseada na arquitetura e especificação openEHR, para que os arquétipos do RM ISO/EN 13606 possam ser processados pelo RES desenvolvido, é necessário um mapeamento entre as classes destes dois RM, sendo que esta não é uma tarefa trivial, pois deve-se identificar nos arquétipos, o tipo de conteúdo clínico que está sendo especificado para que se possa definir a estrutura correta no RM openEHR. O modelo de desenvolvimento resultante desta pesquisa possui uma abstração no acesso aos recursos necessários, o que permite que sejam empregadas diferentes tecnologias, adequadas a cada um dos recursos, de forma que o RES e suas funcionalidades possam ser atualizados e aprimorados, sem a necessidade de reestruturação em toda a arquitetura e com menor impacto nos componentes envolvidos.

A utilização de repositórios e serviços, abordagem utilizada no desenvolvimento do modelo, permite, por exemplo, que sejam utilizados diferentes formas de armazenamento, garantindo uma maior disponibilidade dos dados de RES. No trabalho de Sundvall et al. (2013) foi aplicada a arquitetura REST no desenvolvimento de um RES porém o foco foi na formalização do armazenamento e recuperação de dados baseados em openEHR e no controle de versões destes dados. Não foi focada a integração das diretrizes ao RES e não foi descrito ou sugerido como deve ser realizada esta integração.

Nesta pesquisa os dados foram armazenados em formato JSON, por ser de fácil utilização e entendimento e por manter a hierarquia dos dados, semelhante à apresentada nos arquétipos, o que demonstrou ser adequado e suficiente para o objetivo desta pesquisa.

Contudo é importante citar que não foram realizados testes de desempenho, com relação ao armazenamento e recuperação destes dados. Esta característica não foi abordada nesta pesquisa, porém sabe-se que é uma preocupação pertinente a sistemas críticos, como os sistemas de saúde, nos quais inclui-se o RES, bem como para o uso secundário da informação como em pesquisas epidemiológicas e acompanhamento (FREIRE et al., 2012). Existem grupos de pesquisa que realizam estudos acerca de implementações e desempenho da persistência de RES em arquétipos, como Frade et al. (2013) e Freire et al. (2012) Por isso sugere-se que trabalhos futuros desenvolvidos a partir desta pesquisa incluam a análise e avaliação do desempenho da manipulação dos dados de RES no modelo proposto.

A implementação a partir do modelo resultante desta pesquisa permitiu uma integração completa entre o RES baseado em arquétipos e as diretrizes clínicas, considerando que os dados necessários para a execução das regras de decisão eram coletados no próprio sistema. Com isso, não se faz necessária a utilização de sistemas isolados, para a coleta e para o suporte a decisão, bem como a utilização de diferentes padrões e formalismos na definição do conteúdo do RES e das regras de decisão. Este é outro diferencial desta pesquisa, pois estudos publicados relatam a utilização de softwares específicos para a execução das regras a partir de diretrizes, como por exemplo, no estudo de Anani et al. (2014) que utilizou uma ferramenta para os testes das regras especificadas em GDL isolada do RES.

De forma a prover maior interoperabilidade, abrangendo RES que utilizam diferentes formas de representação de dados padronizados, como por exemplo, o HL7, sugere-se como trabalho futuro o mapeamento entre o modelo de arquétipo openEHR e o modelo HL7, considerando-se que existem iniciativas como a publicada por Gonzalez-Ferrer et al. (2013) que utiliza o modelo de Registro Médico Virtual do HL7 em um SAD distribuído.

Considerando-se o aspecto de saúde desta pesquisa, foram utilizados arquétipos e diretriz relacionados à prevenção da DRC, pois tem-se identificado um aumento na incidência e prevalência deste problema no Brasil, elevando os custos com tratamento e reduzindo a qualidade de vida dos indivíduos (MINISTÉRIO DA SAÚDE/DATASUS, 2008), sendo que, conforme Barretto et al. (2003), a utilização de SAD permite um diagnóstico mais ágil e de acordo com as recomendações definidas nas diretrizes.

Baseado no aumento na incidência e prevalência da DRC no Brasil percebe-se que este cenário, pode ser alterado por meio de ações de diagnóstico precoce e tratamento adequadas. Considerando-se a utilização das diretrizes clínicas por meio de SAD, mesmo que não completamente integradas ao RES, existe um aumento no desempenho da equipe profissional envolvida, conforme descrito por Barretto et al. (2003), permitindo diagnóstico mais ágil e de acordo com as recomendações estabelecidas nestas diretrizes. Ainda assim, a maioria das diretrizes não foca na prevenção, mas no diagnóstico, monitoramento e tratamento dos problemas de saúde.

