

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MAPEAMENTO ONTOLÓGICO COM APLICAÇÃO NO DOMÍNIO BIOMÉDICO

WILLIBY DA SILVA FERREIRA

JOÃO PESSOA – PB
FEVEREIRO 2019



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MAPEAMENTO ONTOLÓGICO COM APLICAÇÃO NO DOMÍNIO BIOMÉDICO

WILLIBY DA SILVA FERREIRA

JOÃO PESSOA – PB
FEVEREIRO 2019

WILLIBY DA SILVA FERREIRA

MAPEAMENTO ONTOLÓGICO COM APLICAÇÃO EM DOMÍNIO BIOMÉDICO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação da Universidade Federal da Paraíba por Williby da Silva Ferreira, sob a orientação do Prof. Dr. Claurton de Albuquerque Siebra, como parte dos requisitos parciais para a obtenção do título de Mestre em Informática.

Linha de Pesquisa: Computação Distribuída

Orientador: Prof. Dr. Claurton de Albuquerque Siebra.

**JOÃO PESSOA – PB
FEVEREIRO 2019**

**Catálogo na publicação Seção de Catalogação e
Classificação**

F383m Ferreira, Williby da Silva.

Mapeamento Ontológico com Aplicação no Domínio
Biomédico / Williby da Silva Ferreira. - João Pessoa,
2019.

98 f. : il.

Orientação: Claurton de Albuquerque Siebra.
Dissertação (Mestrado) - UFPB/CI.

1. Informática em Saúde. 2. Banco de Dados Relacional.
3. Sistemas de Suporte à Decisão Clínica. 4.
Ontologias. I. Siebra, Claurton de Albuquerque. II.
Título.

UFPB/BC

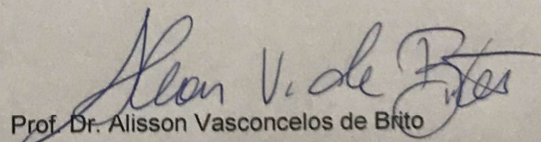


UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

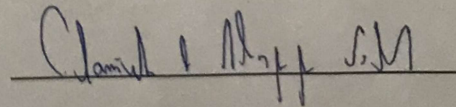


Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Williby da Silva Ferreira, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 10 de junho de 2019.

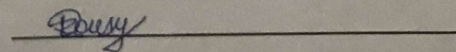
1 Aos dez dias do mês de junho, do ano de dois mil e dezenove, às dez horas, no Centro de
2 Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os membros
3 da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Williby da Silva
4 Ferreira, vinculado a esta Universidade sob a matrícula nº 20171006475, candidato ao grau
5 de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa
6 "Computação Distribuída", do Programa de Pós-Graduação em Informática, da Universidade
7 Federal da Paraíba. A comissão examinadora foi composta pelos professores: Clauriton de
8 Albuquerque Siebra (PPGI-UFPB) Co-Orientador e Presidente da Banca, Danielle Rousy
9 Dias da Silva (UFPB), Examinadora Externa ao Programa, Juliana de Albuquerque
10 Goncalves Saraiva (UFPB), Examinadora Externa ao Programa. Dando início aos trabalhos,
11 o Presidente da Banca cumprimentou os presentes, comunicou aos mesmos a finalidade da
12 reunião e passou a palavra ao candidato para que o mesmo fizesse a exposição oral do
13 trabalho de dissertação intitulado: "Mapeamento Ontológico com Aplicação no Domínio
14 Biomédico". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que
15 emitiu o seguinte parecer: "**aprovado**". Do ocorrido, eu, Alisson Vasconcelos de Brito, Vice-
16 Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai
17 assinada por mim e pelos membros da banca examinadora. João Pessoa, 10 de junho de
18 2019.


Prof. Dr. Alisson Vasconcelos de Brito

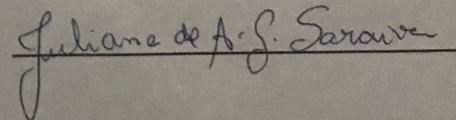
Prof. Dr. Clauriton de Albuquerque Siebra
Co-Orientador (PPGI-UFPB)



Prof. Dr^a. Danielle Rousy Dias Da Silva
Examinadora Externa ao Programa (UFPB)



Prof. Dr^a. Juliana de Albuquerque Goncalves Saraiva
Examinadora Externa ao Programa (UFPB)



*Ainda que a realidade me limite,
a fantasia dos meus sonhos me permite.*

(Roberto Carlos)

AGRADECIMENTOS

Primeiramente, agradeço a Deus e a meus guias de Luz por me concederem sabedoria, força, saúde mental e espiritual durante toda minha vida.

Ao Programa de Pós-Graduação em Informática (PPGI), agradeço a oportunidade oferecida para qualificar minha formação acadêmica.

A meu Orientador, Dr. Claurton Siebra e a Professora Natasha Queiroz, meu especial agradecimento, porque me acolheram com muito profissionalismo, ensinamentos e orientação, norteando minha pesquisa. Muito obrigado!

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo seu apoio financeiro durante o desenvolvimento do meu mestrado.

Aos colegas do Laboratório de Inteligência Artificial Aplicada – LIAA pela troca de conhecimento neste período.

Agradeço as três mulheres da minha vida: minha mãe Ivanalda, minha avó Aldaci e minha tia Valtânia por serem exemplos de inspiração, determinação, coragem e conquista. Sem a luz do presente, a esperança do amanhã e o calor do amor que vocês me transmitem eu não teria condições de trilhar este caminho muitas vezes árduo. Saibam que serei sempre grato por tudo que fazem por mim. Não conseguiria descrever tudo que teria para falar.

Agradeço a meu pai Welliton, a meu avô Luís, a minha tia Cauberta e a meus primos Raialle e Luís Arnaldo, por todas as palavras de carinho e conselhos dados. Além de ouvi-los e compreendê-los pude aplicar os ensinamentos ofertados.

Aos amigos Alysson, Clara, Eduardo, Edvaldo, Gustavo, Joálisson, Joéliton, Lígia, Luciano, Lucíola, Marcos Jr., Rafael, Renan, Romário e Tati, meus agradecimentos. Sem essa família a qual eu pude escolher fora de casa, meus dias não seriam tão alegres e animados. Obrigado por suavizarem minhas preocupações e compreenderem muitas vezes minha ausência durante a realização deste trabalho.

Agradeço as minhas saudosas professoras Juliana Saraiva e Yuska Aguiar por estarem sempre presentes, mesmo distantes fisicamente, contribuindo com meu crescimento profissional e pessoal. Obrigado por todo ensinamento, carinho e atenção.

Enfim, a todas as pessoas que direta ou indiretamente contribuíram para conclusão deste trabalho.

MAPEAMENTO ONTOLÓGICO COM APLICAÇÃO NO DOMÍNIO BIOMÉDICO

Resumo

Introdução: A área da saúde produz diariamente um grande volume de dados que deve ser armazenado de maneira eficiente. É preciosa organizar as informações produzidas para que estas sejam disponibilizadas. Nesse sentido, as tecnologias voltadas para coleta, armazenamento e manipulação de dados vêm evoluindo por meio de técnicas, métodos e ferramentas computacionais automatizadas. Uma das formas mais utilizadas é conhecida como ontologia, a qual permite a representação de um conjunto de conceitos e retrata a semântica das informações. Contudo, o uso completo de ontologias em sistemas computacionais ainda é restrito. Assim, considerando essa restrição e sabendo que os bancos de dados relacionais fornecem, como benefício, escalabilidade para consultas, robustez, desempenho, maturidade, disponibilidade e confiabilidade. Uma alternativa que tem surgido é o mapeamento ontológico. Isto é, desenvolver mecanismos no banco de dados relacionais que executem funções próximas a de ontologias e ainda resguardem os dados. Nesse breve contexto, observou-se a necessidade de desenvolver uma abordagem para mapear uma ontologia do domínio biomédico para um banco de dados relacional de modo que o mesmo auxiliasse o processo decisório no diagnóstico da doença renal crônica. **Objetivos:** O objetivo geral desta proposta é apresentar uma abordagem para mapeamento de ontologia no domínio biomédico para banco de dados relacionais com suporte a decisão clínica e ênfase no processo de diagnóstico das doenças renais crônicas (DRC). **Métodos:** Para desenvolver esse estudo foram definidas etapas metodológicas que envolvem a construção do conhecimento, a modelagem dos dados, a execução do mapeamento e a validação da técnica desenvolvida. Inicialmente foi realizado o levantamento bibliográfico para refinar o conhecimento necessário e compreender o objeto em estudo. Em seguida foram elaboradas as regras de mapeamento e executado as etapas da modelagem de dados, resultando no mapeamento ontológico. **Resultados:** A principal contribuição apresentada é o DB-Ontology, um banco de dados relacional eficiente para suporte a decisão clínica no processo de diagnóstico da DRC. Essa abordagem permite a persistência de dados no banco e preserva a semântica da ontologia de domínio biomédico. O DB-Ontology é resultado do mapeamento ontológico proposto. **Conclusão:** Com a execução das etapas definidas na metodologia, foi possível mapear as principais classes da OntoDecideDRC para o DB-Ontology. Além disso, foi possível compreender a realidade na Unidade de Saúde da Família (USF), de modo que, tal realidade fosse adaptada a hierarquia da ontologia e consequentemente refletisse no banco de dados relacional que foi desenvolvido. Embora a literatura seja carente de estudos de mapeamento ontológico com uso de *stored procedures*, foi possível desenvolver uma abordagem eficiente e capaz de dar suporte a decisão clínica.

Palavra-chave: Informática em Saúde; Banco de Dados Relacional; Sistemas de Suporte à Decisão Clínica; Ontologias.

ONTOLOGICAL MAPPING TO WITH APPLICATION IN THE BIOMEDICAL FIELD

ABSTRACT

Introduction: The health area produces daily a large volume of data that must be stored efficiently. In order for the information produced to be made available, it must be organized. In this sense, technologies focused on data collection, storage and manipulation have been evolving through automated computational techniques, methods and tools. One of the most used forms is known as ontology, which allows the representation of a set of concepts and portrays the semantics of information. However, the full use of ontologies in computational systems is still restricted. Considering this constraint and based on the fact that relational databases provide several benefits such as scalability for queries, robustness, performance, maturity, availability and reliability; an alternative that has arisen is the ontological mapping. That is, developing mechanisms in the relational database that perform functions close to ontologies and still safeguard the data. In this context, it was observed the need to develop an approach to map an ontology from the biomedical domain to a relational database so that it would assist the decision making process in the diagnosis of chronic kidney diseases (CKD). **Objectives:** The general objective of this proposal is to present an approach for ontology mapping in the biomedical domain for relational databases with clinical decision support and emphasis on the diagnosis process of CKD. **Methods:** In order to develop this study, methodological stages were defined, involving knowledge construction, data modeling, mapping execution and validation of the developed technique. Initially, the bibliographical survey was carried out to refine the necessary knowledge and to understand the object under study. Then the mapping rules were elaborated and the data modeling stages were executed, resulting in the ontological mapping. **Results:** The main contribution presented is DB-Ontology, a relational database to support clinical decision in the diagnosis process of CKD. This approach allows the persistence of data in the database and preserves the semantics of the biomedical domain ontology. DB-Ontology is a result of the proposed ontological mapping. **Conclusion:** With the implementation of the steps defined in the methodology, it was possible to map the main classes of OntoDecideDRC to DB-Ontology. In addition, it was possible to understand the reality in USFs, so that such reality would be adapted to the hierarchy of the ontology and consequently reflect in the relational database that was developed. Although the literature presents lacks regarding ontological mapping studies using stored procedures, it was possible to develop an efficient and appropriate approach to support clinical decision making.

Keyword: Relational Database; Clinical Decision Support Systems; Ontologies, Biomedical.

Sumário

1. INTRODUÇÃO.....	15
1.1. MOTIVAÇÃO.....	16
1.2. OBJETIVOS	18
1.2.1. OBJETIVO GERAL.....	18
1.2.2. OBJETIVOS ESPECÍFICOS	18
1.4. ESTRUTURA DA PROPOSTA	19
2. FUNDAMENTAÇÃO TEÓRICA.....	20
2.1. MODELO DE BANCO DE DADOS	20
2.1.1. BANCO DE DADOS RELACIONAL	21
2.1.1.1 MODELO ENTIDADE-RELACIONAMENTO.....	22
2.1.1.2 FUNÇÕES (STORED PROCEDURES).....	23
2.2. REPRESENTAÇÃO DO CONHECIMENTO	24
2.3. ONTOLOGIA	24
2.3.1. OWL.....	26
2.4. SISTEMA DE SUPORTE A DECISÃO.....	26
2.5. INFORMÁTICA EM SAÚDE.....	28
2.6. NEFROLOGIA	28
2.6.1 GRUPO DE RISCO	29
2.6.2 DIAGNÓSTICO DA DRC	30
2.6.3 CLASSIFICAÇÃO DA DRC	31
2.7. ONTODECIDE DRC.....	31
2.8. CONCLUSÃO	34
3. METODOLOGIA	36
4. RESULTADOS E DISCUSSÃO	39
4.1 REGRAS DE MAPEAMENTO	39
4.2 MODELO CONCEITUAL	41
4.3 MODELO LÓGICO	42
4.4 DESENVOLVIMENTO DO PROCESSO DE RACIOCÍNIO	43
4.5 MODELO FÍSICO	46
4.6 FUNÇÕES NO POSTGRESQL	52
4.7 VERIFICAÇÃO DAS FUNCIONALIDADES DO DB-ONTOLOGY.....	62
4.8 AVALIAÇÃO DO DB-ONTOLOGY	64
4.9 VALIDAÇÃO DA ABORDAGEM	71

5. TRABALHOS RELACIONADOS.....	75
5.1 MAPEAMENTO BANCO DE DADOS - ONTOLOGIA	75
5.2 MAPEAMENTO ONTOLOGIA – BANCO DE DADOS.....	79
5.3 CONCLUSÃO.....	83
6. CONSIDERAÇÕES FINAIS.....	86
5.1 LIMITAÇÕES DA PESQUISA	87
5.2 TRABALHOS FUTUROS	88
5.3 PUBLICAÇÕES.....	88
REFERÊNCIAS.....	89
APÊNDICE A	

LISTA DE FIGURA

Figura 1 – Banco de Dados Relacional	22
Figura 2 – Banco de Dados Relacional com uso de Funções.....	23
Figura 3 – Prognóstico por TFG e Classificação Albuminúria	31
Figura 4 – Hierarquia Da Classe Classificação DRC.....	32
Figura 5 – Lista de Propriedades OntoDecideDRC.....	34
Figura 6 – Etapas Metodológicas.....	36
Figura 7 – Etapas Modelagem de Dados.....	37
Figura 8 – Mapeamento Classe Exame para Tabela Exame.....	39
Figura 9 – Mapeamento de SubClasses para Banco de Dados Relacional.....	39
Figura 10 – Modelo Conceitual.....	42
Figura 11 – Modelo Lógico	43
Figura 12 – Teste de Consulta-01	63
Figura 13 – Teste de Consulta-02.....	63
Figura 14 – Teste de Consulta-03.....	63
Figura 15 – Teste de Consulta-04.....	63
Figura 16 – Consulta na Função Calcular Estágio TFG-01	64
Figura 17 – Consulta na Função Calcular Estágio RAC-01	65
Figura 18 – Consulta na Função Calcular Grau de Risco-01	65
Figura 19 – Consulta na Função Calcular Decisão-01	66
Figura 20 – Consulta Histórico Paciente-01 após Função F_Ontology	66
Figura 21 – Consulta na Função Calcular Estágio TFG-02	67
Figura 22 – Consulta na Função Calcular Estágio RAC-02.....	67
Figura 23 – Consulta na Função Calcular Grau de Risco-02	68
Figura 24 – Consulta na Função Calcular Decisão-02	68
Figura 25 – Consulta Histórico Paciente-02 após Função F_Ontology	69
Figura 26 – Consulta na Função Calcular Estágio TFG-03	69
Figura 27 – Consulta na Função Calcular Estágio RAC-03.....	69
Figura 28 – Consulta na Função Calcular Grau de Risco-03	70
Figura 29 – Consulta na Função Calcular Decisão-03	70
Figura 30 – Consulta Histórico Paciente-03 após Função F_Ontology	70
Figura 31 – Consulta Grau de Risco Cenário 01	72
Figura 32 – Consulta Estadiamento Cenário 01	72
Figura 33 – Consulta Diagnóstico e Grau de Risco Cenário 02.....	73
Figura 34 – Consulta na Ontologia Referente Cenário 03	73
Figura 35 – Consulta Classificação DRC Cenário 03	74
Figura 36 – Esquema de Mapeamento Ontologia/Banco de Dados Relacional.....	77
Figura 37 – Mapeamento R2O	78
Figura 38 – Etapas de Mapeamento D2R MAP	79
Figura 39 – Mapeamento OWLMap	80
Figura 40 – Abordagem com Uso de Regras para Mapeamento.....	81
Figura 41 – Visão Geral das Abordagens Identificadas.....	84

LISTA DE TABELA

Tabela 1 – Critérios para DRC.....	30
Tabela 2 – Questões de Competência.....	45
Tabela 3 – Comparação Abordagens Banco de Dados e Ontologia.....	81

LISTA DE ABREVIATURAS E SIGLAS

BD - Banco de Dados;
BDR - Banco de Dados Relacional;
DRC - Doença Renal Crônica;
IA - Inteligência Artificial;
OWL - Ontology Web Language;
OO - Orientado à Objetos;
RAC – Relação Albuminúria Creatinúria
SGBD - Sistema de Gerenciamento de Banco de Dados;
SQL - Structured Query Language;
TFG - Taxa Filtração Glomerular;
USF – Unidade de Saúde da Família
URI - Identificador de Recursos Universal;
XML - Extensible Markup Language.

1. INTRODUÇÃO

Os avanços tecnológicos nos últimos anos permitiram a população mundial maior acesso às informações, tornando-se uma ferramenta indispensável na sociedade atual. Além disso, a tecnologia revolucionou a maneira de como os negócios são gerenciados, como as pessoas se relacionam e como o conhecimento é disseminado (MAEYAMA, 2015). Consequentemente, é gerada uma base de dados com crescimento exponencial e rápida atualização nas diferentes áreas de interesse e conhecimento. Para lidar com esse volume de dados é necessário um processo organizacional que utilize técnicas computacionais para extração da informação realmente importante (VIANNA, 2015).

Neste cenário, a Ciência da Computação tem desenvolvido soluções para as mais diversas áreas, como por exemplo, na área biomédica, onde surge a informática médica através da fusão da computação, tecnologia da informação e históricos médicos associados à saúde e serviços (SILACHAN, 2011). Assim, torna-se possível lidar com o armazenamento e utilização das informações, dados e conhecimentos para resolução de problemas e tomada de decisões (MOTA, 2013).

Segundo Frade (2013), a área da saúde passa por constantes mudanças, dificultando a estabilidade e padronização dos dados. Esses dados são geralmente armazenados em banco de dados relacionais, uma vez que são considerados cientificamente escaláveis, possuem armazenamento eficiente e realizam consultas otimizadas e confiáveis (SAHOO, 2009).

Como os conceitos de domínio biomédico são alimentados continuamente nos modelos de banco de dados, existe uma alta complexidade para alterações nos dados e dificuldades na definição das entidades (FRADE, 2013). Nesse sentido, as tecnologias voltadas para persistência de dados vêm evoluindo por meio de novas técnicas e métodos desenvolvidos. Uma das formas mais utilizadas no domínio biomédico são as ontologias, pois permitem otimizar a representação e organização do conhecimento (HARTUNG 2013). Contudo, a utilização de ontologias em sistemas ainda não é completa devido à falta de ferramenta que, a partir da estrutura definida em uma ontologia, seja capaz de aceitá-la em um banco de dados de modo distribuído.

Considerando o armazenamento em modelo relacional, quando a ontologia é mapeada para um banco de dados relacional, torna-se possível interoperar dados relacionais existentes (HUMAIRA, 2015). O banco de dados relacional também fornece escalabilidade para consultas e raciocínio sobre o conhecimento fornecido pela ontologia.

Assim, uma alternativa encontrada na informática médica para sanar a carência por armazenamento e estabelecer mecanismos de raciocínio, é a utilização de técnicas de mapeamento entre ontologias e banco de dados relacionais (FRADE, 2013). Isto é, definir uma mesma estrutura hierárquica a partir de outra estrutura configurada para fins semelhantes com intermédio de um mecanismo adicional.

1.1. MOTIVAÇÃO

A maioria dos países enfrenta muitos problemas relacionados à saúde, tais como cuidados de saúde caros e a baixa qualidade dos serviços de saúde prestados. Esses problemas começaram a aparecer como resultado do aumento da população e da falta de profissionais de saúde, que se agrava devido ao número crescente de idosos, mais doenças crônicas e a crescente demanda por novos tratamentos de saúde e tecnologias em todo o mundo (MORAES, 2013). A Doença Renal Crônica por exemplo, é uma doença com uma prevalência global média de 12% a 15%, fortemente associada a fatores como custos excessivos de saúde e alta carga de medicação.

Segundo Gomes (2018), estudos tem evidenciado, em todo o mundo, a prevalência de DRC de cerca de 10% para adultos acima de 30 anos e de 28% a 46% em pessoas acima de 64 anos. No Brasil, estima-se que 20 milhões de brasileiros são portadores de DRC em algum grau e 70% desconhecem o fato. Este desconhecimento é justificado pelo subdiagnóstico, mas, principalmente ao fato de que a DRC é uma doença silenciosa que apresenta pouco ou nenhum sinal e/ou sintoma, ao longo de sua progressão (SBN, 2015). Consequentemente, têm se ampliado o número de pesquisas para tentar minimizar os impactos da doença e ampliar os sistemas de saúde, considerando os aspectos geográficos, culturais e hereditários.

O atual modelo de sistema de saúde, embora seja distribuído (teoricamente), funciona (na prática) de maneira centralizado. Tal afirmação é perceptível pela ineficiência da comunicação existente entre os próprios profissionais. No caso da DRC, por exemplo, muitos pacientes com a doença deixam de ser tratados na própria USF (Unidade de Saúde da Família) ou chegam ao médico especialista em nefrologia já em estágio avançado, devido a falha de comunicação e conhecimento necessário no processo decisório (GOMES, 2018).

É preciso efetivar um modelo distribuído operante, a fim de que, os vários profissionais de saúde troquem informações e produzam respostas mais rápidas, eficientes e seguras, possibilitando aos pacientes melhor gerenciar sua própria saúde. O diagnóstico

médico, por exemplo, necessita de uma grande quantidade de dados heterogêneos, como vídeos, imagens e texto (MARAGOUidakis, 2008). E pesquisadores podem utilizar diferentes nomenclaturas para referenciar um mesmo termo médico (SARKAR, 2010).

As tecnologias voltadas para o armazenamento dos dados têm se utilizado de técnicas computacionais automatizadas como as ontologias, que retratam a semântica das informações, de forma que as mesmas possam ser compreendidas pelos sistemas distribuídos. As ontologias são capazes de apresentar melhorias na recuperação da informação ao organizar o conteúdo de fontes de dados que compõem um domínio. Nesse sentido, conhecendo a sintaxe e a semântica dos dados armazenados e padronizando as definições dos dados é possível extrair as informações. Além disso, as ontologias de domínio podem apoiar a recuperação de dados oriundos de banco de dados relacionais para dar suporte a decisões.

Embora haja padrões para representar ontologias, como a linguagem OWL, o uso completo de ontologias em sistemas computacionais ainda é restrito. Além disso, as consultas em ontologias não é tão desenvolvida como nos bancos de dados relacionais. Assim, os bancos de dados relacionais fornecem benefícios como robustez, desempenho, maturidade, disponibilidade e confiabilidade (HUMAIRA, 2015). Neles, o conteúdo é armazenado em uma estrutura de entidades e relacionamentos a partir de uma divisão pré-definida, tão como são definidos os tipos específicos para as propriedades dessas entidades.

Segundo Neumann (2015), os bancos de dados relacionais também fornecem recursos para raciocínio. Isto é, mecanismos avançados que permite ao SGBD consultas complexas, como por exemplo, as *stored procedures* – coleção de comandos criados, compilados e executados na base de dados, e que quando invocados em aplicações exigem apenas parâmetros simples. Partindo deste pressuposto, observa-se a potencialidade dos procedimentos armazenados para representar no banco de dados relacional o processo de raciocínio contido nos axiomas das ontologias, permitindo obter resultados relevantes por meio de regras de inferência.

Norteados pelos aspectos motivacionais, pretende-se investigar mecanismos na Ciência da Computação, especificamente na área de Banco de Dados e Inteligência Artificial para solucionar o problema pertinente a esta pesquisa que está focado no aspecto crítico das operações de ontologias no domínio biomédico, em que a persistência de dados é um dos elementos de impacto e o processo de mapeamento entre ontologia e banco de dados relacional é um mecanismo minimizador desse impacto.

Portanto, levando em consideração a importância do armazenamento dos dados

da área da saúde em modelos relacionais e a relevância das ontologias no aspecto da representação do conhecimento e raciocínio para a tomada de decisão por médicos da atenção primária, foram identificados como problemas de pesquisa (i) a falha de comunicação entre profissionais da atenção primária e especialistas; (ii) a falta de mecanismos para garantir o mapeamento sem perda de dados, semântica, expressividade e que garanta a persistência dos dados; (iii) a ausência de uma abordagem para mapeamento entre ontologia e banco de dados que seja capaz de dar suporte no processo decisório do diagnóstico da DRC aos médicos da atenção primária na UFS.

1.2. OBJETIVOS

Com o intuito de nortear a pesquisa, foram definidos os objetivos para concepção deste trabalho, esses, retratados nas subseções seguinte (1.2.1 e 1.2.2).

1.2.1. OBJETIVO GERAL

Implementar um procedimento de representação do conhecimento e raciocínio, a partir do mapeamento da ontologia de domínio biomédico para banco de dados relacional, que seja capaz de auxiliar médicos da atenção primária no tocante ao diagnóstico de pacientes com Doença Renal Crônica.

1.2.2. OBJETIVOS ESPECÍFICOS

Além do objetivo anterior e visando refinar os resultados pretendidos neste trabalho, foram definidos também os seguintes objetivos específicos:

1. Realizar levantamento de técnicas e ferramentas utilizadas para mapeamento entre banco de dados relacional e ontologia;
2. Identificar as classes da ontologia de domínio biomédico voltadas ao diagnóstico da Doença Renal Crônica;
3. Identificar as propriedades inerentes às classes da ontologia de domínio biomédico voltadas ao diagnóstico da Doença Renal Crônica;
4. Formular o mapeamento da ontologia para banco de dados relacional;
5. Implementar um Sistema Gerenciador de Banco de Dados Relacional voltado para diagnóstico e estadiamento da Doença Renal Crônica com uso de stored procedures;
6. Validar o banco de dados de modo a garantir a semântica e persistência dos dados

7. Implementar procedimentos (*stored procedures*) no Banco de Dados para representar o conhecimento e raciocínio.

1.3. ESTRUTURA DA DISSERTAÇÃO

A proposta de trabalho contida neste documento apresenta 6 capítulos. O atual Capítulo apresenta a contextualização, motivação e os objetivos deste trabalho. Seguido pelo Capítulo 2, que aborda na literatura os principais conceitos para o embasamento teórico necessário no desenvolvimento desta proposta, bem como, o domínio experimental do estudo. O Capítulo 3 define o processo metodológico utilizado. O Capítulo 4 disserta os resultados e discussão. O Capítulo 5 apresenta trabalhos relacionados e/ou metodologias adaptáveis para o mapeamento ontológico em comparação com os resultados apresentados. Por fim, as considerações finais são apresentadas, seguido das referências utilizadas no trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado uma breve contextualização dos aspectos conceituais e filosóficos dos temas relevantes para uma melhor compreensão das atividades que serão desenvolvidas ao longo do trabalho. A Seção 2.1 abordará o modelo de banco de dados utilizado. A Seção 2.2 disserta sobre a representação do conhecimento. As Seções 2.3 e 2.4 abordam respectivamente sobre ontologia e os sistemas de suporte a decisão. O domínio experimental aborda a Seção 2.5 (Informática Médica); Seção 2.6 (Nefrologia) e Seção 2.7 (OntoDecideDRC).

2.1. MODELO DE ARMAZENAMENTO EM BANCO DE DADOS

Na Ciência da Computação o banco de dados é descrito como um repositório estruturado de dados. Os armazenamentos dos dados em um banco são organizados de maneira que permita agilizar a busca e a recuperação. Há vários tipos de banco de dados, sejam eles relacionais, rede, orientado a objetos ou hierárquicos, mas todos utilizam diferentes modelos para organizar os dados.

Existem diferentes regras para descrever como os dados são armazenados e recuperados. Além disso, para sistemas cada vez mais robustos com base de dados organizadas, é preciso considerar os diferentes modelos de banco de dados para representar esses dados (MANOVICH, 2015). Embora os bancos de dados NoSQL venham evoluindo em termos de uso, a pesquisa teve como foco apenas os bancos de dados relacionais devido à grande popularidade dos mesmos.

De acordo com Irina (2007), três fatores principais devem ser considerados no processo de armazenamento de ontologias em banco de dados relacional: (i) dados legados: permitem que as ontologias interoperem grande fluxo de dados quando armazenados em bancos de dados relacionais existentes; (ii) aplicativos herdados: uma vez armazenado em BDR, é possível acessar as ontologias por aplicativos de banco de dados relacional existentes; (iii) ontologias em larga escala: a capacidade de armazenamento de grande fluxo de dados, justifica que os BDRs também são adequados no processo de armazenamento ontológico de grande escala que podem conter milhões de instâncias. Um pré-requisito para este armazenamento é o mapeamento de ontologias para bancos de dados relacionais.

O mapeamento de ontologias em bancos de dados relacionais deve lidar com os

seguintes problemas:

- Perda de dados: o resultado do mapeamento deve descrever adequadamente os dados originais.
- Perda de estrutura: em alguns casos, o mapeamento resulta em perda mínima, considerando que as construções em ontologias não podem ser mapeadas completamente para um BDR.
- Foco em estrutura: é importante considerar o mapeamento das instancias, assim como o mapeamento de estruturas.
- Foco nos dados: os dados devem ser mapeados, com incorporação de tipos de dados.
- Aplicabilidade: em alguns casos, o mapeamento não é realmente geral no sentido de que sua aplicação é bastante restrita. Por exemplo, quando o mapeamento aborda ontologias que não sejam tangíveis a experimentos, existe problema na aplicabilidade.
- Exatidão: o mapeamento deve ser comprovável.

2.1.1. BANCO DE DADOS RELACIONAIS

O Sistema de Gerenciamento de Banco de dados Relacionais impactou positivamente e diretamente a vida das pessoas (WADE e CHAMBERLIN, 2012). No início da década de 1970, Edgar Frank Codd formulou e propôs o primeiro modelo de banco de dados, fundamentado na lógica de predicados de primeira ordem. Porém, Darwen (2012), elucida a existência de um modelo de Sistema de Gerenciamento de Banco de Dados anterior ao apresentado por Edgar. Tratava-se de um SGBD semelhante, mas limitado e composto por métodos de acesso que podiam ser processados por um par chave-valor ou por ponteiros.

O Sistema de Gerenciamento de Banco de Dados Relacional é um modelo matemático que descreve a estrutura dos dados. A representação dos dados é feita em forma de tuplas na quais contem seus relacionamentos. Assim, a organização de uma base de dados em termos de modelo relacional constitui um banco de dados relacional.

Um modelo de dados relacional é representado basicamente por tabelas, sendo que cada tabela deve conter colunas, permitindo determinar restrição por meio de tupla através de chave primária (SILBERSCHATZ et al., 2006). É possível também relacionar duas tabelas com a utilização de uma chave estrangeira, por meio da interseção da chave primaria na tabela de origem.

	Atributo			
	Identificador	Nome	Telefone	Estado
	001	Paula	2134.9058	MG
Linha {	002	Roberto	3232.7421	PE
	Relação			

Figura 1. Banco de Dados Relacional (Exemplo)

Fonte: O Autor.

A Figura 1 apresenta um exemplo simples da definição de relação, linha/tupla e atributo de um banco de dados relacional, que utiliza linhas, colunas e tabela. A chave primária é representada por linha/tupla da tabela ou relação. Além disso, as tabelas podem se relacionar, através de chaves estrangeiras.

2.1.1.1. MODELO ENTIDADE-RELACIONAMENTO

Segundo Darwen (2012), o MER é também chamado de Modelo conceitual porque permite “representar de uma forma abstrata, independente da implementação computacional, os dados que serão armazenados em banco de dados”. O modelo Entidade-Relacionamento (MER) foi desenvolvido com o objetivo de representar facilmente objetos do mundo real, no universo computacional. Permite mapear os fenômenos de interesse para o modelo de dados relacional, inerentes aos bancos de dados relacionais.

No MER, a realidade passa a ser visualizada sob três conceitos fundamentais: Entidade, Atributo e Relacionamento. Internamente estes conceitos são representados na forma de tabela, composto por linhas e colunas.

- Entidade - objeto do mundo real, que desejamos representar, armazenar e recuperar informações sobre ele. Exemplo: pessoa, carro, companhia e emprego.
- Atributo – propriedade particular que descreve as características de uma entidade ou de um relacionamento. Exemplo: para descrever a entidade carro, podemos usar os atributos marca, tipo e placa.
- Relacionamento - associação entre duas ou mais entidades. No caso de relacionamentos binários, estes podem ter cardinalidade expressa por 1:1, 1:N, N:1 ou N:N, indicando o número de vezes que uma entidade pode participar do relacionamento

2.1.1.2. FUNÇÕES – *STORED PROCEDURES*

São subprogramas conhecidos como funções armazenados (stored procedures) cujos códigos fontes ficam armazenadas no servidor de banco de dados, conforme Figura 2. Uma vez programada e compilada, a função pode ser executada por diferentes usuários e aplicações.

Segundo Li (2007), “A prática da construção de funções, no lado do servidor, com códigos complexos utilizados por várias aplicações, tem por objetivo evitar que esses códigos sejam replicados em cada um dos aplicativos no lado cliente”. Isto é, caracterizado pelos quais o cliente pede um recurso e o servidor responde diretamente ao pedido, utilizando seus próprios recursos.

Para criar as funções podemos utilizar diferentes linguagens, as quais podem ser divididas em três grupos ou categorias: linguagens não procedurais, linguagens procedurais e as que utilizam linguagem externa e mais complexa.

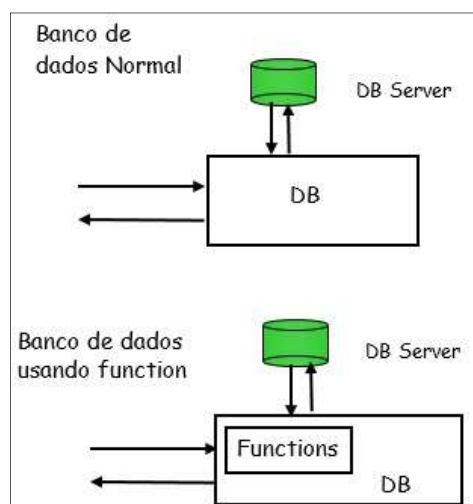


Figura 2. Banco de Dados Relacional com Função

Fonte: Dados da Pesquisa, 2019.

As linguagens procedurais como PL/pgSQL, PL/Java, PL/PHP, PL/Ruby podem utilizar as estruturas comuns de linguagens de programação. Por outro lado, as não procedurais utilizam a linguagem SQL. A seguir detalharemos a linguagem PL/pgSQL que é utilizada na implementação deste trabalho.

PL/pgSQL é uma linguagem procedural, uma extensão da SQL que incorpora comandos de linguagem de programação. Ela incorpora à SQL características procedurais, como controle de decisão (if-then-else) e repetição (for, while) e é estruturada em blocos, nos

quais ficarão todo o código (comandos) referente a definição da função. A estrutura básica de uma função na linguagem PL/pgSQL é apresentada abaixo:

```
CREATE OR REPLACE FUNCTION NOME ( <lista de parametros> )  
[RETURNS <tipo de return>]  
AS  
$$  
DECLARE  
    Declarações; // área de declaração e inicialização de variáveis  
BEGIN  
    comando 1;  
    comando 2;  
    comando n;  
END;  
$$ LANGUAGE 'plpgsql';
```

2.2. REPRESENTAÇÃO DO CONHECIMENTO

O conhecimento pode ser interpretado como a informação baseada em experiências, habilidades e competências de cada pessoa, já a inteligência é adquirida através das transformações das experiências e da aquisição de novos conhecimentos. Na Inteligência Artificial, representa simbolicamente o conhecimento adquirido e manipula-o de maneira automática por programas de raciocínio (RUSSEAL 2009).

A representação do conhecimento configura-se um processo de raciocínio associado ao conhecimento, possibilitando obter conclusões adequadas a partir desta erudição e de regras gerais disponíveis, auxiliando no suporte a decisão (LADEIRA, 1997). Isto é, uma conjunção de estruturas de dados e procedimentos interpretativos que quando utilizando de maneira correta pode auxiliar na tomada de decisões de maneira correta. Assim a representação do conhecimento é determinante em processos eficientes de busca de informações e difusão de conhecimento (HENRIQUE, 2014).

Na Ciência da Computação a representação do conhecimento pode ser feita por uma técnica de referência a um artefato de engenharia, como a ontologia, que é constituída por um vocabulário especificado e utilizada para descrever uma determinada realidade.

2.3. ONTOLOGIA

Surgida na Filosofia, a ontologia em seu campo de origem é a ciência do que é, das estruturas de objetos e dos tipos, eventos, propriedades, processos e relações em todas as áreas da realidade (SMITH, 2003). Hoje é vista por duas perspectivas: a da Filosofia, mencionada e pela perspectiva da Ciência da Computação, inicialmente utilizada pela Inteligência Artificial.

Na perspectiva computacional, percebeu-se a possibilidade de criar um domínio de

conhecimento e estabelecer relações formais entre os itens de conhecimento no domínio, permitindo certos tipos de inferência e raciocínio automatizado, tornando-o compreensível por humanos e computadores.

Segundo Guarino (1998), ontologia é uma especificação de um domínio onde se estabelece um conjunto de propriedades e relações entre os conceitos pertencentes a este domínio. As ontologias têm sido usadas para organizar e definir as informações compartilhadas com uma capacidade notável para especificar uma conceituação compartilhada de forma explícita e formal (GRUBER, 1993).

As ontologias são constituídas essencialmente por um conjunto de conceitos (classes e subclasses) estruturados hierarquicamente, suas propriedades e relacionamentos existentes. Além disso, Russel (2009) enfatiza que as ontologias apresentam axiomas e restrições que disponibilizam informações básicas das quais pode se obter resultados relevantes por meio de regras de inferência. Já a taxonomia refere-se ao modo de classificar e categorizar hierarquicamente um conjunto de conceitos de maneira estruturada, estabelecendo relação de generalização e especialização (DACONTA 2003).

Com base nessas características, a ontologia pode auxiliar na resolução de consultas, na qual o uso de termos é a generalização ou especificação de informações na base de dados. Além disso, as ontologias de domínio foram desenvolvidas para modelar uma estrutura básica de conhecimento para um domínio específico, que combinada com a capacidade de reutilização, permite a construção de novas ontologias (SERGIO, 2016).

Segundo Almeida (2003), as ontologias são instrumentos com alto potencial de organização e representação da informação de fontes de dados em determinado domínio, melhorando os processos de recuperação dos dados. Assim, as ontologias permitem a exploração semântica da informação como um instrumento de representação de relacionamentos conceituais e semânticos (GRUBER, 1993).

O compartilhamento e reuso de ontologias por provedores de informação e desenvolvedores de aplicações é crítico para a interoperabilidade. Se aplicações e fontes de dados distintas usam o mesmo conjunto de termos bem definidos para descrição do domínio, será mais fácil a comunicação entre sistemas informatizados (NOY, 2012). Assim, é importante conhecer os aspectos no desenvolvimento de ontologia, como: o tipo de linguagem utilizada para desenvolvimento e a representatividade do conhecimento.

2.3.1. OWL – *Ontology Web Language*

Através da lógica de descrição é possível formalizar uma ontologia. Segundo McGuinness, (2008), tal formalismo é feito através da aplicação de linguagens que

podemos categorizar em: tradicionais (KIF, Ontolígua), padrão Web (RDF, XML) e baseadas na Web (OIL, OWL). Nesta proposta de trabalho será dado destaque a OWL por ser a principal linguagem de definição e instanciação ontológica, permitindo maximizar a compreensão e interpretação por agentes computacionais.

A OWL (do Inglês *Ontology Web Language*) é considerada uma linguagem para definição de ontologias da Web e é utilizada para o desenvolvimento de aplicações baseadas na Web Semântica (ISOTANI e BITTENCOURT, 2015). Genericamente, a OWL é delineada para utilização em aplicações com processamento de conteúdo de informação (MCGUINNESS, 2008).