Desta forma, foi utilizada para testes do RES desenvolvido, uma diretriz já especificada em GDL por Cintho (2014) para a prevenção da DRC, porém outras diretrizes podem ser integradas, considerando que o RES não é específico para uma única diretriz.

## 6 CONCLUSÃO

Foi possível integrar diretrizes clínicas no RES baseado em arquétipos, fornecendo o suporte a decisão de acordo com a definição da diretriz clínica utilizada. O RES desenvolvido favorece a interoperabilidade considerando que este utiliza apenas arquétipos como modelo de conhecimento.

A forma de armazenamento de dados padronizados, baseados na arquitetura de modelagem de dois níveis foi especificada e implementada de forma a atender às necessidades do RES resultante desta pesquisa, que consiste na execução de regras em dados padronizados e estruturados em arquétipos.

Foi desenvolvido um mecanismo de leitura e processamento de arquétipos para a geração de interfaces para a coleta e apresentação dos dados utilizados também na execução de regras de decisão. Para este último, foi elaborado o mecanismo necessário para o processamento das regras extraídas das diretrizes clínicas permitindo o suporte a decisão utilizando esses dados padronizados, utilizando o motor de inferência Drools Expert.

Por fim, foi construído um Sistema de Apoio a Decisão para a prevenção da DRC completamente integrado ao RES baseado na arquitetura de arquétipos openEHR, a partir do modelo proposto, utilizando os componentes implementados.

Com isso, conclui-se que é possível integrar diretrizes clínicas em um RES baseado em arquétipos, provendo o suporte a decisão e interoperabilidade em um único sistema.

### 6.1 TRABALHOS FUTUROS

A partir dos resultados desta pesquisa, sugere-se como trabalhos futuros:

- Aplicar diretrizes de outras áreas no RES desenvolvido, como diretrizes clínicas para fibrilação atrial, por exemplo;
- Realizar o mapeamento entre o RM openEHR e o RIM HL7, permitindo que sistemas desenvolvidos com estes modelos possam interoperar;
- Avaliar o desempenho da forma de armazenamento proposta, identificando novas formas e técnicas de armazenamento;
- Propor um método de avaliação para o RES desenvolvido;

- Realizar avaliação do RES na prática clínica, comparando, por exemplo, se existe diminuição de pacientes encaminhados à atenção secundária, considerando que a diretriz aplicada é direcionada à prevenção, bem como se a utilização do RES com o suporte a decisão baseado nessa diretriz permitiu reduzir encaminhamentos errôneos;
- Avaliar a usabilidade das interfaces geradas automaticamente, propondo melhorias e adequações aos diferentes contextos onde o RES possa ser utilizado.

## REFERÊNCIAS

- ANANI, N. et al. Retrospective checking of compliance with practice guidelines for acute stroke care: a novel experiment using openEHR's Guideline Definition Language. **BMC medical informatics and decision making**, v. 14, p. 39, doi:10.1186/1472-6947-14-39, 2014.
- BALI, M. **Drools JBoss Rules 5.0 Developer's Guide**. Birmingham: Packt Publishing, 2009. p. 338
- BARRETTO, S. A. et al. Linking guidelines to Electronic Health Record design for improved chronic disease management. **AMIA Annu Symp Proc.**, p. 66–70, 2003.
- BARROS, E. et al. **Nefrologia: Rotinas, Diagnóstico e Tratamento**. 3<sup>a</sup>. ed. Porto Alegre: Artmed, 2006. p. 620
- BASTOS, R. M. R.; BASTOS, M. G. e TEIXEIRA, M. T. B. A Doença Renal Crônica e os Desafios na Atenção Primária à Saúde na sua Detecção Precoce. **Revista APS**, v. 10, n. 1, p. 46–55, 2007.
- BEALE, T. **The openEHR Reference Model: Integration Information Model. Release 1.0.2.** . [S.l: s.n.], 2006.
- BEALE, T. et al. **The openEHR Reference Model: Common Information Model. Release 1.0.2.** . [S.l: s.n.], 2008a.
- BEALE, T. et al. **The openEHR Reference Model: Data Structures Information Model. Release 1.0.2.** . [S.l: s.n.], 2008b.
- BEALE, T. et al. **The openEHR Reference Model: Data Types Information Model. Release 1.0.2.** . [S.l: s.n.], 2008c.
- BEALE, T. et al. **The openEHR Reference Model: Demographic Information Model. Release 1.0.2.** . [S.l: s.n.], 2008d.
- BEALE, T. et al. **The openEHR Reference Model: EHR Information Model. Release 1.0.2.** . [S.l: s.n.], 2008e.
- BEALE, T. et al. **The openEHR Reference Model: Support Information Model. Release 1.0.2.** . [S.l: s.n.], 2008f.
- BEALE, T. e HEARD, S. **openEHR Release 1.0.2: Archetype Definitions and Principles.** . [S.l: s.n.], 2007.
- BEALE, T. e HEARD, S. **openEHR Architecture: Architecture Overview. Release 1.0.2.** . [S.l: s.n.], 2008.