A linguagem OWL é apresentada pela W3C¹ em duas gerações (OWL, OWL2) e suas subversões (OWL Lite, OWL DL, OWL Full) e (OWL EL, OWL RL, OWL QL), respectivamente:

- OWL Lite: é adjacente da OWL DL, possuindo como principal característica a simplicidade da sintaxe e hierarquia de classificação e restrição.
- OWL DL: análogo a lógica de descrição, é um subconjunto da OWL Full. Diferencia-se por apresentar ampla expressividade, com completude e decidibilidade computacional.
- OWL Full: é um subconjunto que abrange os diferentes recursos da linguagem OWL, disponibilizando a maior expressividade da linguagem. Além disso, possibilita combinar OWL com RDF sem solicitar as especificações de conceitos.
- OWL EL: derivado da OWL RL, é utilizado para grande volume de classes e propriedades.
- OWL RL: sugerida para lidar com grande quantidade de instâncias de dados em consultas de banco de dados relacional. Sub-linguagem da OWL QL.
- OWL QL: é um subconjunto que detém todos os recursos da OWL2 e utilizado para minimizar efeitos danosos a expressividade da linguagem.

2.4. SISTEMA DE SUPORTE À DECISÃO

Os Sistemas de Apoio à Decisão (SAD) têm como objetivo disponibilizar instrumentos ou subsídios úteis as organizações no processo de tomada de decisão que apresenta deficiências em sua estruturação (HEINZLE, 2010). Isto é, sistemas de informação que oferecem recursos comparativos, analíticos, simulatórios e apoiador com base em um conjunto de cenários gerados que envolvem o domínio de processo decisório.

¹ www.w3.org

Segundo Laudon (2001), os sistemas de suporte à decisão pertencem a uma classe de sistemas que auxiliam indivíduos a tomarem decisões em situações ou problemas semi ou não-estruturados, únicos ou sujeitos a mudanças rápidas. Além disso, ajudam nas etapas de desenvolvimento, comparação e classificação de riscos, fornecendo subsídios para alternativas através de múltiplos cenários de informação (ALVES, 2018).

Os sistemas de apoio à decisão são caracterizados de modo a diferenciar-se dos demais sistemas: "1. Oferece aos usuários flexibilidade, adaptabilidade e respostas rápidas; 2. Permite aos usuários iniciar e controlar as entradas e saídas; 3. Opera com pouca ou nenhuma assistência de programadores profissionais; 4. Oferece suporte para decisões e problemas para os quais as soluções não podem ser especificadas previamente; 5. Usa análises sofisticadas e ferramentas de modelagem" (LAUDON, 2001).

De acordo com Bordoloi (2012), na área da Inteligência Artificial, os sistemas baseados em conhecimento permitem manipular o conhecimento de maneira inteligente, e são utilizados para lidar com problemas que requerem uma quantidade significativa de conhecimento humano e especializado, como por exemplo em cenários de suporte à decisão clínica.

O sistema de suporte a decisão clínica (SSDC) pode ser definido com o uso de sistemas para aquisição de conhecimento que auxiliem profissionais da área da saúde no processo de tomada de decisão (GREENES, 2014). Isto é, devido ao aumento progressivo do uso de tecnologia na área da saúde e a crescente demanda por conhecimento, surgiu a necessidade de desenvolver mecanismos que ajudassem a minimizar dúvidas dos seres humanos.

Segundo Souza (2018), a expectativa é que os SSDCs diminuam a distância entre as evidências e a prática clínica, ao disponibilizar dados relevantes e conhecimento. Embora tenham surgido a muitos anos, a finalidade e a potencialidade dos SSDCs ainda não foram definidas com clareza. Alguns acreditam que um SSDC seja um sistema que se relaciona com uma base de dados, auxiliando o usuário na tomada de decisão, indicando uma boa alternativa (MODY, 2015).

Muitos fatores podem influenciar a implementação do SSDC, como segurança e fluxo de trabalho. Assim, esses tipos de sistemas podem ser caracterizados por auxiliar na resolução de problemas, tomada de decisões e ações por um ser humano; ser flexível as mudanças do processo decisório; e realizar inferências ou regras na seleção de conhecimento e processamento de dados.

2.5 INFORMÁTICA EM SAÚDE

A tecnologia é responsável por intervenções significativas na vida dos seres humanos, incluindo o processamento, armazenamento, e recuperando dos dados em saúde. Assim, entende-se como Informática em Saúde a aplicação da tecnologia da informação aos profissionais de saúde, objetivando a criação de ferramentas e procedimentos que auxiliem estes profissionais no diagnóstico e tratamento de pacientes com mais exatidão e eficiência (UIC, 2011).

Além disso, existem muitos fatores que impulsionam a adoção das tecnologias da informação em saúde, destacando-se entre as mais importantes as necessidades de: diminuir custos; melhorar a segurança do paciente; e padronizar e melhorar a qualidade dos cuidados médicos (SARKAR, 2010). Contudo é preciso investir em gestão da informação para maximizar tomadas de decisões adequadas e seguras.

Baseado na definição de ontologia e seu detalhamento apresentados no Capítulo 2, podemos identificar diversos usos para esta tecnologia semântica na informática em saúde. Portanto, foi proposto a realização de um mapeamento ontológico para estrutura de banco de dados relacional utilizando *stored procedures*, visto que esse tipo de mecanismo permite executar no BDR operações semelhantes aos axiomas das ontologias, além disso, não foi identificado estudos semelhantes nesta temática. Inicialmente é preciso conhecer detalhadamente a hierarquia do domínio experimental, para sequencialmente definir as regras de mapeamento e realizar a modelagem dos dados.

2.6 NEFROLOGIA

A Doença Renal Crônica (DRC) têm se destacado mundialmente devido ao aumento exponencial no número de casos nos últimos anos. Considerada um problema de saúde pública, a DRC vem sendo responsável pelas elevadas taxas de morbidade e levado muitos pacientes a óbito. Nos Estados Unidos da América, por exemplo, cerca de 13% da população adulta apresenta algum grau de perda de função renal (OKPECHI, 2018). Já de acordo com o Censo Brasileiro de Diálise publicado em 2012, o número de pacientes com DRC no Brasil praticamente duplicou na última década, passando de 42 mil em 2000 para mais de 90 mil casos em 2011.

Segundo a Sociedade Brasileira de Nefrologia no Brasil (2014), dentre as definições de DRC publicadas nas novas Diretrizes, destaca-se: anormalidades da estrutura e/ou função do rim, considerando a Taxa de Filtração Glomerular (TFG) em aspectos atípicos de lesão renal e histopatológica presente em período igual ou superior a três meses; e TFG

<60 mL/min/1,73 m² por mesmo período com ou sem lesão renal.

Embora haja uma diversidade de anormalidades estruturais e funcionais dos rins, algumas não possuem consequências clínicas insatisfatórias para a saúde do indivíduo, motivando a mudança na definição da DRC que passou a incrementar o termo “com implicações para a saúde” (KIRSZTAJN, 2014). Assim, a doença renal crônica é definida pela perda da função renal e/ou lesão do parênquima renal presentes por um período igual ou superior a três meses (KDIGO, 2012).

2.6.1 GRUPO DE RISCO

A busca dos indivíduos por cuidado em saúde e a capacidade dos profissionais, nas mais diversas especialidades, é essencial para a detecção da doença e acompanhamento adequado. Porém, alguns indivíduos (grupo de risco) são suscetíveis a desenvolver a DRC, devido a fatores etiológicos e progressiva perda de função renal (BRASIL, 2014). Assim, é relevante reconhecer potenciais indivíduos com risco de desenvolver a doença. O grupo de risco é composto por:

1. Pessoas com Diabetes (tipo 1 ou tipo 2): O diagnóstico do diabetes se baseia no nível de sérico de glicemia de jejum acima de 126 mg/dL, ou acima de 200 mg/dL com 2 horas de intervalo da ingestão de 75g de glicose, ou qualquer valor de hiperglicemia, na presença de sintomas clássicos, como poliúria, polidipsia ou polifagia;
2. Pessoas Hipertensas: Indivíduos com valores da pressão arterial acima de 140/90 mmHg em duas aferições com um intervalo de 7 a 14 dias;
3. Idosos;
4. Indivíduos com histórico de doença cardiovascular: acidente vascular cerebral, doença coronária, doença vascular periférica, insuficiência cardíaca;
5. Hereditariedade familiar de DRC;
6. Portadores de obesidade: IMC >30 kg/m²;
7. Indivíduos consumidores de medicações nefrotóxicas;
8. Tabagismo.

Determinado o diagnóstico da DRC é necessário definir os preditores de progressão, no intuito de identificar o prognóstico para perda de função renal do indivíduo com DRC. Isto é, marcadores para estabelecer o nível de comprometimento: pessoas com níveis (de colesterol, glicemia e pressão) mal controlados, estágios da DRC, assiduidade albuminúrica, uso de agente nefrotóxicos e tabagismo.

2.6.2. DIAGNÓSTICO DA DRC

Para diagnosticar a DRC recomenda-se seguir alguns critérios (Tabela 1). Tais critérios foram refinados ao longo das diferentes Diretrizes publicadas. Além disso, alguns recursos estão disponíveis para maximizar a eficiência do diagnóstico da DRC: a Taxa de Filtração Glomerular (TFG); o Exame Sumário de Urina (EAS) e um exame de imagem, como ultrassonografia dos rins e vias urinárias (BRASIL, 2014).

Tabela 1. Critérios para DRC (Qualquer um dos seguintes presentes por > 3 meses)

Marcadores de Lesão Renal (≥ 1)
Albuminúria (>30 mg/24h; relação albumina/creatinina 30mg/g)
Anormalidades detectadas por exame histológico
Anormalidades estruturais detectadas por exame de imagem
História de transplante renal
TFG diminuída: $< 60\text{ml/min/1,73m}^2$ (categorias de TFG G3a-G5)

Fonte: Kirsztan (2013).

De acordo com Brasil (2014), é necessário avaliar os recursos disponíveis para garantir o diagnóstico da DRC de forma eficiente:

1. Avaliação da Taxa de Filtração Glomerular: utilizar fórmulas baseadas na creatinina sérica para estimar a TFG. Deve evitar a utilização da depuração de creatinina medida através da coleta de urina de 24 horas, devido pontencial risco de erro da coleta e inconvenientes da coleta temporal.
2. Alterações parenquimatosas: devem ser analisadas através do exame sumário de urina (EAS) ou da pesquisa de albuminúria, que identifica albumina na urina. Todos os indivíduos com indícios de DRC devem fazer o EAS. Em caso de pessoas diabéticas e hipertensas com EAS mostrando ausência de proteinúria, está indicada a pesquisa de albuminúria em amostra isolada de urina corrigida pela creatininúria, a Relação Albuminúria Creatininúria (RAC);
3. Avaliação de imagem: realizado para indivíduos com história de DRC familiar, infecção urinária de repetição e doenças urológicas. O exame de imagem recomendado é a ultrassonografia dos rins e vias urinárias.

Em casos de avaliações de indivíduos com risco de DRC e que não apresentaram anormalidades, se recomenda uma reavaliação anual da TFG e EAS em uma unidade básica de saúde.

2.6.3. CLASSIFICAÇÃO DA DRC

Após o diagnóstico é estruturado o tratamento da DRC e estimar o prognóstico. O paciente com a doença é classificado de acordo com os principais desfechos da DRC: doença cardiovascular, evolução para terapia renal substitutiva (TRS) e mortalidade (BRASIL, 2014). O prognóstico (Figura 3) da doença é estabelecido a partir da TFG que têm diferentes níveis e classificação de albuminúria persistente (KDIGO, 2012).

				Estágios de albuminúria persistente (mg/g)		
				A1	A2	A3
				Normal a levemente aumentada	Moderadamente aumentada	Severamente aumentada
				< 30	30-300	> 300
Estágios de TFG (ml/min/1.73 m ²)	1	Normal ou elevada	≥ 90	-	Monitorar	Encaminhar
	2	Redução leve	60-89	-	Monitorar	Encaminhar
	3a	Redução leve a moderada	45-59	Monitorar	Monitorar	Encaminhar
	3b	Redução moderada a grave	30-44	Monitorar	Monitorar	Encaminhar
	4	Redução severa	15-29	Encaminhar	Encaminhar	Encaminhar
	5	Falência renal	< 15	Encaminhar	Encaminhar	Encaminhar

Figura 3. Prognóstico por TFG e Classificação de Albuminúria

Fonte: KDIGO (2012).

A TFG utiliza como referência o limite de 60mL/min/1,73m², o que representa diminuição de cerca de 50% da função renal regular, e consequentemente o aumento significativo do comprometimento da doença renal crônica (KDIGO, 2012).

Brasil (2014) especifica que a classificação do estágio da DRC norteia as tomadas de decisões para garantir o encaminhamento do paciente aos serviços de saúde de referência no tratamento da doença e especialistas. A doença é classificada em: conservador, pré-diálise e TRS. O conservador, compreende os estágios 1 a 3 e é responsável pelo controle dos fatores de risco da progressão da doença; o pré-diálise, representa os pacientes dos estágios 4 a 5-ND (não dialítico) e consiste na manutenção da fase conservadora e preparação do paciente para a TRS em estágios avançados; e a TRS (estágio 5-D) estabelece a substituição da função renal (BRASIL, 2014).

2.7 ONTODECIDEDRC

A OntoDecideDRC é uma ontologia de domínio biomédico desenvolvida para representar os conhecimentos relacionados aos procedimentos necessários para diagnosticar pacientes com a DRC, bem como, definir o estadiamento e o encaminhamento a serviço de saúde especializado. Tavares (2016), delineou três etapas principais para a construção de ontologias: conceitualização, formalização e implementação.

A conceitualização, corresponde ao processo inicial de construção da ontologia OntoDecideDRC e se baseia na elaboração do Léxico Ampliado da Linguagem (LAL). Esta linguagem permite mapear o vocabulário corrente em uma representação simples, e compartilhar o conhecimento das informações do domínio. Além disso, o LAL auxilia no levantamento, modelagem e validação dos conceitos básicos, funções, qualidades e relacionamentos das entidades importantes. Em seguida, na formalização, foi realizado o processo de mapeamento dos termos como, classes, propriedades, hierarquias de conceitos, restrições e axiomas. Por fim, a implementação da ontologia foi desenvolvida em linguagem expressiva e passível de raciocínio automatizado, OWL (OWL-DL); utilizando a ferramenta Protégé como editor (TAVARES, 2016).

Após a conclusão das etapas a OntoDecideDRC foi efetivamente construída, objetivando também, auxiliar médicos da atenção primária e especialistas na tomada de decisões, disponibilizando informações sobre: os procedimentos de diagnóstico e estadiamento da DRC; definir o devido momento para encaminhamento do paciente ao nefrologista; e disponibilizar técnicas de manejo clínico básico relacionados a doença renal crônica (TAVARES, 2016).

A ontologia proposta por Tavares (2016) apresenta 184 classes, bem como, anotações, dados e propriedades de objeto. A métrica da ferramenta Protégé estabelece que a ontologia possui 1240 axiomas, sendo 479 axiomas lógicos. Em relação a expressividade da lógica de descrição, o Protégé classificou a OntoDecideDRC como “SHIF(D)”, devido suas características como: interseções de conceitos, classes disjuntas, restrições universais, propriedades: inversas, funcionais e de dados; entre outras.

A OntoDecideDRC é composta por 12 classes principais e suas subclasses, onde as hierarquias das classes e relacionamentos foram definidas por regras de decisão.

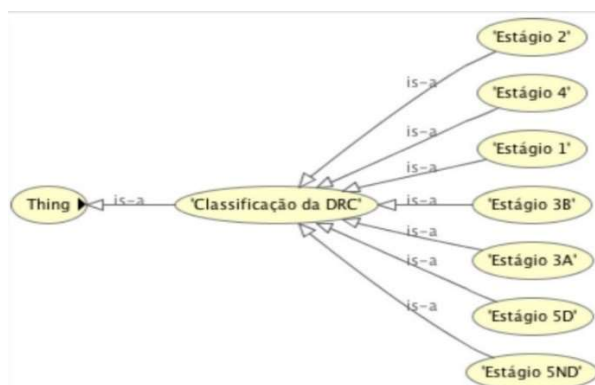


Figura 4. Hierarquia Da Classe Classificação DRC

Fonte: Tavares (2016).

A classe Paciente, é composta por simples informações (pressão arterial, peso

corporal, gênero, idade, etc) necessárias para auxiliar médicos na tomada de decisões. A classe DRC, descreve os conceitos básicos da doença, informando as condições para diagnosticar a doença renal crônica e está relacionada com as classes Fatores de Risco, Médicos de Atenção Primária, e Nefrologista. A classe Encaminhamento, especifica as condições e situações para determinar o encaminhamento ao especialista, no intuito de analisar a progressão da doença. A classe Médico, indica os tipos de médicos especialistas para cada estágio da DRC. A classe Fórmula, informa como calcular a taxa de filtração glomerular. A classe Classificação da DRC (Figura 4), representa os estágios relacionados a doença. A classe Estratégias, define as estratégias para tratamento e prevenção da DRC e se relacionada com as subclasses Manejo Clínico Básico e Terapia de Substituição Renal. Já a classe Exames, abrange os exames utilizados para tomada de decisões relacionadas ao diagnóstico, estadiamento encaminhamento de pacientes com a doença. A classe Grau de Risco da DRC, apresenta a probabilidade de risco do paciente com DRC. A classe Fatores de Risco, indica elementos de risco relacionados a doença. A classe Vacinação, informa as vacinas necessárias para o paciente manter seu quadro de vacinação atualizado. A classe Medicamentos, restrições medicamentosas durante a DRC.

Além das classes, foram definidas as propriedades presentes na ontologia, representando sua estrutura interna. As propriedades são responsáveis pelo refinamento das informações através de ligações com as classes, de modo que, foram planejadas para garantir o conhecimento necessário que auxilie os médicos nas tomadas de decisões e resolução de problemas. Assim, para conectar conceitos e retratar ligações do mundo real, foram definidas 26 propriedades do objeto (Figura 5) para atuar entre as classes.

Para assegurar a representação dos valores numéricos das classes foram criadas 3 propriedades de dados (Figura 5 – coluna à direita superior). Já para representar as informações detalhadas sobre as classes foram definidas 5 propriedades de anotações (Figura 5 – coluna à direita inferior).

Segundo Tavares (2016), além das classes e propriedades, foram descritos os mecanismos para o raciocínio semântico sobre a base de conhecimento. Para tal, foi criado regras para o raciocínio semânticas baseadas em conceitos ontológicos que permitem realizar inferências, no intuito de fornecer suporte à decisão aos médicos da atenção primária e aos especialistas. Após descrever por completo as relações lógicas existentes no domínio, o método foi validado.

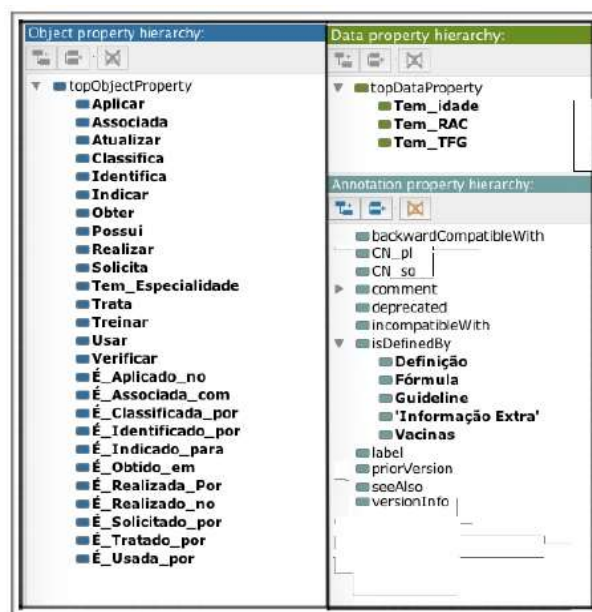


Figura 5. Lista de Propriedades OntoDecideDRC

Fonte: Tavares (2016).

A OntoDecideDRC foi validada com base em critérios de avaliação, como: verificação da estrutura, sintaxe e conteúdo representado pelas definições. Foram definidos procedimentos de avaliação da hierarquia de classes e componentes da ontologia, onde a avaliação das regras e axiomas definidos na ontologia foram feitas através da utilização de questões de competência e de experimento com casos clínicos. Por fim, foi desenvolvido um protótipo de uma *web service* para coletar, raciocinar e fornecer informações em torno da DRC. Assim, a ontologia permite de forma satisfatória auxiliar médicos no processo de diagnóstico, estadiamento e encaminhamento da DRC (TAVARES, 2016).

2.8 CONCLUSÃO

Neste capítulo, foram apresentados os principais conceitos para definição dos passos metodológicos aplicados a este estudo. Foram abordados os modelos de banco de dados relacional e NoSQL, destacando a importante compreensão em torno do tema; bem como a estrutura da ontologia e as características de sistemas de suporte a decisão. Embora se tenha apresentado os vários modelos de armazenamento de dados, é importante ressaltar que este estudo utilizou especificamente o modelo relacional, pois o grande volume atual de dados ainda é armazenado em modelo relacional (SAHOO, 2009). Além disso, quando a ontologia é documentada no banco de dados relacional torna-se possível interoperar dados relacionais existentes.