BEMMEL, J. H. VAN e MUSEN, M. A. **Handbook of Medical Informatics**. Heidelberg-New York: Springer-Verlag, 1997. p. 628

BENSON, T. **Principles of Health Interoperability HL7 and SNOMED**. London: Springer London, 2010. p. 263

BIDGOOD, W. D. et al. Understanding and using DICOM, the data interchange standard for biomedical imaging. **Journal of the American Medical Informatics Association : JAMIA**, v. 4, n. 3, p. 199–212, 1997.

BOXWALA, A. A. et al. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. **Journal of biomedical informatics**, v. 37, n. 3, p. 147–61, doi:10.1016/j.jbi.2004.04.002, 2004.

BRASIL. **Ministério da Saúde. Secretaria de Atenção à Saúde. Departamento de Atenção Básica. Prevenção Clínica Cardiovascular, Cerebrovascular e Renal Crônica**. , Caderno de atenção básica. Brasília: Editora MS, 2006.

BRASIL. **Ministério da Saúde. Portaria Nº 2.073 de 31 de agosto de 2011 – Regulamenta o uso de padrões de interoperabilidade e informação em saúde para sistemas de informação em saúde no âmbito do Sistema Único de Saúde, nos níveis Municipal, Distrital, Estadual e .** [S.l: s.n.], 2011.

BRASIL. **Diretrizes Clínicas para o Cuidado ao Paciente com Doença Renal Crônica - DRC no Sistema Único de Saúde**. . Brasília: [s.n.]. Disponível em: <<http://sonerj.org.br/wp-content/uploads/2014/03/diretriz-cl-nica-drc-versao-final2.pdf>>. Acesso em: 25 out. 2014, 2014.

CHEN, R. et al. Early Experiences from a guideline-based computerized clinical decision support for stroke prevention in atrial fibrillation. **Studies in health technology and informatics**, v. 192, p. 244–7, 2013.

CHEN, R.;; GEORGII-HEMMING, P. e ÅHLFELDT, H. Representing a Chemotherapy Guideline Using openEHR and Rules. **Studies in Health Thechnology And Informatics**, v. 150, p. 653–657, doi:10.3233/978-1-60750-044-5-653, 2009.

CICOGNA, P. E. S. L. **Integração de Diretrizes Médicas ao Prontuário Eletrônico: Método padronizado baseado em arquétipos**. Dissertação (Mestrado em Tecnologia em Saude) - Pontifícia Universidade Católica do Paraná, Curitiba - Paraná. 2013.

CINTHO, L. M. M. **Modelo de Registro Eletrônico de Saúde para Apoio à Prevenção da Doença Renal Crônica**. Dissertação (Mestrado em Tecnologia em Saúde) - Pontifícia Universidade Católica do Paraná, Curitiba - Paraná. 2014.

DIAS, R. D.;; COOK, T. W. e FREIRE, S. M. Modeling healthcare authorization and claim submissions using the openEHR dual-model approach. **BMC medical informatics and decision making**, v. 11, n. 1, p. 60, doi:10.1186/1472-6947-11-60, 2011.

DOLIN, R. H. et al. An update on HL7's XML-based document representation standards. **Proc AMIA Symp.**, p. 190–4, 2000.

DOMBAL, F. T. DE et al. Computer-aided Diagnosis of Acute Abdominal Pain. **BMJ**, v. 2, n. 5804, p. 9–13, doi:10.1136/bmj.2.5804.9, 1972.

DUFTSCHMID, G.; CHALOUPKA, J. e RINNER, C. Towards plug-and-play integration of archetypes into legacy electronic health record systems: the ArchiMed experience. **BMC medical informatics and decision making**, v. 13, p. 11, doi:10.1186/1472-6947-13-11, 2013.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. University of California, Irvine - [S.I.]. 2000.

FORGY, C. L. Rete: A fast algorithm for the many pattern/many object pattern match problem. **Artificial Intelligence**, v. 19, n. 1, p. 17–37, doi:10.1016/0004-3702(82)90020-0, 1982.

FRADE, S. et al. Survey of openEHR storage implementations. In: PROCEEDINGS OF THE 26TH IEEE INTERNATIONAL SYMPOSIUM ON COMPUTER-BASED MEDICAL SYSTEMS. **Anais...** [S.I.]: IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6627806>>. Acesso em: 23 nov. 2014, 2013.

FREIRE, S. M. et al. Performance of XML Databases for Epidemiological Queries in Archetype-Based EHRs. In: SCANDINAVIAN CONFERENCE ON HEALTH INFORMATICS. **Anais...** Linkoping: [s.n.], 2012.

GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. [S.I.]: Addison-Wesley Professional, 1994. p. 395

GARCIA, D. et al. Method to integrate clinical guidelines into the electronic health record (EHR) by applying the archetypes approach. **Studies in health technology and informatics**, v. 192, p. 871–5, 2013.

GARCIA, D.; CINTHO, L. M. M. e MORO, C. M. C. Integrando Regras de Decisão de Guidelines a Sistemas com Arquétipos. In: XIV CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAUDE. **Anais...** Santos, SP: In press, 2014a.

GARCIA, D.; CINTHO, L. M. M. e MORO, C. M. C. Electronic health record to support Chronic Kidney Disease prevention — Integrating guidelines and archetypes. In: IEEE-EMBS INTERNATIONAL CONFERENCE ON BIOMEDICAL AND HEALTH INFORMATICS (BHI). **Anais...** [S.I.]: IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6864337>>. Acesso em: 20 out. 2014b, 2014.

GARDE, S. et al. Towards semantic interoperability for electronic health records. **Methods of information in medicine**, v. 46, n. 3, p. 332–43, doi:10.1160/ME5001, 2007.

GERSHKOVICH, P. e SHIFFMAN, R. N. An implementation framework for GEM encoded guidelines. **Proc AMIA Symp.**, p. 204–8, 2001.

GONZALEZ-FERRER, A. et al. Use of the virtual medical record data model for communication among components of a distributed decision-support system. In: IEEE-EMBS INTERNATIONAL CONFERENCE ON BIOMEDICAL AND HEALTH INFORMATICS (BHI). **Anais...** [S.l.]: IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6864418>>. Acesso em: 24 out. 2014, 2014.

GONZÁLEZ-FERRER, A. et al. **Data Integration for Clinical Decision Support Based on openEHR Archetypes and HL7 Virtual Medical Record**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. v. 7738

HEALTH LEVEL SEVEN INTERNATIONAL. **About HL7**. Disponível em: <<http://www.hl7.org/about/index.cfm?ref=nav>>. Acesso em: 20 out. 2014.

INSTITUTO HL7 BRASIL. **Institucional: HL7 - Health Level Seven**. Disponível em: <<http://hl7brazil.org/>>. Acesso em: 20 out. 2014.

ISO. **ISO/TR 20.514:2005 Technical Report – Health Informatics – Electronic health record – Definition, scope and context**. . [S.l.: s.n.]. Disponível em: <[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=39525](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39525)>, 2005.

ISO. **ISO 13.606-1:2008 – Health Informatics – Electronic health record communication – Part 1: Reference model**. . [S.l.: s.n.]. Disponível em: <[http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=40784](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40784)>, 2008.

KUPERMAN, G.;; GARDNER, R. e PRYOR, T. A. **HELP: A Dynamic Hospital Information System**. New York: [s.n.], 1991. p. 334

LEZCANO, L.;; SICILIA, M.-A. e RODRÍGUEZ-SOLANO, C. Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules. **Journal of biomedical informatics**, v. 44, n. 2, p. 343–53, doi:10.1016/j.jbi.2010.11.005, 2011.

MARCOS, M. e MARTÍNEZ-SALVADOR, B. Towards the Interoperability of Computerised Guidelines and Electronic Health Records: An experiment with openEHR archetypes and a chronic heart failure guideline. **Knowledge Representation for Health-Care**, v. 6512, p. 101–113, 2011.

MARTÍNEZ-COSTA, C.;; MENÁRGUEZ-TORTOSA, M. e FERNÁNDEZ-BREIS, J. T. An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. **Journal of biomedical informatics**, v. 43, n. 5, p. 736–46, doi:10.1016/j.jbi.2010.05.013, 2010.

MCDONALD, C. J. et al. LOINC, a universal standard for identifying laboratory observations: a 5-year update. **Clinical chemistry**, v. 49, n. 4, p. 624–33, 2003.

MENÁRGUEZ-TORTOSA, M.;; MARTÍNEZ-COSTA, C. e FERNÁNDEZ-BREIS, J. T. A generative tool for building health applications driven by ISO 13606 archetypes. **Journal of medical systems**, v. 36, n. 5, p. 3063–75, doi:10.1007/s10916-011-9783-1, 2012.

MINISTÉRIO DA SAÚDE/DATASUS. **Departamento de Informática do SUS: Hiperdia – Sistema de Cadastramento e Acompanhamento de Hipertensos e Diabéticos**. . [S.l.]: Ministério da Saúde/DATASUS. Disponível em: <<http://portal.saude.gov.br/portal/se/datasus/area.c>>, 2008.

NATIONAL KIDNEY FOUNDATION. K/DOQI clinical practice guidelines for chronic kidney disease: evaluation, classification, and stratification. **American journal of kidney diseases : the official journal of the National Kidney Foundation**, v. 39, n. 2 Suppl 1, p. S1–266, 2002.

NEHTA. **eHealth Interoperability Framework. Versão 1.1**. Disponível em: <<http://www.nehta.gov.au/implementation-resources/ehealth-foundations/EP-1020-2012/NEHTA-1024-2012>>. Acesso em: 20 out. 2014.

OPENEHR. **openEHR Java Reference Implementation**. . [S.l.]: openEHR. Disponível em: <<http://www.openehr.org/wiki/display/projects/Java+Project+Download>>, 2009.

ORACLE. **Java EE at a Glance**. Disponível em: <<http://www.oracle.com/technetwork/java/javasee/overview/index.html>>. Acesso em: 20 out. 2014.

PĂUN, I. D. et al. Local EHR management based on openEHR and EN13606. **Journal of medical systems**, v. 35, n. 4, p. 585–90, doi:10.1007/s10916-009-9395-1, 2011.

PELEG, M. et al. The InterMed approach to sharable computer-interpretable guidelines: a review. **Journal of the American Medical Informatics Association : JAMIA**, v. 11, n. 1, p. 1–10, doi:10.1197/jamia.M1399, 2004.

PORTELA, M. C. Diretrizes Clínicas como Instrumento de Melhoria da Qualidade da Assistência Suplementar: o papel da Agência Nacional de Saúde. **Documentos técnicos de apoio ao Fórum de Saude Suplementar de 2003**. Rio de Janeiro: Brasil. Ministério da Saúde. Agência Nacional de Saude Suplementar, 2004. p. 177–210.

POWER, D. **Decision support systems: concepts and resources for managers**. Westport: Quorum Books, 2002. p. 272

REZENDE, S. **Sistemas Inteligentes: Fundamentos e Aplicações**. Barueri, SP: Manole, 2005. p. 525

RINNER, C. et al. Semantic validation of standard-based electronic health record documents with W3C XML schema. **Methods of information in medicine**, v. 49, n. 3, p. 271–80, doi:10.3414/ME09-02-0027, 2010.

ROMÃO JUNIOR, J. E. Doença Renal Crônica: Definição , Epidemiologia e Classificação. **J. Bras. Nefrol.**, v. 26, n. 3, p. 1–3, 2004.

RONCHI, D. C. M. et al. Desafios no desenvolvimento de prontuários eletrônicos baseados em arquétipos: avaliação fisioterapêutica funcional. **Fisioterapia em Movimento**, v. 25, n. 3, p. 497–506, doi:10.1590/S0103-51502012000300005, 2012.

RŮZICKA, M. e SVÁTEK, V. Mark-up based analysis of narrative guidelines with the Stepper tool. **Studies in health technology and informatics**, v. 101, p. 132–6, 2004.

SANTOS, M. R.;; BAX, M. P. e KALRA, D. Dealing with the archetypes development process for a regional EHR system. **Applied clinical informatics**, v. 3, n. 3, p. 258–75, doi:10.4338/ACI-2011-12-RA-0074, 2012.

SCHLOEFFEL, P.;; BEALE, T. e HAYWORTH, G. The relationship between CEN 13606, HL7, and openEHR. **HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer**, v. 7, p. 1–4, 2006.

SHIFFMAN, R. N. et al. Bridging the guideline implementation gap: a systematic, document-centered approach to guideline implementation. **Journal of the American Medical Informatics Association : JAMIA**, v. 11, n. 5, p. 418–26, doi:10.1197/jamia.M1444, 2004.

SHORTLIFFE, E. H. **Computer-Based Medical Consultations: MYCIN**. New York: Elsevier, 1976.

SHORTLIFFE, E. H. e CIMINO, J. J. **Biomedical Informatics: Computer Applications in Health Care and Biomedicine**. 3rd ed ed. New York: Springer, 2006. p. 1038

SOCIEDADE BRASILEIRA DE NEFROLOGIA. **Censo de Diálise SBN 2013**. Disponível em: <[http://www.sbn.org.br/pdf/censo\\_2013-14-05.pdf](http://www.sbn.org.br/pdf/censo_2013-14-05.pdf)>. Acesso em: 25 jul. 2014.

SPIGOLON, D. N. e MORO, C. M. C. Arquétipos do conjunto de dados essenciais de enfermagem para atendimento de portadoras de endometriose. **Revista Gaúcha de Enfermagem**, v. 33, n. 4, p. 22–32, doi:10.1590/S1983-14472012000400003, 2012.

STEINDORFER, P. Development of clinical practice guidelines (CPGs): An outline exemplified on recent CPG reviews. **European Surgery**, v. 38, n. 1, p. 21–26, doi:10.1007/s10353-005-0205-9, 2006.