No cenário dos bancos de dados relacionais, é possível identificar diversas sugestões

de pesquisas realizadas para estabelecer a colaboração mútua entre ontologias e banco de dados, no intuito de interpretar o conhecimento contido nos dados. Alguns exemplos: construção de ontologias a partir de banco de dados; sistemas de acesso a dados com apoio ontológico; estudo de ontologias a partir de dados em SGBD; e mapeamento de ontologia para banco de dados relacionais, sendo esta última o objeto deste estudo. Esta junção dos bancos de dados relacionais e ontologias fornece escalabilidade para consultas e raciocínio sobre o conhecimento fornecido pela própria estrutura ontológica.

No caso do presente trabalho, a união destes dois conceitos será feita por meio de funções armazenadas (*stored procedures*), as quais serão responsáveis por um mapeamento ontológico eficiente para o banco de dados relacional, permitindo que os dados sejam armazenados, persistidos e semanticamente preservados. Ou seja, objetivou-se desenvolver um esquema relacional com a utilização de funções que atuam com as mesmas características da ontologia.

3 METODOLOGIA

A necessidade de armazenar ontologias em bases de dados relacionais tem crescido significativamente. Um dos aspectos relevantes é a preservação semântica da ontologia no tocante ao armazenamento no banco de dados relacional. Assim, esta proposta de pesquisa busca garantir a persistência de dados a partir do mapeamento de ontologia para esquema de banco de dados relacional. Para isso, os conceitos serão mapeados para tabelas relacionais, bem como, relações e atributos. Já a semântica de restrições e propriedades serão armazenadas em tabelas exclusivas. Pretende-se com isto, obter esquemas relacionais pertinentes e preservar a semântica da ontologia.

A execução deste trabalho foi dividida e apresentada em cinco etapas principais. Primeiramente, foi realizado o levantamento bibliográfico, seguido da análise do material obtido no levantamento. A terceira etapa consistiu na elaboração da etapa metodológica; permitindo executar a construção; e por fim a validação. As etapas mencionadas estão representadas na Figura 6.

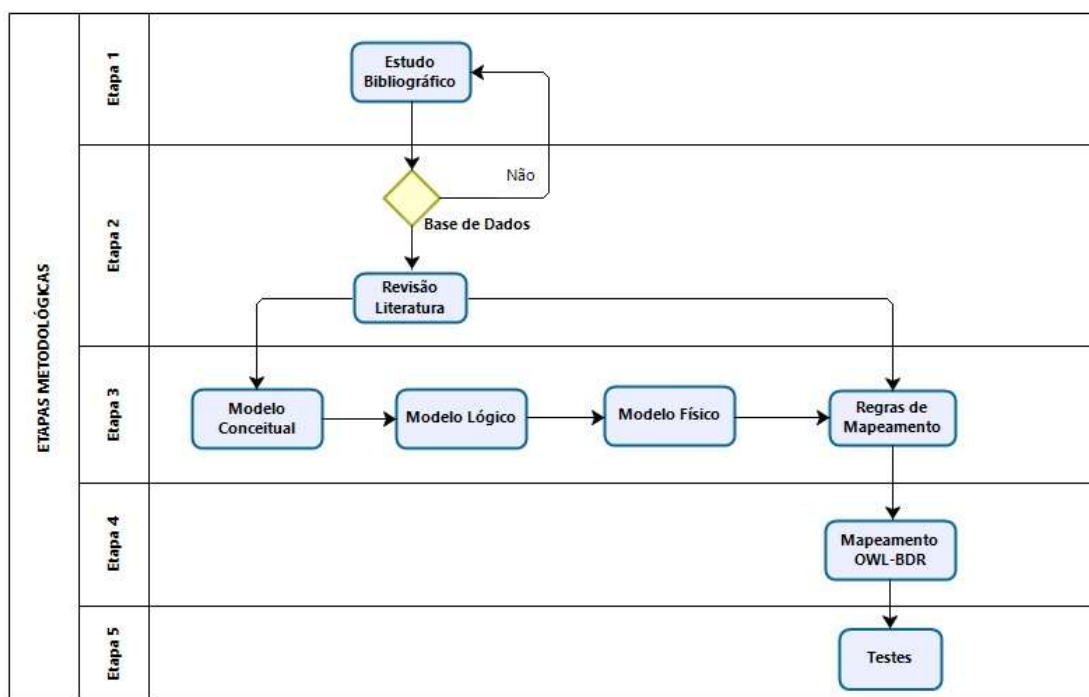


Figura 6. Etapas Metodológicas

Fonte: Dados da Pesquisa, 2019.

Na Etapa 1 foi realizado um estudo bibliográfico em torno dos principais termos envolvidos. Assim, foram considerados trabalhos relevantes sobre ontologias e banco de dados. Considerando que a construção do conhecimento é um processo contínuo, quando havia necessidade de nova busca na base de dados e um novo trabalho era identificado, o

mesmo era incluído. Após este levantamento foi possível analisar os dados coletados (Etapa 2), ou seja, verificar os principais termos; avaliar mecanismos que permitam a persistência de dados; identificar ferramentas e estratégias de mapeamento entre a ontologia e o banco de dados relacional; e explorar condições para preservação semântica. Os resultados das etapas 1 e 2 foram apresentados em capítulos anteriores.

Em seguida, a Etapa 3 trata especificamente e paralelamente de:

(I) *Delinear as fases da modelagem de dados*. Silberschatz (2012), disserta que para todo projeto de banco de dados é preciso elaborar de maneira precisa as principais fases da modelagem de dados (Figura 7): aquisição e abstração do mundo real, criação do modelo conceitual, elaboração do modelo lógico e implementação do modelo físico. De modo que, a representação computacional satisfaça a realidade, absorvendo todo conhecimento preciso neste processo.

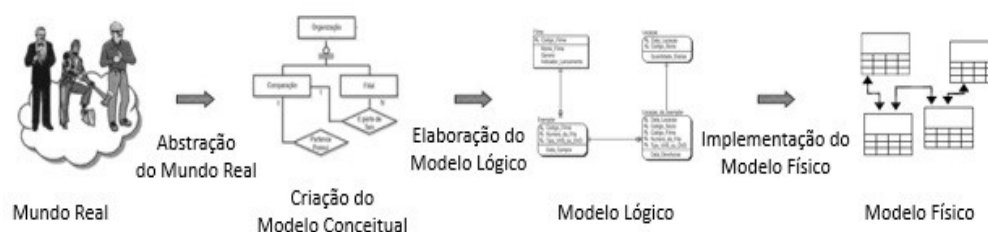


Figura 7. Etapas Modelagem de Dados

Fonte: Dados da Pesquisa, 2019.

Para compreender a realidade (mundo real) foi escolhida como método de pesquisa a entrevista semiestruturada, pois ela permite explorar com mais detalhes o objeto em estudo. Esta fase foi conduzida para verificar se a realidade está em conformidade com as definições da OntoDecideDRC, e compreendeu uma entrevista semiestruturada com médicos da Atenção Primária e médicos especialistas em nefrologia com o intuito de entender a realidade nesses tipos de atendimento, considerando aspectos como: informações obtidas junto aos pacientes, recursos disponíveis para tomada de decisão, e transferência de conhecimento em cenários de encaminhamento.

Após verificar a conformidade do mundo real com a ontologia foi possível criar o modelo conceitual, elaborar o modelo lógico, e implementar o modelo físico. Foi possível representar e descrever a realidade do ambiente do problema, a partir do modelo conceitual, onde constitui-se a visão global dos principais dados e seus relacionamentos, independentemente dos aspectos de implementação. No modelo lógico, foi considerada a abordagem relacional do SGBD para estruturar e estabelecer a lógica dos relacionamentos existentes entre os dados definidos no modelo conceitual. A partir dos modelos já

definidos, foi implementado o modelo físico, onde as estruturas físicas de armazenamento de dados foram descritas.

(II) *Definir as regras para o mapeamento ontológico.* Ernestas (2012) afirma que para ser armazenada em um banco de dados relacional sem perda de informação, a ontologia deve estar em conformidade com as regras do projeto. Além disso, deve ser precisa, correta e semanticamente adequada. Para isso a OntoDecideDRC foi examinada em detalhes, objetivando entender sua estrutura hierárquica. Essa execução na etapa de Projeto fez-se necessária também para delimitar as classes e propriedades que auxiliem a tomada de decisão na atenção primária. As regras de mapeamento entre ontologia e o banco de dados relacional foi detalhada no capítulo de Resultados e Discussão.

A Etapa 4 disserta detalhadamente o mapeamento manual da ontologia para o banco de dados relacional. Nesta etapa será apresentada a contribuição principal deste estudo. A hierarquia das classes da ontologia foram mapeadas através de um conjunto de tabelas e de funções armazenados criadas no SGBD *PostgreSQL*, devido seu desempenho na construção de *stored procedures*, resultando no banco de dados DB-Ontology. É importante ressaltar que as funções desenvolvidas no banco de dados atuam como um “raciocinador” da ontologia, isto é, adicionam a capacidade de controlar a execução das instruções SQL através da utilização de uma linguagem procedural, sendo determinantes nas tomadas de decisões no banco.

Por último, na Etapa 5 foram realizados testes que permitam validar o DB-Ontology, ou seja, foi desenvolvido um estudo de caso utilizando a abordagem para persistência de dados em ontologia de domínio biomédico. Além disso, para testar as funcionalidades criou-se cenários de suporte à decisão na atenção primária. Dentre os cenários podemos citar: duas situações onde os exames não detectaram a DRC; dois casos para identificar o estadiamento da doença e duas ocorrências que exijam como decisão o encaminhamento para médico especialista.

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta principalmente a resolução de pesquisa referente as etapas 4 e 5 definidas na Seção Metodológica. Assim, serão dissertadas as regras de mapeamento, os resultados da modelagem de dados, desenvolvimento de funções para que atuam de maneira inteligente para suporte a decisão e a validação destes artefatos.

4.1 REGRAS DE MAPEAMENTO

As regras de mapeamento foram definidas com base nas especificações utilizadas por Loan (2015), Enerstas (2012) e Irina (2007), na qual dissertam a necessidade de estabelecer regras de mapeamento como primeiro passo a ser executado no processo de mapeamento OWL/BDR.

Um dos conceitos básicos da ontologia que define um grupo de indivíduos juntos é a Classe. O mapeamento das classes é a fase principal, pois permite posteriormente a conversão dos demais conceitos OWL. Assim, uma classe nomeada será mapeada para uma tabela do banco de dados relacional. Considerando que na ontologia os nomes das classes são exclusivos, essa tabela é nomeada com o mesmo nome da classe e é gerado uma chave primária automaticamente. Como por exemplo (Figura 8), a classe Exame é mapeada para a tabela Exame que contém a chave primária ExameCod.



Figura 8. Mapeamento Classe Exame para Tabela Exame

A taxinomia das classes é mapeada para um relacionamento com uso de herança. Isto é, a tabela criada corresponde à subclasse (Figura 9). Esta tabela obtém sua chave primária como uma chave estrangeira relacionada à sua tabela de superclasse. É importante ressaltar que uma classe também relaciona outras classes. Assim uma tabela é criada para cada classe na ontologia com relações um-para-um entre classes e suas subclasses.

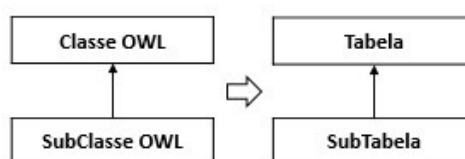


Figura 9. Mapeamento de SubClasses para Banco de Dados Relacional

Para as Propriedades, a OWL possui dois tipos diferentes: propriedades do tipo de dados e propriedades do objeto. As propriedades do tipo de dados evidenciam a relação entre as instâncias de uma classe e valores de tipos de dados. Já as propriedades do tipo do objeto estão relacionadas as instâncias entre duas classes. Também deve ser levado em consideração a cardinalidade, uma vez que, as propriedades podem ser de valor único ou múltiplo. Se a cardinalidade for de valor máximo (1), a propriedade configura-se com valor único. Os demais casos a propriedade terão valor múltiplo.

No mapeamento, a propriedade do objeto é transformada em chave estrangeira ou tabela intermediária no banco de dados. Além disso, considerando o tipo de propriedade o relacionamento entre tabelas pode ser de um para muitos ou de muitos para muitos. Neste último caso deve ser criada uma tabela intermediária. Já no segundo tipo, para cada propriedade do tipo de dados é feito o mapeamento para uma coluna da tabela. Tal tabela equivale à classe especificada no domínio da propriedade e a coluna será nomeada com mesmo nome da propriedade de tipo de dados. Posteriormente à conversão de XSD para SQL do tipo de coluna.

Segundo Ernestas (2012), o SQL não oferece suporte a colunas com valores diferentes em caso de propriedades do tipo de valor multivalorado. Assim, como alternativa, a propriedade será mapeada para uma tabela. Essa tabela será nomeada com mesmo nome da propriedade do tipo de dados. Além disso, a chave primária combina a tabela correspondente à uma chave estrangeira que relaciona a tabela que corresponde a classe especificada no domínio da propriedade. Caso exista restrição em um tipo de dado, ele será mapeado para uma coluna correspondente (restrição de verificação).

Apesar dos tipos RDF, o OWL ainda possibilita construir intervalo de valores de dados, ou seja, mapeamento para uma coluna com restrição de verificação. Esse tipo de construção pode ser identificado, por exemplo, na propriedade idade na classe Informações Básicas (OntoDecideDRC) que possui vários valores (adulto, adolescente, criança, prematura e idoso).

A OWL também disponibiliza mecanismos de raciocínio sobre as propriedades a partir da definição de características de propriedade. Assim, definiu-se propriedade simétrica com valor único, propriedade simétrica multivalorada e propriedade transitiva.

Considerando que a propriedade simétrica seja de valor único, a regra estabelecerá a transformação da propriedade em chave estrangeira. Esse tipo de ocorrência é identificado em casos onde a propriedade é mapeada para a coluna de uma tabela onde esta coluna também é chave estrangeira na mesma tabela. Já em casos de propriedade simétrica multivalorada, o mapeamento é feito para uma tabela. Essa tabela será composta

por sua chave primária e pelas chaves estrangeiras que fazem referência à tabela correspondente. Por fim, o mapeamento em casos que a propriedade é transitiva ocorrerá da mesma forma que a propriedade simétrica multivalorada, ou seja, mapeado para a tabela correspondente. Esse tipo de propriedade é identificado quando existe um tipo de “herança” na coluna. Por exemplo, o hereditarismo da DRC, onde o paciente pode ter registro histórico familiar da doença, ou seja, herdar a doença dos avós ou pais.

De acordo com Zarembo (2015), para limitar o intervalo de uma propriedade em cenário específico, a OWL utiliza a restrição de propriedade. Assim, podemos manipular as restrições de propriedades tanto para propriedades de tipo de dados quanto para propriedades de objeto. Os dados oriundos deste tipo de propriedade são armazenados em tabelas especiais com o intuito de preservar a semântica dos dados durante o mapeamento. Além disso, serão criadas tabelas para cada tipo de restrição de propriedade.

A última etapa do mapeamento da ontologia OntoDecideDRC para banco de dados relacional trata da inserção das instâncias. Assim, após mapear as classes, propriedades e restrições, serão acrescentadas nas linhas de suas respectivas tabelas todas as instâncias das classes. Além disso, a partir das questões de competências definidas os axiomas serão verificados com base nos procedimentos criados com *stored procedures* que os representam no ambiente de banco de dados relacional desta abordagem.

4.2 MODELO CONCEITUAL

A modelagem conceitual partiu da estrutura hierárquica (classes e propriedades) da OntoDecideDRC em consonância com a abstração da realidade (mundo real). Isto é, o projeto do ambiente de aplicação do banco de dados completo que atenda às necessidades dos profissionais da atenção primária, atentando ao amplo conjunto de aspectos.

Considerando o escopo do estudo que busca desenvolver mecanismos para auxiliar a tomada de decisão desses profissionais, foi definido no modelo conceitual as entidades Paciente, Médico, Exame, Fatores Complementares e Resultado, bem como seus respectivos relacionamentos. A modelagem desenvolvida resultou na definição de um diagrama com as estruturas que serão criadas no banco de dados DB-Ontology, conforme Figura 10.

O modelo conceitual é composto por entidades homônimas aos tipos de dados da ontologia, onde a entidade Médico representa a abstração da classe médico; a entidade Paciente absorve as características da classe paciente e informações básicas; a entidade Exame é composta pelos principais atributos para diagnóstico da DRC; a entidade Fatores Complementares representa os elementos que devem ser considerados no processo de

tomada de decisão referente à DRC; a entidade Resultado objetiva armazenar o histórico dos pacientes.

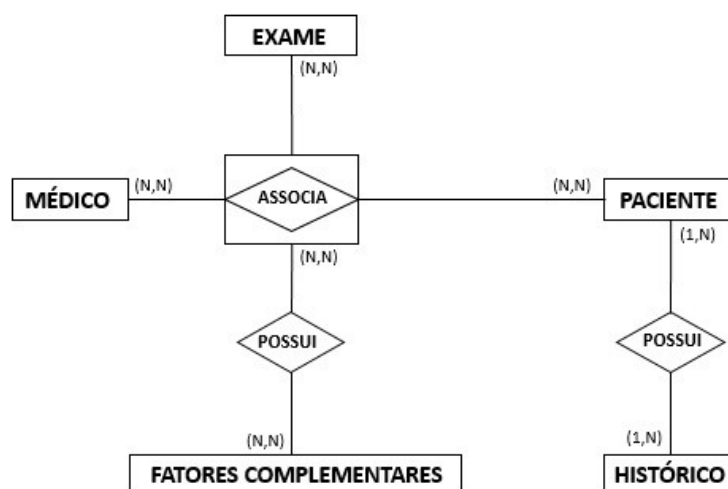


Figura 10. Modelo Conceitual.

4.3 MODELO LÓGICO

Definida a modelagem conceitual, partiu-se para a representação dos objetos, características e relacionamentos conforme as regras de implementação e possíveis limitantes. Isto é, foi elaborado o modelo lógico (Figura 11), respeitando e implementando os conceitos com chaves de acesso, controle de chaves duplicadas, itens de repetição e normalização. Este último, refere-se ao processo organizacional no projeto do DB-Ontology para diminuir a redundância dos dados, aumentar a integridade dos dados e o desempenho, ou seja, examina todos os artefatos objetivando evitar anomalias durante os registros.

Com o intuito de padronizar o banco de dados foi definido desde o modelo lógico, as seguintes nomenclaturas para os objetos do DB-Ontology:

- Para as tabelas, acrescentou-se o prefixo TB_ antes dos nomes de cada tabela a ser criada. A exemplo de TB_Paciente.
- Nos campos das tabelas, identificou-se diferentes tipos de dados com as seguintes nomenclaturas:
 - Para tipos numéricos, adotou-se o prefixo nu_ antes do nome do campo. A exemplo de nu_SUS e nu_CRM.
 - Para o tipo de caracter, utilizou-se o prefixo nm_ antes do nome do campo. A exemplo de nm_Paciente e nm_Medico.
 - Para o tipo de tempo, adotou-se o prefixo ts_ antes do nome do campo. A exemplo de ts_DataNasc.

- Para as funções (create function) desenvolvidas no modelo físico, utilizou-se o prefixo F_ no início do nome destas.

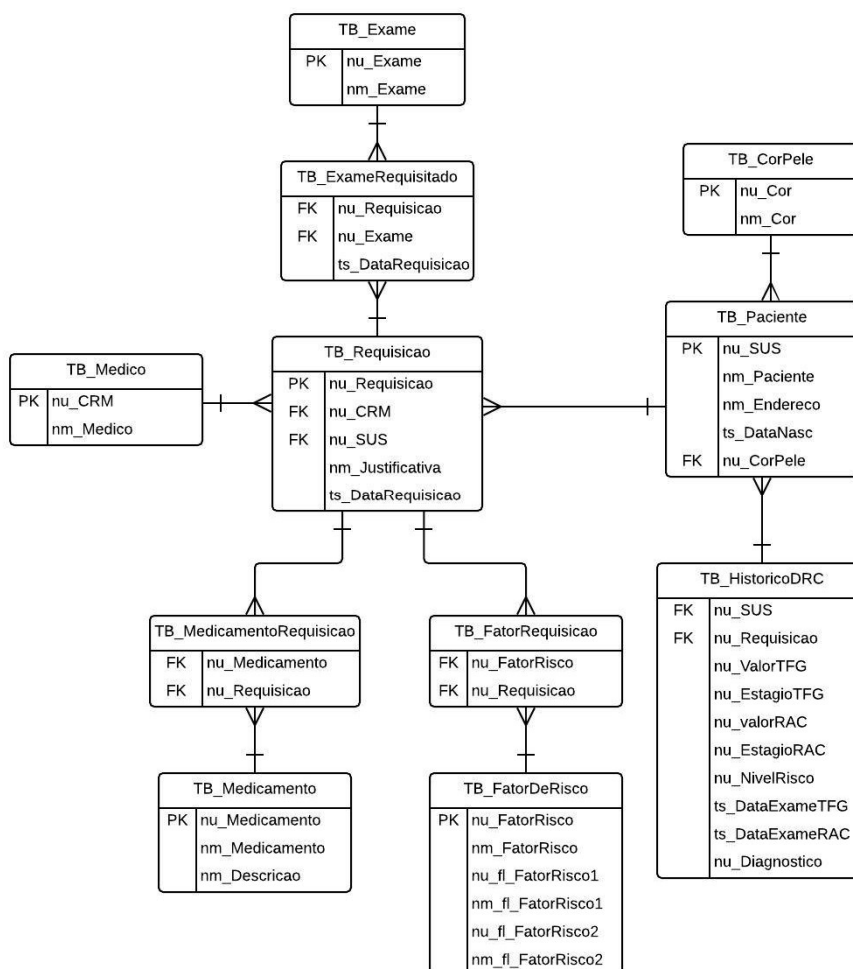


Figura 11. Modelo Lógico.