SUNDEVALL, E. et al. Applying representational state transfer (REST) architecture to archetype-based electronic health record systems. **BMC medical informatics and decision making**, v. 13, p. 57, doi:10.1186/1472-6947-13-57, 2013.

SVÁTEK, V. e RŮŽIČKA, M. Step-by-step mark-up of medical guideline documents. **International Journal of Medical Informatics**, v. 70, n. 2-3, p. 329–335, doi:10.1016/S1386-5056(03)00041-8, 2003.

THE JBOSS DROOLS TEAM. **Drools Expert User Guide: Version 6.0.0.Beta5.**  
[S.l: s.n.], [S.d.].

VESELÝ, A.;; ZVÁROVÁ, J. e PELEŠKA, J. Computerized presentation of medical guidelines. **Proceedings Medinfo 2004 AMIA**, p. 107, 2004.

## APÊNDICES



**APÊNDICE B – CÓDIGOS HTTP RETORNADOS PELOS SERVIÇOS**

<b>Código</b>	<b>Nome</b>	<b>Descrição</b>
200	OK	Indica que a requisição foi realizada com sucesso.
201	<i>Created</i>	Indica que a requisição foi realizada com sucesso e resultou na criação de um novo recurso.
400	<i>Bad Request</i>	Indica que a requisição não pôde ser entendida, por um problema na sintaxe do conteúdo. A requisição não pode ser repetida sem modificações nos dados enviados.
404	<i>Not Found</i>	Indica que o recurso requisitado não foi encontrado.
406	<i>Not Acceptable</i>	Indica que o recurso indicado pela requisição não é capaz de processar os dados enviados.
409	<i>Conflict</i>	Indica que a requisição não pôde ser completada por um conflito existente com o estado atual do recurso.

## APÊNDICE C – JSON DOS DADOS ESTRUTURADOS EM ARQUÉTIPO

```

{
  "content": [
    {
      "protocol": {
        "rm_type": "ITEM_TREE",
        "items": [
          {
            "rm_type": "ELEMENT",
            "value": {
              "rm_type": "DV_CODED_TEXT",
              "defining_code": {
                "terminology_id": {
                  "rm_type": "TERMINOLOGY_ID",
                  "name": "local",
                  "value": "local"
                },
                "rm_type": "CODE_PHRASE",
                "code_string": "at0017"
              }
            },
            "archetype_node_id": "at0013"
          },
          {
            "rm_type": "CLUSTER",
            "items": [
              {
                "rm_type": "ELEMENT",
                "value": {
                  "rm_type": "DV_CODED_TEXT",
                  "defining_code": {
                    "terminology_id": {
                      "rm_type": "TERMINOLOGY_ID",
                      "name": "local",
                      "value": "local"
                    },
                    "rm_type": "CODE_PHRASE",
                    "code_string": "at0026"
                  }
                },
                "archetype_node_id": "at0014"
              },
              {
                "rm_type": "ELEMENT",
                "archetype_node_id": "at1034"
              }
            ],
            "archetype_node_id": "at1033"
          },
          {
            "rm_type": "ELEMENT",
            "value": {
              "rm_type": "DV_CODED_TEXT",
              "defining_code": {

```

```

        "terminology_id": {
          "rm_type": "TERMINOLOGY_ID",
          "name": "local",
          "value": "local"
        },
        "rm_type": "CODE_PHRASE",
        "code_string": "at1036"
      }
    },
    "archetype_node_id": "at1035"
  },
  {
    "rm_type": "ELEMENT",
    "archetype_node_id": "at1038"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_CODED_TEXT",
      "defining_code": {
        "terminology_id": {
          "rm_type": "TERMINOLOGY_ID",
          "name": "local",
          "value": "local"
        },
        "rm_type": "CODE_PHRASE",
        "code_string": ""
      }
    },
    "archetype_node_id": "at1010"
  },
  {
    "rm_type": "CLUSTER",
    "items": [
      {
        "rm_type": "ELEMENT",
        "archetype_node_id": "at0001"
      },
      {
        "rm_type": "ELEMENT",
        "archetype_node_id": "at0002"
      },
      {
        "rm_type": "ELEMENT",
        "archetype_node_id": "at0003"
      },
      {
        "rm_type": "ELEMENT",
        "value": {
          "rm_type": "DV_COUNT",
          "accuracy_unknown": false
        },
        "archetype_node_id": "at0016"
      },
      {
        "rm_type": "ELEMENT",
        "archetype_node_id": "at0004"
      }
    ]
  }

```

```

    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_DATE_TIME",
        "accuracy_unknown": false
      },
      "archetype_node_id": "at0005"
    },
    {
      "rm_type": "ELEMENT",
      "archetype_node_id": "at0006"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_DATE_TIME",
        "accuracy_unknown": false
      },
      "archetype_node_id": "at0007"
    },
    {
      "rm_type": "CLUSTER",
      "items": [
        {
          "rm_type": "ELEMENT",
          "archetype_node_id": "at0014"
        },
        {
          "rm_type": "ELEMENT",
          "archetype_node_id": "at0015"
        }
      ],
      "archetype_node_id": "at0013"
    },
    {
      "rm_type": "ELEMENT",
      "archetype_node_id": "at0008"
    }
  ],
  "archetype_node_id": "at0000",
  "archetype_details": {
    "rm_type": "ARCHETYPED",
    "rm_version": "v1",
    "archetype_id": {
      "qualified_rm_entity": "openEHR-EHR-CLUSTER",
      "rm_type": "ARCHETYPE_ID",
      "rm_name": "EHR",
      "domain_concept": "device",
      "value": "openEHR-EHR-CLUSTER.device.v1",
      "rm_entity": "CLUSTER",
      "rm_originator": "openEHR",
      "version_id": "1"
    }
  },
  "hierarchy_node_id": "at1025"
}

```

```

    ],
    "archetype_node_id": "at0011"
  },
  "rm_type": "OBSERVATION",
  "data": {
    "rm_type": "HISTORY",
    "events": [
      {
        "time": {
          "rm_type": "DV_DATE_TIME",
          "accuracy_unknown": false,
          "value": "20141023T070745",
          "magnitude": 1414098465000
        },
        "rm_type": "POINT_EVENT",
        "state": {
          "rm_type": "ITEM_TREE",
          "items": [
            {
              "rm_type": "ELEMENT",
              "value": {
                "rm_type": "DV_CODED_TEXT",
                "defining_code": {
                  "terminology_id": {
                    "rm_type": "TERMINOLOGY_ID",
                    "name": "local",
                    "value": "local"
                  },
                  "rm_type": "CODE_PHRASE",
                  "code_string": "at1001"
                }
              }
            },
            {
              "archetype_node_id": "at0008"
            }
          ]
        },
        "rm_type": "ELEMENT",
        "archetype_node_id": "at1052"
      },
      {
        "rm_type": "CLUSTER",
        "items": [
          {
            "rm_type": "CLUSTER",
            "items": [
              {
                "rm_type": "ELEMENT",
                "archetype_node_id": "at0016"
              },
              {
                "rm_type": "ELEMENT",
                "value": {
                  "rm_type": "DV_CODED_TEXT",
                  "defining_code": {
                    "terminology_id": {
                      "rm_type": "TERMINOLOGY_ID",
                      "name": "local",
                      "value": "local"
                    }
                  }
                }
              }
            ]
          }
        ]
      }
    ]
  }

```

```

    },
    "rm_type": "CODE_PHRASE",
    "code_string": ""
  }
},
"archetype_node_id": "at0011"
},
{
  "rm_type": "ELEMENT",
  "value": {
    "rm_type": "DV_QUANTITY",
    "accuracy_unknown": false,
    "units": "J/min"
  },
  "archetype_node_id": "at0005"
}
],
"archetype_node_id": "at0010"
},
{
  "rm_type": "ELEMENT",
  "value": {
    "rm_type": "DV_CODED_TEXT",
    "defining_code": {
      "terminology_id": {
        "rm_type": "TERMINOLOGY_ID",
        "name": "local",
        "value": "local"
      },
      "rm_type": "CODE_PHRASE",
      "code_string": ""
    }
  },
  "archetype_node_id": "at0009"
}
],
"archetype_node_id": "at0000",
"archetype_details": {
  "rm_type": "ARCHETYPED",
  "rm_version": "v1",
  "archetype_id": {
    "qualified_rm_entity": "openEHR-EHR-CLUSTER",
    "rm_type": "ARCHETYPE_ID",
    "rm_name": "EHR",
    "domain_concept": "level_of_exertion",
    "value": "openEHR-EHR-CLUSTER.level_of_exertion.v1",
    "rm_entity": "CLUSTER",
    "rm_originator": "openEHR",
    "version_id": "1"
  }
},
"architecture_node_id": "at1030"
},
{
  "rm_type": "ELEMENT",
  "value": {
    "rm_type": "DV_CODED_TEXT",