4.4 DESENVOLVIMENTO DO PROCESSO DE RACIOCÍNIO

As camadas da ontologia permitem a definição de classes e propriedades para a conceitualização e formalização de um domínio, estas quais, estão representadas no banco de dados relacional por tabelas e relacionamentos. Porém, a camada ontológica não é suficiente para descrever por completo as relações lógicas existentes em determinado domínio, uma vez que, inclui apenas composições de hierarquias de classes e propriedades.

Assim, a fim de complementar a modelagem semântica dos domínios, a OntoDecideDRC dispõe de regras semânticas que foram adicionadas à ontologia na forma de axiomas, objetivando realizar inferências e chegar a conclusões. Neste contexto, no projeto de banco de dados relacional foram desenvolvidas regras semânticas na forma de sentenças (linhas de comando) a partir de *stored procedures*, que representam e atuam como

os axiomas dispostos na OntoDecideDRC.

Essas sentenças, foram desenvolvidas no formato (Condições) -> (Ação). A inferência pode ser realizada da seguinte forma: dado um conjunto de condições e combinando-as com determinada tabela, ao se relacionarem verdadeiramente, então uma ação é determinada. Além disso, considerando o escopo proposto neste estudo, o raciocínio foi explicado com foco nos aspectos para diagnóstico e estadiamento da DRC, onde o processo de raciocínio começa na obtenção das principais informações sobre o paciente e encerra com as recomendações aos médicos.

O processo de raciocínio inicia-se com a identificação do tipo de médico que fará uso da abordagem. Considerando que o propósito delineado é apenas aos médicos da atenção primária, esta etapa se justifica importante, pois restringe o escopo de conhecimento, preservando a recomendação de procedimentos ou informações incompatíveis com o tipo de usuário. Esse tipo de informação pode ser obtido por meio de *query* no BD-Ontology.

A etapa seguinte determinou a coleta de informações sobre o paciente. Nesta etapa foi considerada a representação do conhecimento apresentado na OntoDecideDRC, objetivando utilizar as informações realmente relevantes para o processo de raciocínio. Isto é, foram utilizadas as informações alimentadas nas tabelas TB_Paciente, TB_FatoresDeRisco, estas que representam o mapeamento das classes Paciente, Informações Básicas e Fatores de Risco. Além disso, foram consideradas as tabelas TB_Exame e TB_Requisicao, que juntas com as informações do paciente emitem informações relevantes para que pudessem ser realizados raciocínios sobre os procedimentos de diagnóstico e estadiamento da DRC. Adaptavelmente estas tabelas traduzem o conhecimento representado na ontologia do domínio experimental, apresentando a mesma finalidade em suas respectivas classes. Assim, esse conjunto de dados permitiram o desenvolvimento de sentenças para a realização de consultas na base de dados.

Genericamente, a lógica de raciocínio recebe os dados como parâmetros e os formaliza em forma de sentenças dentro das *stored procedures*, com o intuito de executar o raciocínio e identificar orientação de ações. Desta maneira, e a partir delas o processo de raciocínio é iniciado com a entrada de dados dispostos para o delineamento do diagnóstico da doença e estadiamento da DRC. Embora não se tenha produzido interfaces, as operações podem ser executadas diretamente no banco de dados relacional.

Para ressaltar o processo de raciocínio mediante o mapeamento da ontologia para o banco de dados relacional, foram então realizadas as formulações das perguntas de

competência (Tabela 2) para avaliar se os objetivos do DB-Ontology estão de acordo com os artefatos estabelecidos na etapa metodológica de Projeto. Assim, para caracterizar a qualidade do projeto de banco de dados relacional, as *stored procedures* e mecanismos do DB-Ontology devem ser capazes de representar as questões de excelência, caracterizando suas respectivas respostas, assim como ocorreu na OntoDecideDRC. Isto é, assim como as classes e propriedades na ontologia são representadas por tabelas e relacionamentos no banco de dados relacional, os axiomas retratam as sentenças nas funções armazenadas.

Tabela 2. Questões de Competência

Questões de Competência	OntoDecideDRC Exemplos e Definição de Axiomas	DB-Ontology Exemplo e Definição de Sentenças nas <i>Stored Procedures</i>
O que define um diagnóstico de DRC em pacientes?	Um exemplo seria esse axioma relacionado a definição de DRC: <i>“Possui some (T2 and A2)”</i>	Um exemplo seria a função: <i>F_CalcularTFG</i> e <i>F_CalcularRAC</i> .
Como é realizado o estadiamento da DRC?	Um exemplo de definição de como é realizado o estadiamento seria através da propriedade <i>“Classifica”</i> e <i>“É_Classificada_por”</i> , que possuem como domínio e escopo as classes <i>“Taxa de Filtração Glomerular”</i> e <i>“Classificação da DRC”</i> . Indicando que o estadiamento da doença é realizado através dos valores da TFG	Um exemplo de definição de como é realizado o estadiamento seria através da função <i>F_CalcularTFG</i> que recebe como parâmetro o valor referente ao <i>Exame</i> de Taxa de Filtração Glomerular.
Quais as condições para que um paciente seja encaminhado ao médico Nefrologista?	Um exemplo, seria através da <i>query</i> : <i>“Realizar some Encaminhamento”</i> . A inferência retornaria as classes relacionadas aos Estágios 4 e 5, indicando que pacientes classificados nesses estágios devem ser encaminhados para os nefrologistas.	Um exemplo seria através da execução da função <i>F_DecisaoDRC</i> . Considerando as restrições de médico da atenção primária para estadiamento de 1 a 3B, só após esses estágios que o paciente seria encaminhado para o médico Nefrologista.
Quais os exames essenciais que devem ser solicitados para um paciente com suspeita de DRC?	As subclasses da classe “Exames” representam os principais exames relacionados à investigação da DRC.	A tabela “TB_Exame” armazena os principais exames relacionados à DRC, com ênfase para os exames de TFG e RAC, conforme (KDIGO, 2012).
Quais são os dados essenciais do paciente para poder realizar um diagnóstico preciso ou confiável, estadiamento e encaminhamento?	Os dados essenciais são representados pelas subclasses das classes “Fatores de Risco”, “Informações Básicas” e “Exames”.	Com os valores de TFG e RAC, as funções <i>F_CalcularTFG</i> , <i>F_CalcularRAC</i> , <i>F_Calcular_GrauDeRisco</i> e <i>F_DesicaoDRC</i> , definem

		diagnóstico preciso ou confiável, estadiamento e encaminhamento
--	--	---

Embora tenham sido abordados apenas alguns exemplos, o DB-Ontology é capaz de responder diversas outras questões de competências, isto é, o projeto de banco de dados relacional desenvolvido atende as definições de representação do conhecimento e raciocínio, alcançando objetivos projetados em seu processo metodológico. Mediante as possibilidades de condições que podem ser manipuladas, as principais informações relacionadas a este estudo, disponibilizadas por meio do DB-Ontology são:

- Informações básicas referentes ao processo de diagnóstico da DRC;
- Informações relevantes acerca do procedimento de estadiamento da DRC e a definição de cada estágio dessa síndrome clínica;
- Informações inerentes ao processo decisório para monitoramento de pacientes por médicos da atenção primária ou encaminhamento à médicos especialistas em Nefrologia.

4.5 MODELO FÍSICO

A DRC é caracterizada pela perda lenta, progressiva e irreversível das funções renais. Assim, quanto mais cedo for realizado o diagnóstico, mais precocemente será definido o monitoramento e estadiamento desta síndrome clínica. Segundo recomendações da KDIGO, devem ser considerados como parâmetros a causa da doença, a albuminúria e a TFG para o estadiamento da DRC, conforme apresentado na Figura 3.

A execução do estudo de caso para domínio da nefrologia levou em consideração cenários de aplicação que auxiliem a tomada de decisão para diagnóstico e definissem o grau de risco da doença com base nos parâmetros estabelecidos pela KDIGO. Assim, conforme apresentado na Figura 3, as informações juntas definem o que chamamos de cruzamento de risco, ou seja, para se diagnosticar a DRC é necessário verificar a existência de interseção entre a TFG e RAC. Esse cruzamento de risco é definido pelo algoritmo A_Parametros abaixo, onde ref1 e ref2 representam os pontos de risco.

```

Algoritmo A_Parametros (ref1, ref2)
Início
  Se existe cruzamento de risco entre os valores de ref1 e ref2
  Então
    Obter o dado P(x,y) do cruzamento de risco
    Obter o valor ref1' sobre o dado P(x, y) em ref1
    Obter o valor ref2' sobre o dado P(x,y) em ref2
    Se ref1' = ref2'
      Então resultado <= verdadeiro

```

```

        Senão resultado <= falso
    Fim-se
    Senão resultado <= falso
Fim-se
Fim

```

Para cada entidade do modelo lógico apresentado anteriormente foi criada uma tabela no banco do DB-Ontology que são: TB_Paciente, TB_Medico, TB_Exame, TB_Requisicao, TB_FatorDeRisco, TB_Medicamento, TB_ExameRequisitado, TB_FatorRequisicao, TB_CorPele, TB_HistoricoDRC e TB_MedicamentoRequisicao. Essas tabelas foram geradas buscando especificamente garantir o auxílio à tomada de decisão para diagnóstico da DRC, preservando a semântica e a persistência dos dados.

A tabela TB_Paciente abstrai as classes Paciente e Informações Básicas em única tabela, armazenando em seus campos as principais informações do paciente. Através dos comandos SQL abaixo, a tabela foi criada. Observa-se nesses comandos que a TB_Paciente foi definida com apenas o atributo numérico nu_SUS, que é a chave primária, seguido dos atributos do tipo de caractere: nome do paciente, nome do endereço e cor da pele. Por fim, o tipo tempo para a data de nascimento.

```

CREATE TABLE TB_Paciente (
    nu_SUS integer not null PRIMARY KEY,
    nm_Paciente text not null,
    nm_Endereco text,
    ts_DataNasc timestamp not null,
    nm_CorPele text
);

```

A tabela TB_Medico armazena das informações básicas sobre os Médicos: o número de identificação do Conselho Regional de Medicina, e o nome do profissional, representado respectivamente por nu_Medico e nm_Medico. Por se tratar do atendimento na USF, o escopo da pesquisa restringiu os dados dessa tabela apenas a médicos da atenção primária, uma vez que, a classe Médico da OntoDecideDRC ainda incorpora médicos especialistas.

```

CREATE TABLE TB_Medico (
    nu_CRM integer not null PRIMARY KEY,
    nm_Medico text not null
);

```

A tabela TB_Requisicao além de abstrair a realidade da Atenção Primária, compreende o principal relacionamento entre Paciente e Médico. Essa tabela armazena uma breve descrição da situação clínica do paciente, sendo composta pela chave primária (nu_Requisicao), por nm_Justificativa, nu_Medicamento, ts_DataRequisicao, nu_SUS e

nu_CRM. Sendo estas duas últimas chaves estrangeiras.

```
CREATE TABLE TB_Requisicao (  
    nu_Requisicao integer not null PRIMARY KEY,  
    nu_CRM integer not null,  
    nu_SUS integer not null,  
    nm_Justificativa text,  
    nu_Medicamento integer,  
    ts_DataRequisicao timestamp,  
    FOREIGN KEY (nu_Cor) REFERENCES TB_CorPele (nu_Cor) ON DELETE SET NULL,  
    CONSTRAINT FOREIGN KEY (nu_Medicamento) REFERENCES TB_Medicamento  
    (nu_Medicamento) ON DELETE SET NULL,  
    CONSTRAINT tb_const_requi_nu_crm_PK FOREIGN KEY (nu_CRM) REFERENCES TB_Medico  
    (nu_CRM) ON DELETE SET NULL,  
    CONSTRAINT tb_const_requi_nu_sus_PK FOREIGN KEY (nu_SUS) REFERENCES  
    TB_Paciente (nu_SUS) ON DELETE SET NULL  
);
```

A tabela TB_Exame (comando SQL abaixo) é composta por sua chave primária nu_Exame e o atributo (nm_Nome) que armazena o nome do exame. Os tipos de exames mapeados da classe Exame na ontologia OntoDecideDRC são: albuminúria, alteração de imagem, creatinina, exame sumário de urina, proteinúria, relação albuminúria creatinúria e taxa de filtração glomerular. Esses exames são mecanismos importantes para o diagnóstico da DRC, destacando os dois últimos tipos de exames.

```
CREATE TABLE TB_Exame (  
    nu_Exame integer not null PRIMARY KEY,  
    nm_Exame text not null  
);
```

A tabela TB_FatorDeRisco é resultado do mapeamento da classe Fatores de Risco. Essa tabela armazena informações inerentes as doenças, sintomas e o grupo de risco que tornam o paciente mais propenso a desenvolver a DRC. Para representar esta tabela com grande quantidade de informações, foram utilizados também flags (prefixo fl_) para ajudar a nomear e identificar as colunas da tabela no DB-Ontology; e facilitar a manipulação e manutenção dos dados.

```
CREATE TABLE TB_FatorDeRisco (  
    nu_FatorRisco integer not null PRIMARY KEY,  
    nm_FatorRisco text not null,  
    nu_fl_FatorRisco1 char (1) not null,
```

```

    nm_fl_FatorRisco1 text not null,
    nu_fl_FatorRisco2 char (1) not null,
    nm_fl_FatorRisco2 text not null
);

```

A tabela TB_Medicamento armazena informações relacionadas aos medicamentos que os pacientes possam estar utilizando ou que futuramente poderão vir a utilizar. Esta tabela representa a classe Medicamentos na ontologia. Para torna-la mais precisa foram incluídos os campos nu_Medicamento, para armazenar o código de identificação do medicamento; nm_Medicamento para guardar o nome; e nm_Descrição para conservar anotações de como utilizar esse fármaco.

```

CREATE TABLE TB_Medicamento (
    nu_Medicamento integer not null PRIMARY KEY,
    nm_Medicamento text,
    nm_Descricao text
);

```

A tabela TB_CorDePele busca representar a informação sobre tonalidade da pele, logo possui os campos para identificação e nomeação da cor de pele. Essa tabela representa a subclasse de Informações básicas e estar relacionada com a tabela TB_Paciente, uma vez que, são informações do paciente.

```

CREATE TABLE TB_CorPele (
    nu_Cor integer not null PRIMARY KEY,
    nm_Cor text
);

```

Foram criadas algumas tabelas que atendessem as regras de mapeamento definidas e mativem a normalização, a saber: TB_ExameRequisitado, TB_FatorRequisicao e TB_MedicamentoRequisicao, respectivamente apresentadas abaixo:

```

CREATE TABLE TB_ExameRequisitado (
    nu_Requisicao integer not null,
    nu_Exame integer not null,
    ts_Data timestamp,
    CONSTRAINT tb_cons_exame_requi_nuRequisicao_PK PRIMARY KEY (nu_Requisicao,
    nu_Exame),
    CONSTRAINT tb_const_exame_requi_nu_Exame_PK FOREIGN KEY (nu_Exame) REFERENCES

```

```
TB_Exame (nu_Exame) ON DELETE CASCADE
);
```

```
CREATE TABLE TB_FatorRequisicao (
    nu_Requisicao integer not null,
    nu_FatorRisco integer,
    CONSTRAINT tb_cons_requi_nu_fator_PK PRIMARY KEY (nu_Requisicao,
    nu_FatorRisco),
    CONSTRAINT tb_const_requi_nu_fator_FK FOREIGN KEY (nu_FatorRisco) REFERENCES
    TB_FatorDeRisco (nu_FatorRisco) ON DELETE CASCADE
);
```

```
CREATE TABLE TB_MedicamentoRequisicao (
    nu_Medicamento integer,
    nu_Requisicao integer,
    CONSTRAINT tb_cons_requi_nu_medicamento_PK PRIMARY KEY (nu_Requisicao,
    nu_Medicamento),
    CONSTRAINT tb_const_requi_nu_medicamento_FK FOREIGN KEY (nu_Medicamento)
    REFERENCES TB_Medicamento (nu_Medicamento) ON DELETE CASCADE,
    CONSTRAINT tb_const_requi_nu_medicamentos_FK FOREIGN KEY (nu_Requisicao)
    REFERENCES TB_Requisicao (nu_Requisicao) ON DELETE CASCADE
);
```

Por fim, foi criado a tabela TB_DiagnosticoDRC. Essa tabela armazena os dados do paciente com base nas informações fornecidas e manipuladas para a tomada de decisão, estabelecendo redundância mínima. Isto é, essa tabela representa o histórico do paciente, armazenando as principais informações inerentes a verificação e estadiamento da DRC para os médicos da atenção primária. Facilitando o acesso a todos os registros do paciente em relação a doença.

```
CREATE TABLE TB_HistoricoDRC (
    nu_SUS integer not null PRIMARY KEY,
    nu_Requisicao integer,
    nu_ValorTFG integer not null,
    nm_EstagioTFG integer not null,
    nu_valorRAC integer not null,
    nm_EstagioRAC integer not null,
    nm_NivelRisco integer not null,
    nm_Diagnostico text not null,
    ts_DataExameTFG timestamp,
    ts_DataExameRAC timestamp,
```

```

FOREIGN KEY (nu_Requisicao) REFERENCES TB_Requisicao,
FOREIGN KEY (nu_SUS) REFERENCES TB_Paciente
);

```

As tabelas criadas foram alimentadas com dados fictícios. Assim, foram realizados comandos SQL para inserção de dados sem auxílio de função. É importante ressaltar que os dados inseridos correspondem as informações apresentadas na OntoDecideDRC, sendo essas adaptadas para o DB-Ontology. A seguir será demonstrado apenas uma inserção de cada tabela. A inserção completada pode ser verificada no Anexo A. É importante ressaltar que a TB_HistoricoDRC só foi alimentada no processo de avaliação da abordagem, uma vez que, dados das funções são armazenados automaticamente nessa tabela após a execução.