```

```

      "defining_code": {
        "terminology_id": {
          "rm_type": "TERMINOLOGY_ID",
          "name": "local",
          "value": "local"
        },
        "rm_type": "CODE_PHRASE",
        "code_string": "at1044"
      }
    },
    "archetype_node_id": "at1043"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "precision": 0,
      "rm_type": "DV_QUANTITY",
      "accuracy_unknown": false,
      "magnitude": 45,
      "units": "°"
    },
    "archetype_node_id": "at1005"
  }
],
"archetype_node_id": "at0007"
},
"data": {
  "rm_type": "ITEM_TREE",
  "items": [
    {
      "rm_type": "ELEMENT",
      "value": {
        "precision": 0,
        "rm_type": "DV_QUANTITY",
        "accuracy_unknown": false,
        "magnitude": 156,
        "units": "mm[Hg]"
      },
      "archetype_node_id": "at0004"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "precision": 0,
        "rm_type": "DV_QUANTITY",
        "accuracy_unknown": false,
        "magnitude": 97,
        "units": "mm[Hg]"
      },
      "archetype_node_id": "at0005"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "precision": 0,
        "rm_type": "DV_QUANTITY",
        "accuracy_unknown": false,

```

```

        "units": "mm[Hg]"
      },
      "archetype_node_id": "at1006"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "precision": 0,
        "rm_type": "DV_QUANTITY",
        "accuracy_unknown": false,
        "units": "mm[Hg]"
      },
      "archetype_node_id": "at1007"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_TEXT",
        "value": "Paciente com sinais de nervosismo"
      },
      "archetype_node_id": "at0033"
    }
  ],
  "archetype_node_id": "at0003"
}
]
},
"archetype_details": {
  "rm_type": "ARCHETYPED",
  "rm_version": "v1",
  "archetype_id": {
    "qualified_rm_entity": "openEHR-EHR-OBSERVATION",
    "rm_type": "ARCHETYPE_ID",
    "rm_name": "EHR",
    "domain_concept": "blood_pressure",
    "value": "openEHR-EHR-OBSERVATION.blood_pressure.v1",
    "rm_entity": "OBSERVATION",
    "rm_originator": "openEHR",
    "version_id": "1"
  }
}
},
{
  "rm_type": "OBSERVATION",
  "data": {
    "rm_type": "HISTORY",
    "events": [
      {
        "time": {
          "rm_type": "DV_DATE_TIME",
          "accuracy_unknown": false,
          "value": "20141023T070745",
          "magnitude": 1414098465000
        },
        "rm_type": "POINT_EVENT",
        "data": {

```

```

"rm_type": "ITEM_TREE",
"items": [
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0004"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0005"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0006"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0007"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0009"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0010"
  },
  {
    "rm_type": "ELEMENT",
    "value": {
      "rm_type": "DV_BOOLEAN",
      "value": false
    },
    "archetype_node_id": "at0011"
  }
]

```

```

    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0008"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0012"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0013"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0014"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0015"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0016"
    },
    {
      "rm_type": "ELEMENT",
      "value": {
        "rm_type": "DV_BOOLEAN",
        "value": false
      },
      "archetype_node_id": "at0017"
    },
  },

```

```

        {
            "rm_type": "ELEMENT",
            "value": {
                "rm_type": "DV_BOOLEAN",
                "value": false
            },
            "archetype_node_id": "at0018"
        }
    ],
    "archetype_node_id": "at0003"
}
]
},
"archetype_details": {
    "rm_type": "ARCHETYPED",
    "rm_version": "v2",
    "archetype_id": {
        "qualified_rm_entity": "openEHR-EHR-OBSERVATION",
        "rm_type": "ARCHETYPE_ID",
        "rm_name": "EHR",
        "domain_concept": "res_drc_prevrim_a",
        "value": "openEHR-EHR-OBSERVATION.res_drc_prevrim_a.v2",
        "rm_entity": "OBSERVATION",
        "rm_originator": "openEHR",
        "version_id": "2"
    }
}
}
],
"rm_type": "COMPOSITION",
"context": {
    "rm_type": "EVENT_CONTEXT",
    "location": "Hospital Municipal",
    "end_time": {
        "rm_type": "DV_DATE_TIME",
        "accuracy_unknown": false,
        "value": "20141023T071008",
        "magnitude": 1414098608062
    },
    "start_time": {
        "rm_type": "DV_DATE_TIME",
        "accuracy_unknown": false,
        "value": "20141023T071008",
        "magnitude": 1414098608062
    }
},
"language": {
    "terminology_id": {
        "rm_type": "TERMINOLOGY_ID",
        "name": "ISO_639-1",
        "value": "ISO_639-1"
    },
    "rm_type": "CODE_PHRASE",
    "code_string": "pt-br"
},
"archetype_details": {

```

```
"rm_type": "ARCHETYPED",  
"rm_version": "1.4",  
"archetype_id": {  
  "qualified_rm_entity": "openEHR-EHR-OBSERVATION",  
  "rm_type": "ARCHETYPE_ID",  
  "rm_name": "EHR",  
  "domain_concept": "blood_pressure",  
  "value": "openEHR-EHR-OBSERVATION.blood_pressure.v1",  
  "rm_entity": "OBSERVATION",  
  "rm_originator": "openEHR",  
  "version_id": "1"  
}  
}  
}
```