- **Inserção de dados na TB_Paciente**

```

INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc,
nu_CorPele)

```

```

VALUES (1, 'Erick Sutilly', 'R. Projetada, 250', '1940-09-12',1);

```

Inserção de dados na TB_Medico

```

INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (1, 'Vilma Cruz ');

```

- **Inserção de dados na TB_Requisicao**

```

INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa,
ts_DataRequisicao)

```

```

VALUES (1,1,1,'Dor no meio das costas','2018-11-19');

```

- **Inserção de dados na TB_Exame**

```

INSERT INTO TB_Exame (nu_Exame, nm_Exame)

```

```

VALUES (1, 'Taxa de Filtração Glomerular (TFG)');

```

- **Inserção de dados na TB_FatorDeRisco**

```

INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)

```

```

VALUES (1, 'Doença','D', 'Doença Túbulo Intersticial','U','Unico');

```

- **Inserção de dados na tabela TB_Medicamento**

```

INSERT INTO TB_Medicamento (nu_Medicamento, nm_Medicamento, nm_Descricao)
VALUES (1, 'Anti-Hipertensivo','2x após refeição');

```

- **Inserção de dados na TB_CorPele**

```

INSERT INTO TB_CorPele (nu_Cor, nm_Cor) VALUES (1,'Branca');

```

- **Inserção de dados na TB_ExameRequisitado**

```

INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)

```

```

VALUES (1,1,'2018-12-12');

```

- **Inserção de dados na tabela TB_FatorRequisicao**

```

INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (0,4);

```

- **Inserção de dados na tabela TB_MedicamentoRequisicao**

```
INSERT INTO TB_MedicamentoRequisicao (nu_Medicamento, nu_Requisicao)
VALUES (1,1);
```

- **Inserção de dados na TB_HistoricoDRC**

```
INSERT INTO TB_HistoricoDRC (nu_SUS, nu_Requisicao, nu_ValorTFG,
nu_EstagioTFG, nu_valorRAC, nu_EstagioRAC, nu_NivelRisco,ts_DataExameTFG,
ts_DataExameRAC, nu_Diagnostico)
VALUES (1,1,100,1,25,1,1,'2018-12-12','2018-12-12',1);
```

4.6 FUNÇÕES NO POSTGRESQL

A inserção de dados no DB-Ontology pode ser realizada também com o auxílio de funções (stored procedures). A função F_Inserir_Dados, definida abaixo, mostra como pode ser gerado requisições de forma dinâmica, utilizando funções já disponíveis no DB-Ontology. Para isso foi criada também a função F_NewRequisicao. Ambas funções são demonstrações que podem ser adaptadas e utilizadas por outras tabelas do banco de dados.

```
CREATE OR REPLACE FUNCTION F_InserirDados () RETURNS void
AS
$$
BEGIN
    RAISE NOTICE '<<< INSERINDO DADOS NO DB-Ontology >>>';
    perform F_NewRequisicao (3, 2, 'Exame de rotina', 1, 1, '2018-11-05');
    perform F_NewRequisicao (3, 2, 'Dor nas costas', 1, 1, '2018-10-04');
    perform F_NewRequisicao (3, 2, 'Problemas urinários', 1, 1, '2018-10-03');
    perform F_NewRequisicao (3, 2, 'Dor de cabeça', 1, 1, '2018-10-02');
    RAISE NOTICE ' <<< FIM >>>';
END;
$$ LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION F_NewRequisicao (medico integer, paciente integer,
justificativa text, data timestamp) RETURNS integer
AS
$$
DECLARE
    new_nu_Requisicao integer;
BEGIN
    SELECT max(nu_Requisicao)+1 INTO new_nu_Requisicao FROM TB_Requisicao;

    INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa,
ts_DataRequisicao)
    VALUES (new_nu_Requisicao, medico, paciente, justificativa, data);

    RETURN new_nu_Requisicao;
END;
$$ LANGUAGE 'plpgsql';
```

A função F_Inserção_Medico foi criada para identificar se um médico que atende na USF possui cadastro. Com a crescente demanda por cuidados em saúde e baixa quantidade de profissionais na área, na prática existe uma rotatividade minimamente crescente na atenção

primária. Assim, essa função garante que se deparando com a necessidade de atualização de dados do médico ou novo cadastro de médico naquela unidade, exista mecanismos de suporte para esse tipo de atividade.

```
CREATE OR REPLACE FUNCTION F_Inserção_Medico (nu_CRMIn integer, medico text) RETURNS
integer
AS
$$

DECLARE
    new_nu_CRM integer;
    new_nm_Medico text;

BEGIN
    new_nm_Medico = medico;
    SELECT nu_CRM INTO new_nu_CRM from TB_Medico where nu_CRM = nu_CRMIn;

    IF nu_CRMIn = new_nu_CRM THEN
        RAISE NOTICE '<<< CRM EXISTENTE >>> ';
        UPDATE TB_Medico set nm_Medico = new_nm_Medico where nu_CRM = new_nu_CRM;
        RAISE NOTICE 'ATUALIZAÇÃO DOS DADOS EFETIVADA! ';
        return new_nu_CRM;
    ELSE
        IF new_nu_CRM IS NULL THEN
            RAISE NOTICE '<<< NOVO CRM SERÁ CADASTRADO >>> ';
            INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (nu_CRMIn,
new_nm_Medico);
            RAISE NOTICE '<<< DADOS CADASTROS >>> ';
        END IF;
    END IF;
    return new_nu_CRM;
END;
$$LANGUAGE 'plpgsql';
```

Foi criado também função para os casos em que é a primeira vez que o paciente realiza consulta na USF. Assim, seus dados podem não estar cadastrados no DB-ontology. As funções F_Criar_Paciente e F_Inserir_Paciente objetivam solucionar esse tipo de ocorrência.

```
CREATE OR REPLACE FUNCTION F_Criar_Paciente (integer) RETURNS void
AS
$$

DECLARE
    new_nu_Paciente integer;

BEGIN
    SELECT nu_SUS INTO new_nu_Paciente from TB_Paciente where nu_SUS = nu_SUSIn;
    IF (new_nu_Paciente = nu_SUSIn) THEN
        RAISE NOTICE ' USUARIO EXISTENTE';
    ELSE
        IF new_nu_Paciente IS NULL THEN
```

```

        RAISE NOTICE 'USUARIO NOVO IDENTIFICADO! ';
        perform F_ADD_Paciente (12, 'João Guaribon', 'R. Santa, s/n',
'1995-07-05', 1);
        RAISE NOTICE 'depois de mandar executar perform. ';
    END IF;
END IF;
END;
$$ LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION F_Inserir_Paciente (nu_SUSin integer,
new_nm_Paciente text, new_nm_Endereco text, new_ts_DataNasc timestamp,
new_nu_CorPele integer) RETURNS integer
AS
$$
DECLARE
new_nu_SUS integer;
BEGIN
SELECT max (nu_SUS)+1 INTO new_nu_SUS FROM TB_Paciente;
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc,
nu_CorPele)
VALUES (nu_SUSin, new_nm_Paciente, new_nm_Endereco, new_ts_DataNasc,
new_nu_CorPele);
END;
$$ LANGUAGE 'plpgsql';

```

Considerando a necessidade de algumas funções auxiliares, foram criadas as funções F_Tratar_Idade para sanar problemas como: formato de data de nascimento e verificar idade com base na data de nascimento do paciente: F_CalcularIdade.

```

CREATE OR REPLACE FUNCTION F_Tratar_Idade (nu_SUSin bigint, nm_PacienteIn text)
RETURNS timestamp
AS
$$
DECLARE
new_nu_SUS bigint;
new_ts_DataNasc timestamp;
new_ts_DatAtual current_Date;
BEGIN
perform set_config ('datestyle', 'dmy', false); -- formato ddmyyy
SELECT ts_DataNasc INTO new_ts_DataNasc from TB_Paciente where nu_SUS = nu_SUSin;
IF nu_SUSin = new_nu_SUS THEN
    UPDATE TB_Paciente set nm_Paciente = nm_PacienteIn, ts_DataNasc =
new_ts_DataNasc where nu_SUS = new_nu_SUS;
    return new_ts_DataNasc;

```

```

ELSE
IF new_nu_SUS IS NULL THEN

            INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, nm_CorPele,
ts_DataNasc, nm_Sexo) VALUES (nu_SUSin, nm_PacienteIn, 'endereco1', 'cor pele','28-
01-1990', 'M');
        END IF;
    END IF;
    return new_ts_DataNasc;
END;
$$LANGUAGE 'plpgsql';
-----
CREATE OR REPLACE FUNCTION F_CalcularIdade (ts_DataNasc timestamp) RETURNS void
AS
$$
BEGIN
SELECT extract(year from age(p.ts_dataNasc)) FROM TB_Paciente p where nu_Sus = 1
END;
$$ LANGUAGE 'plpgsql';

```

As funções a seguir tratam da tomada de decisão para diagnóstico da DRC. O grupo de funções está incorporado à função F_Ontology que trata de identificar o estágio da TFG, o estágio da RAC, o grau de risco e por fim a decisão para auxiliar o médico, com suas respectivas funções, F_Calcular_Estagio_TFG, F_Calcular_Estagio_RAC, F_Calcular_GrauDeRisco, F_DecisaoDRC. De acordo com a KDIGO, essas informações são necessárias para o diagnóstico da DRC. Isto é, com esses dados o médico da atenção primária poderá com mais exatidão tomar decisões a respeito do monitoramento e encaminhamento do paciente.

```

CREATE Function F_Ontology (integer) RETURNS void
AS
$$
DECLARE
BEGIN
CREATE OR REPLACE FUNCTION F_Calcular_Estagio_TFG (valorTFG integer) RETURNS
integer
AS
$$
DECLARE
estagio_TFG integer;
BEGIN
IF valorTFG >= 90 THEN
        RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >=
90] >>> ';
        RAISE NOTICE ' <<< Valor TFG normal ou elevada >>> ';
        estagio_TFG = 1;
        RAISE NOTICE '<<< PACIENTE EM ESTÁGIO 1 TFG >>> ';

```

```

ELSIF ((valorTFG >= 60) and (valorTFG <= 89)) THEN
    RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG 60-
89] >>> ';
    RAISE NOTICE ' <<< Valor TFG com redução leve normal >>> ';
    estagio_TFG = 2;
    RAISE NOTICE '<<< PACIENTE EM ESTÁGIO 2 TFG >>> ';
ELSIF ((valorTFG >= 45) and (valorTFG <= 59)) THEN
    RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG 45-
59]>>> ';
    RAISE NOTICE ' <<< Valor TFG com redução leve a moderada >>>';
    estagio_TFG = 3;
    RAISE NOTICE '<<< PACIENTE EM ESTÁGIO 3A TFG >>> ';

ELSIF ((valorTFG >= 30) and (valorTFG <= 44)) THEN
    RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG 30-
44] >>>';
    RAISE NOTICE ' <<< Valor TFG com redução moderada a grave >>> ';
    estagio_TFG = 4;
    RAISE NOTICE '<<< PACIENTE EM ESTÁGIO 3B TFG >>> ';
ELSIF ((valorTFG >= 15) and (valorTFG <= 29)) THEN
    RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG 15-
29] >>>';
    RAISE NOTICE ' <<< Valor TFG com redução severa. ';
    estagio_TFG = 5;
    RAISE NOTICE '<<< PACIENTE EM ESTÁGIO 4 TFG >>> ';
ELSIF (valorTFG < 15) THEN
    RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG < 15]
>>> ';
    RAISE NOTICE ' <<< FALÊNCIA RENAL >>> ';
    estagio_TFG = 6;
    RAISE NOTICE '<<< PACIENTE EM ESTÁGIO 5 TFG >>>';
ELSE
    RAISE NOTICE '<<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC >>> ';
    estagio_TFG = NULL;
    RAISE NOTICE '<<< VALOR INEXISTENTE >>>';
END IF;
return estagio_TFG;
END;
$$ LANGUAGE 'plpgsql';
-- =====
CREATE OR REPLACE FUNCTION F_Calcular_Estagio_RAC (valorRAC integer) RETURNS

```

```

text
AS
$$
DECLARE
estagio_RAC integer;
BEGIN
IF valorRAC < 30 THEN
    RAISE NOTICE '<<< REFERÊNCIA PARA DIAGNÓSTICO DRC  valorRAC < 30>>>'
';

    RAISE NOTICE ' <<< Valor RAC normal a levemente aumentada >>> ';
    estagio_RAC = 1;
    RAISE NOTICE 'PACIENTE ESTÁGIO A1 RAC.';

ELSIF ((valorRAC >= 30) and (valorRAC < 300)) THEN
    RAISE NOTICE '<<< REFERÊNCIA PARA DIAGNÓSTICO DRC  ( 30 <= valorRAC
< 300 ) >>> ';
    RAISE NOTICE ' <<< Valor RAC moderadamente aumentada >>> ';
    estagio_RAC = 2;
    RAISE NOTICE 'PACIENTE ESTÁGIO A2 RAC. ';
ELSIF (valorRAC > 300) THEN
    RAISE NOTICE '<<< REFERÊNCIA DIAGNÓSTICO DRC (valorRAC > 300) >>> ';
    RAISE NOTICE ' <<< Valor RAC severamente aumentada >>>';
    estagio_RAC = 3;
    RAISE NOTICE 'PACIENTE ESTÁGIO A3 RAC.';
ELSE
    RAISE NOTICE '<<< REFERÊNCIA PARA DIAGNÓSTICO DRC >>> ';
    estagio_RAC = NULL;
    RAISE NOTICE '<<< Valor informado incorreto ou não existe >>> ';
END IF;
return estagio_RAC;
END;
$$ LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION F_Calcular_GrauDeRisco (valorTFGin integer,
valorRACin integer) RETURNS integer
AS
$$
DECLARE
estagio_TFG integer;
estagio_RAC integer;
grauDeRisco integer;

```

```

resultado1 text;
BEGIN
estagio_TFG = F_Calcular_Estagio_TFG (valorTFGin);
estagio_RAC = F_Calcular_Estagio_RAC (valorRACin);
IF ((estagio_TFG = 1) AND (estagio_RAC = 1)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 1; --(Verde - Baixo risco)
    RAISE NOTICE '<<< BAIXO RISCO [Estágio 1 TFG e Estágio A1 RAC]>>> ';
ELSIF ((estagio_TFG = 2) AND (estagio_RAC = 1)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 1; --(Verde - Baixo risco)
    RAISE NOTICE '<<< BAIXO RISCO [Estágio 2 TFG e Estágio A1 RAC]>>> ';
ELSIF ((estagio_TFG = 3) AND (estagio_RAC = 1)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 2; --(Amarelo - Risco moderadamente aumentado)
    RAISE NOTICE '<<< RISCO MODERADAMENTE AUMENTADO [Estágio 3 TFG e
Estágio A1 RAC]>>> ';
ELSIF ((estagio_TFG = 1) AND (estagio_RAC = 2)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 2; --(Amarelo - Risco moderadamente aumentado)
    RAISE NOTICE '<<< RISCO MODERADAMENTE AUMENTADO [Estágio 1 TFG e
Estágio A2 RAC]>>> ';
ELSIF ((estagio_TFG = 2) AND (estagio_RAC = 2)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 2; --(Amarelo - Risco moderadamente aumentado)
    RAISE NOTICE '<<< RISCO MODERADAMENTE AUMENTADO [Estágio 2 TFG e
Estágio A2 RAC]>>> ';
ELSIF ((estagio_TFG = 4) AND (estagio_RAC = 1)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 3; --(Laranja - Alto Risco)
    RAISE NOTICE '<<< ALTO RISCO [Estágio 4 TFG e Estágio A1 RAC]>>> ';
ELSIF ((estagio_TFG = 3) AND (estagio_RAC = 2)) THEN
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

    grauDeRisco = 3; --(Laranja - Alto Risco)

```

```

        RAISE NOTICE '<<< ALTO RISCO [Estágio 3 TFG e Estágio A2 RAC]>>> ';
ELSIF ((estagio_TFG = 1) AND (estagio_RAC = 3)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 3;
        RAISE NOTICE '<<< ALTO RISCO [Estágio 1 TFG e Estágio A3 RAC]>>> ';
ELSIF ((estagio_TFG = 2) AND (estagio_RAC = 3)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 3;
        RAISE NOTICE '<<< ALTO RISCO [Estágio 2 TFG e Estágio A3 RAC]>>> ';
ELSIF ((estagio_TFG = 4) AND (estagio_RAC = 2)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 4;
        RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 4 TFG e Estágio A2 RAC]>>>
';

        ELSIF ((estagio_TFG = 5) AND (estagio_RAC = 1)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 4;
        RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A1 RAC]
>>> ';

        ELSIF ((estagio_TFG = 5) AND (estagio_RAC = 2)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 4;
        RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A2 RAC]>>>
';

        ELSIF ((estagio_TFG = 3) AND (estagio_RAC = 3)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 4;
        RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 3 TFG e Estágio A3 RAC]>>>
';

        ELSIF ((estagio_TFG = 4) AND (estagio_RAC = 3)) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO
>>> ';

        grauDeRisco = 4;
        RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 4 TFG e Estágio A3 RAC]
>>> ';

```

```

        ELSIF ((estagio_TFG = 5) AND (estagio_RAC = 3)) THEN
            RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

            grauDeRisco = 5;
            RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A3 RAC]
>>> ';

        ELSIF ((estagio_TFG = 6) AND (estagio_RAC = 1)) THEN
            RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

            grauDeRisco = 5;
            RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A1 RAC]
>>> ';

        ELSIF ((estagio_TFG = 6) AND (estagio_RAC = 2)) THEN
            RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

            grauDeRisco = 5; --(Vermelho Escuro - Risco Muito Alto)
            RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A2 RAC]
>>> ';

        ELSIF ((estagio_TFG = 6) AND (estagio_RAC = 3)) THEN
            RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

            grauDeRisco = 5;
            RAISE NOTICE '<<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A3 RAC]
>>> ';

        ELSE
            RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA - GRAU DE RISCO
>>> ';

            grauDeRisco = NULL;
            RAISE NOTICE ' <<< Relação TFG e RAC >>>';

        END IF;

        return grauDeRisco;
    END;
$$ LANGUAGE 'plpgsql';

-- ===== --

CREATE OR REPLACE FUNCTION F_DecisaoDRC (valorTFGin integer, valorRACin
integer, grauDeRiscoIn integer) RETURNS integer
AS
$$
DECLARE
    estagio_TFG integer;
    estagio_RAC integer;

```

```

    grauDeRisco integer;
    decision integer;
    update TB_HistoricoDRC set nu_Diagnostico = 2 where nu_SUS = 2
    BEGIN
    estagio_TFG = F_Calcular_Estagio_TFG (valorTFGin);
    estagio_RAC = F_Calcular_Estagio_RAC (valorRACin);
    grauDeRisco = F_Calcular_GrauDeRisco (valorTFGin, valorRACin);
    IF (grauDeRisco = 1) THEN
        RAISE NOTICE '<<< TOMADA DE DECISÃO PARA DOENÇA RENAL CRÔNICA >>> ';
        decision = 1;
        RAISE NOTICE ' <<< ACOMPANHAR >>> ';
        RAISE NOTICE 'O paciente deve ser acompanhado 1 uma vez por ano na
própria USF se presença da doença. Refazer exames de rotina para verificação de
DRC. ';
    ELSIF (grauDeRisco = 2) THEN
        RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA >>> ';
        decision = 2;
        RAISE NOTICE ' <<< MONITORAR >>>';
        RAISE NOTICE 'O paciente deve ser acompanhado 1 vezes por ano na
própria USF para monitorar a DRC.';
    ELSIF (grauDeRisco = 3) THEN
        IF (((estagio_TFG1 = 1) OR (estagio_TFG2 = 2)) AND (estagio_RAC2 =
3)) THEN
            RAISE NOTICE '<<< TOMADA DE DECISÃO PARA DOENÇA RENAL CRÔNICA
>>> ';
            decision = 3;
            RAISE NOTICE ' <<< ENCAMINHAR >>> ';
            RAISE NOTICE 'O paciente deve ser encaminhando para médico
especialista. Acompanhar 2 vezes por ano a DRC do indivíduo.';
        ELSE
            RAISE NOTICE '<<< TOMADA DE DECISÃO PARA DOENÇA RENAL CRÔNICA
>>> ';
            decision = 4;
            RAISE NOTICE ' <<< MONITORAR >>>';
            RAISE NOTICE 'O paciente deve ser acompanhado 2 vezes por ano
na própria USF para monitorar a DRC. ';
        END IF;
    ELSIF (grauDeRisco = 4) THEN
        IF ((estagio_TFG1 = 4) AND (estagio_RAC2 = 2)) THEN
            RAISE NOTICE '<<< TOMADA DE DECISÃO PARA DOENÇA RENAL CRÔNICA
>>> ';

```

```

        decision = 5;
        RAISE NOTICE ' <<< MONITORAR >>>';
        RAISE NOTICE 'O paciente deve ser acompanhado 3 vezes por ano
na própria USF para monitorar a DRC. ';
    ELSE
        RAISE NOTICE '<<< TOMADA DE DECISÃO PARA DOENÇA RENAL CRÔNICA
>>> ';

        decision = 6;
        RAISE NOTICE ' <<< ENCAMINHAR >>> ';
        RAISE NOTICE 'O paciente deve ser encaminhado para médico
especialista e acompanhado 3 vezes por ano para verificar a DRC.';
    END IF;
ELSE
    RAISE NOTICE '<<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA >>> ';
    decision = 7;
    RAISE NOTICE ' <<< ENCAMINHAR >>> ';
    RAISE NOTICE 'O paciente deve ser encaminhado para médico especialista
e acompanhado 4 vezes por ano ou mais para verificar a DRC.';
    END IF;
    return decision;
END;
$$ LANGUAGE 'plpgsql'
-- =====
UPDATE TB_HistoricoDRC SET nu_EstagioTFG = estagio_TFG, nu_EstagioRAC =
estagio_RAC, nu_NivelRisco = grauDeRisco, nu_Diagnostico = decision;
    RAISE NOTICE 'ATUALIZAÇÃO DOS DADOS EFETIVADA! ';
    INSERT INTO TB_HistoricoDRC (nu_SUS, nu_Requisicao, nu_ValorTFG,
nu_estagioTFG, nu_ValorRAC, nu_EstagioRAC, nu_NivelRisco, ts_DataExameTFG,
ts_DataExameRAC, nu_Diagnostico)
        VALUES (nu_SUS, nu_Requisicao, ValorTFG, estagio_TFG, valorRAC,
estagio_RAC, grauDeRisco, ts_DataExameTFG, ts_DataExameRAC, decision);
    RAISE NOTICE 'ATUALIZAÇÃO DOS DADOS NO HISTÓRICO! ';
    END;
$$ LANGUAGE 'plpgsql';

```

4.7 VERIFICAÇÃO DE FUNCIONALIDADES NO DB-ONTOLOGY

Nessa seção buscou-se testar as funcionalidades do DB-Ontology. Para tanto, consultas SQL foram construídas, utilizando scripts simples e avançados com uso de joins. Para o teste de funções foi construído cenários de aplicação, objetivando validar o estudo.

A consulta (Figura 12) solicitou o código identificador do fator de risco e o nome do fator de risco de paciente com número do cartão do SUS igual a 7. Para isso foram consultadas as tabelas de relacionamentos para gerar a saída de dados.

```
SELECT f.nu_FatorRisco, f.nm_fl_FatorRisco1
FROM TB_Paciente p, TB_Requisicao r, TB_FatorDeRisco f, TB_FatorRequisicao fr
WHERE p.nu_SUS = 7
AND p.nu_SUS = r.nu_SUS
AND r.nu_Requisicao = fr.nu_Requisicao
AND f.nu_FatorRisco = fr.nu_FatorRisco
```

Painel de saída

Saída de Dados

Explain

Mensagens

Histórico

	nu_fatorrisco	nm_fl_fatorris
	integer	text
1	4	Doença Vascular

Figura 12. Teste de consulta-01.

Foi pedido a apresentação do número CRM do médico que atendeu o paciente de identificador (nu_SUS) igual a 1e que justificou a consulta por sentir “dor nas costas”. Além do CRM, exibir o nome do paciente e a justificativa mencionada. Conforme Figura 13.

```
SELECT m.nu_CRM, p.nm_Paciente, r.nm_Justificativa
FROM TB_Medico m, TB_Paciente p, TB_Requisicao r
WHERE p.nu_SUS = 2
AND r.nm_Justificativa = 'Dor no meio das costas'
AND p.nu_SUS = r.nu_SUS AND r.nu_CRM = m.nu_CRM
```

Painel de saída

Saída de Dados

Explain

Mensagens

Histórico

	nu_crm	nm_paciente	nm_justificativa
	integer	text	text
1	8	Juan Huck	Dor no meio das costas

Figura 13. Teste de consulta-02.

A Figura 14, exibe um exemplo de consulta onde foi solicitado o nome do paciente e o nome do exame de número 1. De acordo com a tabela TB_ExameRequisitado, o exame de número 1 é “Taxa de Filtração Glomerular”.

```
SELECT p.nm_Paciente, e.nm_Exame
FROM TB_Paciente p, TB_Requisicao r, TB_Exame e, TB_ExameRequisitado er
WHERE e.nu_Exame = 1 AND p.nu_SUS = r.nu_SUS AND e.nu_Exame = er.nu_Exame
AND er.nu_Requisicao = r.nu_Requisicao
```

Painel de saída

Saída de Dados

Explain

Mensagens

Histórico

	nm_paciente	nm_exame
	text	text
1	Erick Sutilly	Taxa de Filtração Glomerular
2	Erick Sutilly	Taxa de Filtração Glomerular
3	Juan Huck	Taxa de Filtração Glomerular
4	Suraia Rezente	Taxa de Filtração Glomerular

Figura 14. Teste de consulta-03

```
SELECT extract(year from age(p.ts_dataNasc))
FROM TB_Paciente p where nu_Sus = 1;
```

Painel de saída

Saída de Dados

Explain

	date_part
	double precis
1	78

```
SELECT ts_DataNasc FROM TB_Paciente where nu_Sus = 1
```

Painel de saída

Saída de Dados

Explain

	ts_datana
	timestamp
1	1940-09-12

Figura 15. Teste de consulta-04

Com o uso da função age é possível receber a data de nascimento e calcular a idade do usuário, conforme apresentado na Figura 15. Após realizar alguns testes para verificar o funcionamento do DB-ontology, foram construídos cenários para testar as funções principais, que tange o suporte a tomada de decisão para profissionais da atenção primárias na USF, de modo que, esse mecanismo maximize o poder decisório, minimize a falha de comunicação entre médicos da atenção primária e médicos especialistas e preconize o diagnóstico da DRC.

4.8 AVALIAÇÃO DO DB-ONTOLOGY

Para validar as funções criadas foram gerados alguns cenários de aplicação. A partir da definição desses cenários foram utilizadas as funções desenvolvidas no DB-Ontology para tomada de decisão dos médicos da atenção primaria.

Cenário 1: Um paciente que faz parte do grupo de risco por ser portador de diabetes e utilizar medicamentos para esse tipo de enfermidade, bem como analgésicos. Com 38 anos e cor de pele branca, apresentou-se na USF para realizar exames se queixando de sangramentos. O médico que o atendeu, apesar de não ter suspeitas e sem considerar ou temer grandes fatores de risco, resolveu solicitar exames de taxa de filtração glomerular (TFG) e Relação Albuminúria Creatinúria (RAC) para verificar possíveis suspeitas renais. O paciente fez os exames e retornou a USF para apresentar os resultados da qual o exame de TFG tinha valor 100 (ml/min 1.73m²) e RAC 10 ml/g. Pelo resultado apresentado e pela experiência profissional, o médico ficou com dúvida quanto à decisão para diagnóstico daquele paciente.

Primeiramente o médico da atenção primária, já tendo disponível o valor do exame de Taxa de Filtração Glomerular, a lógica de raciocínio executa sentenças no DB-Ontology. Este profissional fornece os dados da TFG disponibilizadas no exame do paciente. Com a inserção do valor da taxa de filtração glomerular, chamou-se a função F_Calcular_Estagio_TFG passando como parâmetro de entrada o valor 100 do exame TFG. Essa função (Figura 16) é uma função dentro da função F_Ontology. Além da saída de dados o banco disponibiliza mensagem de notificação, conforme abaixo.

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >= 90] >>>

NOTICE: <<< Valor TFG normal ou elevada >>>

NOTICE: <<< PACIENTE EM ESTÁGIO 1 TFG >>>

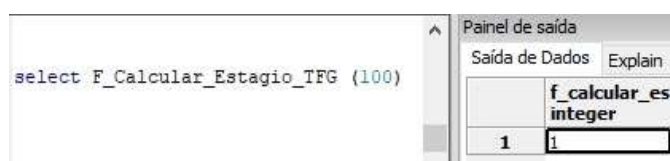


Figura 16. Consulta na função estágio da TFG-01

Definido o estágio da TFG (1) – saída de dados é necessário executar outras funções, uma vez que, apenas com o valor da TFG não é possível diagnosticar a DRC. Assim, foi invocado a função F_Calcular_Estagio_RAC, disponibilizando como parâmetro de entrada o valor do exame de relação albuminúria e creatinúria. Com o valor do exame é possível identificar o estágio RAC isolado do paciente. Conforme mensagem e imagem (Figura 17) abaixo.

NOTICE: <<< REFERÊNCIA PARA DIAGNÓSTICO DRC valorRAC < 30>>>

NOTICE: <<< Valor RAC normal a levemente aumentada >>>

NOTICE: PACIENTE ESTÁGIO A1 RAC.

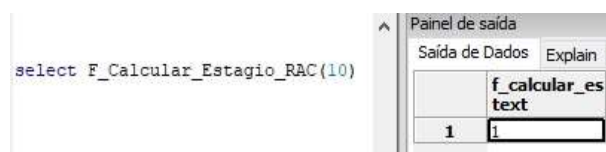


Figura 17. Consulta na função estágio da RAC-01

Após a consulta, observou-se que o valor da RAC informado se configura no primeiro estágio, o que representa na tabela da KDIGO o estágio A1. Como possíveis estágios no banco de dados com saída igual a 1 ou 2, representam, respectivamente os estágios A1 e A2 da RAC. Definida os estágios da TFG e RAC individualmente, foi possível identificar o grau de risco da doença.

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >= 90] >>>

NOTICE: <<< Valor TFG normal ou elevada >>>

NOTICE: <<< PACIENTE EM ESTÁGIO 1 TFG >>>

NOTICE: <<< REFERÊNCIA PARA DIAGNÓSTICO DRC valorRAC < 30>>>

NOTICE: <<< Valor RAC normal a levemente aumentada >>>

NOTICE: PACIENTE ESTÁGIO A1 RAC.

NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO >>>

NOTICE: <<< BAIXO RISCO [Estágio 1 TFG e Estágio A1 RAC]>>>

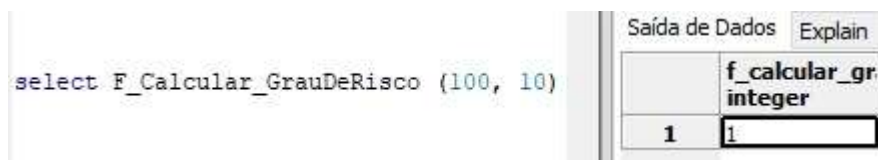


Figura 18. Consulta na função grau de risco-01

A função F_Calcular_GrauDeRisco (Figura 18) recebe como parâmetro de entrada os estágios da TFG e da RAC e verifica o grau de risco da DRC. Em seguida, definido o estágio de cada um dos exames coletados e verificado o grau de risco, a função decisão (Figura 19) retorna a tomada de decisão que auxiliará o médico da atenção primária a definir se o paciente deve ser encaminhado para um médico especialista ou monitorado na

própria USF, juntamente com a mensagem de execução.

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >= 90] >>>

NOTICE: <<< Valor TFG normal ou elevada >>>

NOTICE: <<< PACIENTE EM ESTÁGIO 1 TFG >>>

NOTICE: <<< REFERÊNCIA PARA DIAGNÓSTICO DRC valorRAC < 30>>>

NOTICE: <<< Valor RAC normal a levemente aumentada >>>

NOTICE: PACIENTE ESTÁGIO A1 RAC.

NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO >>>

NOTICE: <<< BAIXO RISCO [Estágio 1 TFG e Estágio A1 RAC]>>>

NOTICE: <<< TOMADA DE DECISÃO PARA DOENÇA RENAL CRÔNICA >>>

NOTICE: <<< ACOMPANHAR >>>

NOTICE: O paciente deve ser acompanhado 1 uma vez por ano na própria USF se presença da doença. Refazer exames de rotina para verificação de DRC.



Figura 19. Consulta na função decisão-01

Assim, com todas essas informações, os dados do paciente serão armazenados na tabela TB_HistoricoDRC. A partir dessa tabela os profissionais terão acesso às informações dos pacientes, bem como controle de versões, o que também auxiliará o médico em tomadas de decisão no tocante à compreensão e diagnóstico da DRC. Abaixo, a Figura 20 apresenta a tabela de histórico atualizada após a inclusão da saída de dados das funções.

Saída de Dados		Explain	Mensagens	Histórico						
	nu_sus integer	nu_requisica integer	nu_valortfg integer	nu_estagiortf integer	nu_valorrac integer	nu_estagiора integer	nu_nivelrisco integer	ts_dataexam timestamp w	ts_dataexam timestamp w	nu_diagnosti integer
1	1	1	100	1	10	1	1	2018-12-12 00:	2018-12-12 00:	1

Figura 20. Consulta histórico Paciente-01

Cenário 2: Um paciente se dirigiu até a USF do seu bairro para ser consultado por um médico após persistir dores em suas costas. Ao indicar ao médico seus sintomas, o mesmo gerou uma requisição com as informações do paciente, isto é, armazenou dados como: código da requisição (nu_Requisicao), identificação do médico (nu_CRM), identificação do paciente (nu_SUS), a justificativa da requisição e a data e solicitou os exames de TFG e RAC. O paciente também é portador de doença congênita. Com os resultados em mãos o médico identificou que o exame de TFG do paciente está em 92

(ml/min 1.73m²) e o exame de RAC apontou como resultado 30 ml/g. O médico quer ter certeza se o paciente tem ou não a DRC e seu possível grau de risco.

Inicialmente, para dar suporte a tomada de decisão do médico, foi utilizada a função F_Calcular_Estagio_TFG. Esta é responsável por raciocinar a partir de determinado valor TFG como parâmetro de entrada e retorna como saída o estágio da TFG sozinha. Assim, é notificado a mensagem abaixo junto com o a saída de dados (Figura 21).

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >= 90] >>>
 NOTICE: <<< Valor TFG normal ou elevada >>>
 NOTICE: <<< PACIENTE EM ESTÁGIO 1 TFG >>>

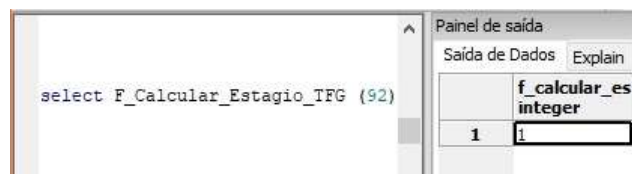


Figura 21. Consulta Calcular Estagio TFG-02

Após identificado o estágio da TFG, a função F_Calcular_Estagio_RAC é solicitada para que seja certificado o estágio da relação albuminúria e creatinúria com base no valor do resultado em exame RAC do paciente. A mensagem abaixo foi gerada junto com a saída de dados (Figura 22) após aplicar esta função.

NOTICE: <<< REFERÊNCIA PARA DIAGNÓSTICO DRC (30 <= valorRAC < 300) >>>
 NOTICE: <<< Valor RAC moderadamente aumentada >>>
 NOTICE: PACIENTE ESTÁGIO A2 RAC.



Figura 22. Consulta Calcular Estagio RAC-02

Com os estágios disjuntos da TFG e RAC, o médico pode verificar com mais exatidão o grau de risco da possível doença antes da tomada de decisão. Assim, a função F_Calcular_GrauDeRisco é acionada para então verificar qual o grau de risco da doença e que procedimentos devem ser executados. Foram emitidos a saída de dados (Figura 23) e mensagens exibidas abaixo.

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >= 90] >>>
 NOTICE: <<< Valor TFG normal ou elevada >>>
 NOTICE: <<< PACIENTE EM ESTÁGIO 1 TFG >>>
 NOTICE: <<< REFERÊNCIA PARA DIAGNÓSTICO DRC (30 <= valorRAC < 300) >>>
 NOTICE: <<< Valor RAC moderadamente aumentada >>>

NOTICE: PACIENTE ESTÁGIO A2 RAC.

NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO >>>

NOTICE: <<< RISCO MODERADAMENTE AUMENTADO [Estágio 1 TFG e Estágio A2 RAC]>>>

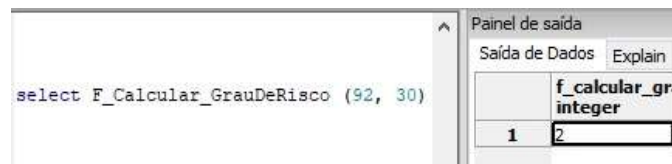


Figura 23. Consulta Calcular Estagio Grau de Risco-02

Obtido o grau de risco da DRC, o médico na USF precisa tomar algumas decisões com base nos resultados obtidos no DB-Ontology. Para isso a função `F_DecisaoDRC` utiliza os dados das funções anteriores, disponibilizando ao médico um conhecimento mais aprofundado acerca dos dados clínicos do paciente. A função de decisão irá auxiliar o médico a monitorar ou encaminhar o paciente para o médico especialista. Abaixo é exibida a saída de dados (Figura 24) e mensagem do banco de dados.

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG >= 90] >>>

NOTICE: <<< Valor TFG normal ou elevada >>>

NOTICE: <<< PACIENTE EM ESTÁGIO 1 TFG >>>

NOTICE: <<< REFERÊNCIA PARA DIAGNÓSTICO DRC (30 <= valorRAC < 300) >>>

NOTICE: <<< Valor RAC moderadamente aumentada >>>

NOTICE: PACIENTE ESTÁGIO A2 RAC.

NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO >>>

NOTICE: <<< RISCO MODERADAMENTE AUMENTADO [Estágio 1 TFG e Estágio A2 RAC]>>>

NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA >>>

NOTICE: <<< MONITORAR >>>

NOTICE: O paciente deve ser acompanhado 1 vezes por ano na própria USF para monitorar a DRC.



Figura 24. Consulta Calcular Decisão-02

Essas funções juntas fazem parte da `F_Ontology`, função responsável por manipular os dados dos pacientes, auxiliar o médico na tomada de decisão e armazenar o conhecimento novo gerado no histórico do Paciente. Assim, após apresentar a melhor decisão, e a mesma ser executada, os dados dispostos ficam armazenados (Figura 25) para possível recuperação por qualquer profissional da área da saúde que dispunha de CRM.

Painel de saída										
Saída de Dados		Explain	Mensagens	Histórico						
	nu_sus integer	nu_requisica integer	nu_valortfg integer	nu_estagiotf integer	nu_valorrac integer	nu_estagiora integer	nu_nivelrisco integer	ts_dataexam timestamp w	ts_dataexam timestamp w	nu_diagnosti integer
1	2	3	92	1	30	2	2	2018-12-10 00:	2018-12-11 00:	2

Figura 25. Consulta Histórico do Paciente-02 após Função F-Ontology

Ao final desses dois cenários é possível observar que embora os pacientes possuam fatores de risco para DRC, as decisões disponibilizadas pelo DB-Ontology mostraram que os principais atributos para diagnóstico da doença são os valores da TFG e RAC, estando o banco em conformidade com o que foi apresentado pela KDIGO. Conforme foi visto no primeiro cenário, embora o paciente seja diabético, este fator de risco não o diagnosticou com DRC aumentada. Além disso, o armazenamento persistente dos dados permite maior segurança e controle aos cuidados em saúde dos pacientes, uma vez que, os médicos da USF podem verificar artefatos passados que possam auxiliar estes profissionais para tomada de decisões futuras.

Cenário 3: Um paciente procura a USF para realizar exames de rotina e verificar o quão saudável está seu corpo. O médico que o atendeu solicitou vários exames para analisar as mais diferentes atividades e órgãos do corpo humano, dentre eles requisitou que o paciente realizasse exames de TFG e RAC. O paciente não apresenta sintomas, não faz parte do grupo de risco e não possui fatores relevantes de DRC. A seguir é apresentado em ordem crescente de execução as mensagem e saída de dados para aplicação do cenário 3.

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG < 15] >>>

NOTICE: <<< FALÊNCIA RENAL >>>

NOTICE: <<< PACIENTE EM ESTÁGIO 5 TFG >>>

Painel de saída	
Saída de Dados	
	f_calcular_es integer
1	6

Figura 26. Consulta Calcular Estagio TFG-03

NOTICE: <<< REFERÊNCIA DIAGNÓSTICO DRC (valorRAC > 300) >>>

NOTICE: <<< Valor RAC severamente aumentada >>

NOTICE: PACIENTE ESTÁGIO A3 RAC.

Painel de saída	
Saída de Dados	
	f_calcular_es text
1	3

Figura 27. Consulta Calcular Estagio RAC-03

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG < 15] >>>
 NOTICE: <<< FALÊNCIA RENAL >>>
 NOTICE: <<< PACIENTE EM ESTÁGIO 5 TFG >>>
 NOTICE: <<< REFERÊNCIA DIAGNÓSTICO DRC (valorRAC > 300) >>>
 NOTICE: <<< Valor RAC severamente aumentada >>>
 NOTICE: PACIENTE ESTÁGIO A3 RAC.
 NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO >>>
 NOTICE: <<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A3 RAC] >>>

The screenshot shows a SQL query editor with the text: `select F_Calcular_GrauDeRisco (7, 320)`. To the right, a 'Painel de saída' (Output Panel) is visible with two tabs: 'Saída de Dados' (Data Output) and 'Explain'. The 'Saída de Dados' tab is active, showing a table with one row and one column. The column is labeled 'f_calcular_gr' with a data type of 'integer'. The row contains the value '5'.

	f_calcular_gr integer
1	5

Figura 28. Consulta Calcular Grau de Risco-03

NOTICE: <<< REFERÊNCIA TFG PARA DIAGNÓSTICO DRC [valorTFG < 15] >>>
 NOTICE: <<< FALÊNCIA RENAL >>>
 NOTICE: <<< PACIENTE EM ESTÁGIO 5 TFG >>>
 NOTICE: <<< REFERÊNCIA DIAGNÓSTICO DRC (valorRAC > 300) >>>
 NOTICE: <<< Valor RAC severamente aumentada >>>
 NOTICE: PACIENTE ESTÁGIO A3 RAC.
 NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA – GRAU DE RISCO >>>
 NOTICE: <<< RISCO MUITO ALTO [Estágio 5 TFG e Estágio A3 RAC] >>>
 NOTICE: <<< DIAGNOSTICO DA DOENÇA RENAL CRÔNICA >>>
 NOTICE: <<< ENCAMINHAR >>>

The screenshot shows a SQL query editor with the text: `select F_DecisaoDRC (7,320,5)`. To the right, a 'Painel de saída' (Output Panel) is visible with two tabs: 'Saída de Dados' (Data Output) and 'Explain'. The 'Saída de Dados' tab is active, showing a table with one row and one column. The column is labeled 'f_decisaodrc' with a data type of 'integer'. The row contains the value '7'.

	f_decisaodrc integer
1	7

Figura 29. Consulta Calcular Decisão-03

Com a aplicação da função F_Ontology no cenário 3 foi possível observar que a ausência de fatores de risco não corrobora significativamente neste cenário para o diagnostico da DRC, uma vez que, sem apresentar sintomas e fatores de risco o paciente foi diagnosticado com grau de risco muito alto.

The screenshot shows a 'Painel de saída' (Output Panel) with a table titled 'Saída de Dados'. The table has 11 columns: 'nu_sus' (integer), 'nu_requisica' (integer), 'nu_valortfg' (integer), 'nu_estagiotfg' (integer), 'nu_valorrac' (integer), 'nu_estagiora' (integer), 'nu_nivelrisco' (integer), 'ts_dataexam' (timestamp w), 'ts_dataexam' (timestamp w), and 'nu_diagnosti' (integer). The first row contains the following values: 1, 5, 4, 7, 6, 320, 3, 5, 2018-12-12 00:, 2018-12-12 00:, 7.

	nu_sus integer	nu_requisica integer	nu_valortfg integer	nu_estagiotfg integer	nu_valorrac integer	nu_estagiora integer	nu_nivelrisco integer	ts_dataexam timestamp w	ts_dataexam timestamp w	nu_diagnosti integer
1	5	4	7	6	320	3	5	2018-12-12 00:	2018-12-12 00:	7

Figura 30. Histórico Paciente-03 após F-Ontology

Além disso, não foram necessários outros exames para identificar o estadiamento da doença. É importante ressaltar que o estágio 6 da TFG no banco DB-Ontology representa o estágio 5 na KDIGO. A representação no banco foi alterada, motivada pelo fato dos estágios da TFG 3a e 3b serem alfanuméricos. Sendo assim, foram transformados em dados numéricos por se tratar de banco de dados relacional. Outro fator importante que deve ser destacado é que a F-Ontology foi construída unicamente para auxiliar o suporte a decisão clínica no aspecto de diagnóstico da DRC, sendo este o foco do estudo.

Os cenários apresentados utilizaram diferentes argumentos para demonstrar o funcionamento corrento, eficiente, rápido, seguro e sem perda de informações, o que nos permite afirmar que a utilização de banco de dados relacional com aplicação de subprogramas é tão vantajosa quanto ao uso de ontologias para os casos estudados, sendo que no DB-Ontology ainda podemos persistir os dados das informações do paciente a qualquer momento, lugar e por qualquer profissional.

4.9 VALIDAÇÃO DA ABORDAGEM

A última etapa deste trabalho compreendeu a validação da abordagem apresentada na seção 6.8, ou seja, utilizou-se os mesmos cenários definidos anteriormente, na ontologia, para verificar a conformidade com a saída dos dados apresentados no DB-Ontology. Desta forma é possível garantir a corretude do raciocínio, de modo a analisar se a abordagem fornece conhecimento verdadeiro aos médicos da Atenção Primária.

Para validação foram considerados os testes nos cenários de aplicação e a utilização das questões de competências definidas. Assim, com o conhecimento adquirido na ontologia utilizada nesta abordagem foi possível executar consultas em DL Querys e compara-las e identificar os benefícios e a veracidade da saída de dados dispostos no DB-Ontology.

Com o intuito de realizar as consultas na OntoDecideDRC, inerentes aos cenários apresentados e validar a abordagem, foram utilizadas as propriedades: “Possui” que está relacionada à condição do paciente, sendo utilizada para representar o resultado dos exames que o paciente possui e para diagnosticar a DRC; e a propriedade “É_Classificada_por” que representa os relacionamentos com foco no estadiamento da DRC, de modo que, os resultados dos exames auxiliem na definição do estágio da doença. Além disso, uma vez diagnosticado com a doença, foram realizadas consultas para identificar o grau de risco da mesma.

No cenário 1, o DB-Ontology apresentou que o paciente estava em Estágio 1 (TFG e RAC), e consequentemente com grau de risco definido como “Risco Ausente”. Executado

a mesma consulta (Figura 31) na ontologia OntoDecideDRC com auxílio da ferramenta Protégè, foi possível obter os mesmos resultados, o que mostra que o mapeamento ontológico executado está em conformidade com as especificações e requisitos, preserva a semântica, não identifica perda de informações e estabelece a persistência dos dados no banco de forma eficiente e segura. Além disso, com base no resultado dos exames de TFG e RAC disponibilizados, é possível definir o estadiamento (Figura 32), que no caso seria Estágio 1.

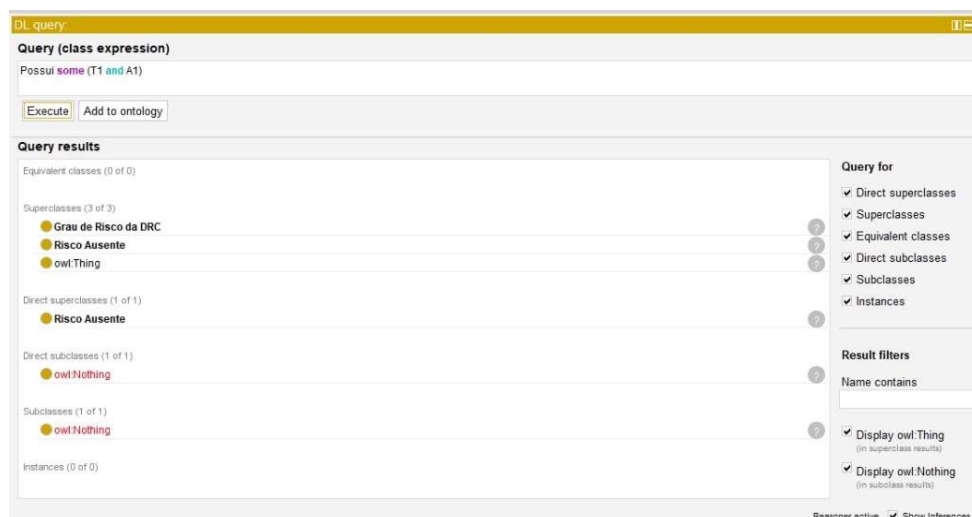


Figura 31. Consulta Grau de Risco Cenário 1



Figura 32. Consulta Estadiamento Cenário 1

No cenário 2, mediante os valores da TFG e RAC o médico da Atenção Primária já indicara que o paciente estava com DRC. Ao verificar junto ao DB-Ontology foi possível confirmar o diagnóstico, bem como demais dados relevantes (grau de risco e estadiamento). A mesma operação foi executada na OntoDecideDRC, com obtenção de mesmas respostas (Figura 33). Isto nos permite afirmar que o banco de dados relacional proposto, após o mapeamento ontológico, funciona de maneira eficiente e nos mesmos parâmetros em nível de comparação. O processo de raciocínio do DB-Ontology é adequado durante as consultas

no banco, e apresenta-se com nível de expressividade equivalente ao apresentado na ontologia. É notório os benefícios das escolhas relacionados ao ambiente de desenvolvimento, pois o conjunto de ferramentas utilizadas durante a abordagem possibilitaram efetuar consultas elaboradas e verificar a pontencialidade do banco de dados através do mecanismo de inferência, ou seja, permitindo analisar a consistência do DB-Ontology, verificando, por exemplo, se o mesmo responde as questões de competência; minimiza a perda de informação; garante a persistência dos dados e mantem a semântica.

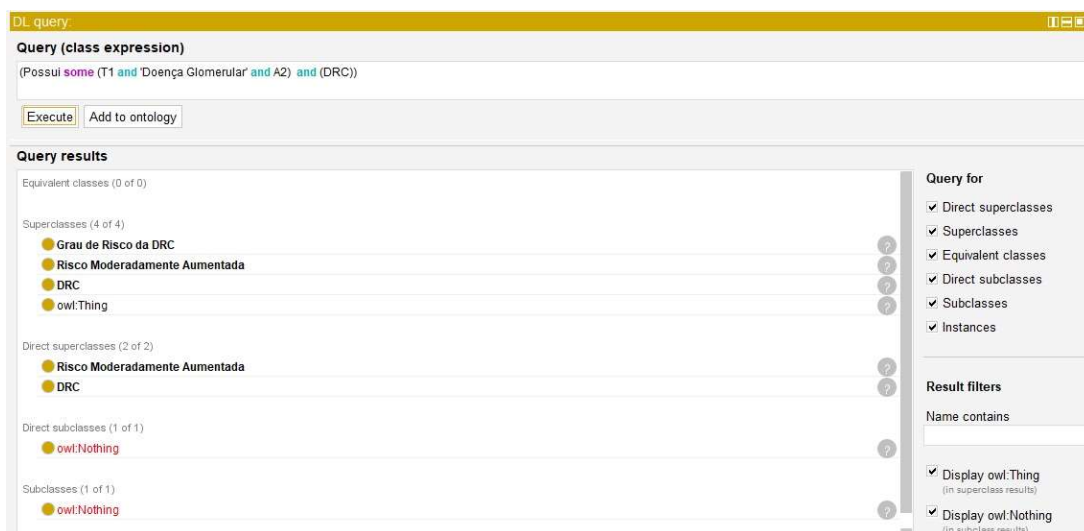


Figura 33. Consulta Diagnóstico e Grau de Risco Cenário 2

No último cenário de aplicação (cenário 3), em conformidade com a saída de dados no DB-Ontology, a OntoDecideDRC apresentou que o paciente possui DRC, está em grau de risco definido como “Risco Muito Alto” e em estágio 5, conforme nas imagens abaixo.

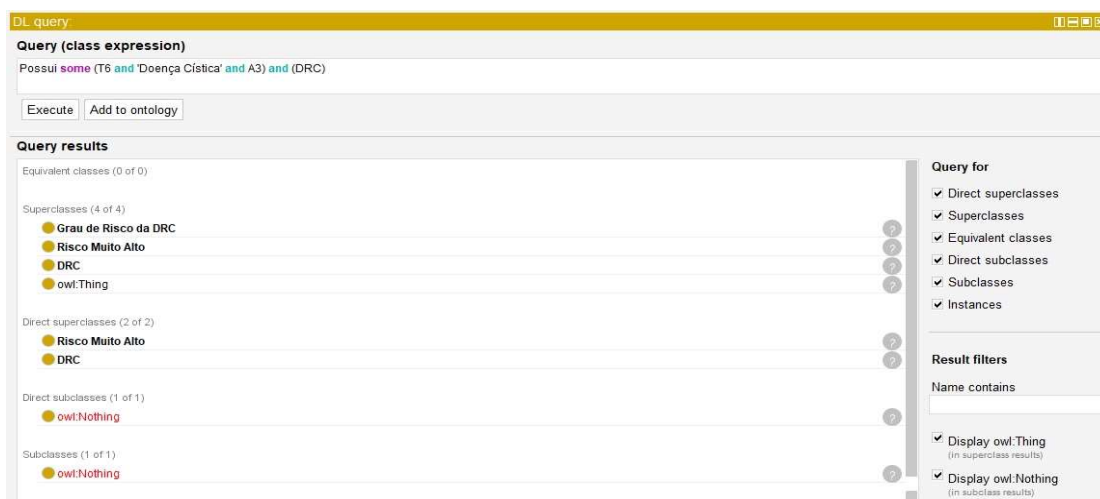


Figura 34. Consulta na Ontologia Referente Cenário 3



Figura 35. Consulta Classificação DRC Cenário 3

A validação da abordagem foi possível partindo do pressuposto de que para caracterizar a completude do mapeamento e consequentemente do banco de dados relacional desenvolvido, os axiomas representados em forma de sentenças nas *Stored Procedures* e definições no DB-Ontology devem ser capazes de representarem as questões de competências, apresentando suas respectivas saídas de dados.

5 TRABALHOS RELACIONADOS

Este Capítulo apresenta os trabalhos relacionados diretamente ou indiretamente em torno do tema em estudo para refinar o conhecimento desejado. Além disso, os trabalhos expostos buscam solucionar problemas semelhantes, embora cada trabalho apresente suas particularidades e domínio específico.

É possível observar, ao longo do capítulo, as diferentes técnicas utilizadas para realizar o mapeamento de banco de dados para ontologias, bem como, suas características, definições, e aspectos de validação. Dividido em duas seções, os Trabalhos Relacionados apresentam as estratégias utilizadas por outros autores para realizar o mapeamento partindo de um banco de dados para ontologia (Seção 5.1); e no sentido inverso, isto é, mapeamento de ontologia para banco de dados (Seção 5.2).

5.1 MAPEAMENTO BANCO DE DADOS - ONTOLOGIA

Motivado pelo número crescente de pesquisas em torno da construção de ontologias a partir de banco de dados relacionais, Mogotlane (2016) investiu seus esforços no mapeamento automático entre ontologias e banco de dados relacionais (BDR). Primeiramente foi aplicado dois plug-ins (DataMaster e OntoBase) Protegé para desenvolver automaticamente ontologias a partir de um BDR. Em seguida as ontologias desenvolvidas foram organizadas e ajustadas aos princípios de mapeamento de banco de dados em ontologia, ou seja, regras para realizar o mapeamento, realizando a correlação da sintaxe do banco de dados com a sintaxe ontológica. A partir da adaptação de um banco de dados Oracle já desenvolvido, foi executado o experimento. Esse tipo de banco de dados, segundo o autor é altamente recomendado para operações da Web Semântica. Além disso, utilizou-se uma ferramenta geradora de documentação estruturada de ontologia (Parrot) para apresentar a estrutura dos códigos ontológicos de saída dos plug-ins. Os resultados apresentados revelam que os plug-ins convertem de maneira satisfatória um BDR para ontologia, porém com pequenos desvios de integridade no mapeamento.

Zarembo (2015) propôs uma esquema para realizar o mapeamento de ontologia para banco de dados relacional. Inicialmente foram identificados as diferenças entre o BDR, e métodos de representação do conhecimento da ontologia. Em seguida são definidas as regras de mapeamento a serem executadas. Por fim, foi desenvolvido um protótipo de

uma ferramenta para validação. O esquema proposto é implementado de forma automatizada e segue cinco passos: modelo de banco de dados é preparado para processamento como metadados XML; utiliza-se XMI para troca de informações via XML; o arquivo XMI é analisado pela ferramenta; aplica-se as regras de mapeamento; e ontologia é salva no formato OWL. Obteve algumas limitações de uso, permitindo aplicação apenas para integração de tecnologia Web Semântica com banco de dados legados; e para mapeamento entre BDR e ontologia, com fins de correspondência ontológica.

Morph (Priyatna, 2014) é uma técnica que permite o mapeamento de banco de dados relacional para ontologia, mais precisamente para representações RDF. Oriundo da linguagem R2RML, a técnica processa os mapeamentos da linguagem. A linguagem R2RML é utilizada em transformações entre banco de dados relacional e conjunto de dados RDF. No Morph, as consultas SPARQL são convertidas em consultas SQL e mapeadas por R2RML para posteriormente serem avaliados pelos BDR. Além disso, os mapeamentos realizados permitem verificar possíveis dados relacionais em modelo RDF.

Liu (2012) apresentou uma abordagem híbrida que permite mapear banco de dados relacionais para ontologia de forma semi-automatizada. Utilizando técnicas combinatórias em nível de elemento e nível estrutural, foi possível extrair a correlação entre BDR e a ontologia. Em seguida, no processo combinatório, aplicou técnicas baseadas em sequência no nível de elemento, e utilizou WordNet para medir a similaridade de nomes entre os pares de elementos de vários tipos. Os experimentos realizados demonstraram seu desempenho favorável no cálculo de similaridade que permitiu o mapeamento do banco de dados relacional para ontologia.

RDOTE (Vavliakis, 2012) é uma estrutura capaz de mapear vários banco de dados relacionais, contendo estruturas normalizadas em diferentes esquemas ontológicos e integrando-os em um único arquivo OWL. Além disso, fornece interfaces gráficas intuitivas, gera RDF automático de dados relacionais, e é compatível com as definições de mapeamento R2RML e D2RQ. Responsável pelo mapeamento manual, o RDOTE permite ao usuário definir todos os relacionamentos entre tabelas, colunas e classes. Embora possua interfaces intuitivas, a expressividade aceitável permite analisar consultas SQL complexas para mapeamentos que não podem ser declarados através de ferramentas apropriadas.

O RDOTE segue algumas etapas para realizar efetivamente o mapeamento do banco de dados relacional para ontologia. Primeiramente é definido um esquema ontológico pertinente de um repositório semântico; em seguida, são definidas as consultas SQL que selecionam as tuplas para serem processadas como gráfico RDF. Posteriormente, as

consultas definidas são interligadas com as classes da ontologia, ou seja, é executado o mapeamento de classes. Esse mapeamento é conectado com outros mapeamentos de classes através da propriedade de objetos ou por consultas SQL em nível de propriedade de tipos de dados. Por fim, a ontologia é instanciada, validada e armazenada em formato de texto ou em repositório persistente. Opcionalmente, em caso de consultas com diversas colunas é possível renomear as cadeias de caracteres.

Bumans (2010) apresenta como os bancos de dados relacionais (BDR) podem ser utilizados para determinar mecanismos entre BDR e ontologia OWL. Com as informações de mapeamento armazenadas em tabelas relacionais, foi possível processá-las utilizando SQL, triplas RDF para instanciar as classes e propriedades OWL. Isto é, um esquema (Figura 36) de ligação entre banco de dados relacionais e ontologias.

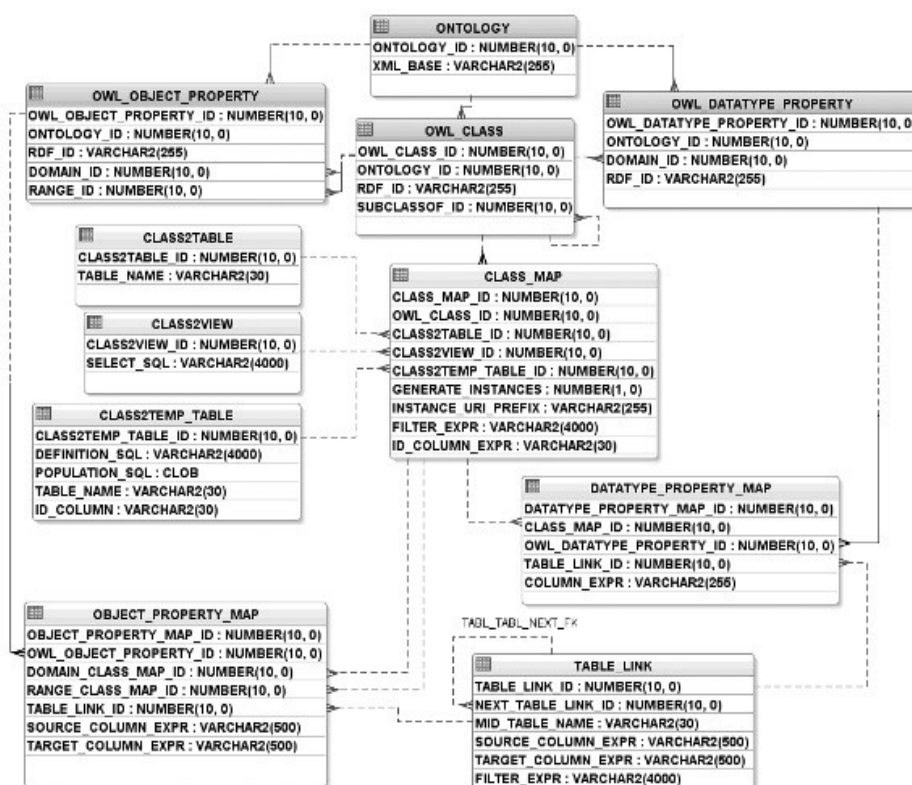


Figura 36. Esquema de Mapeamento Ontologia/Banco de Dados Relacional

Fonte: Bumans (2010).

A proposta utiliza os relacionamentos do BDR para processar informações de mapeamento e gerar sentenças SQL, que ao serem executadas descrevem classes, tipos de dados e propriedades do objeto OWL que satisfazem os dados oriundos do banco de dados relacional. Através da linguagem DB2OWL, o mapeamento é executado automaticamente e se utiliza um banco de dados externo que armazena os resultados da relação do BD e

ontologia original. Dentre as características do framework DB2OWL, destaca-se a reutilização das informações contida nas chaves de tabela do banco de dados para garantir a eficiência do mapeamento. Primeiramente, são identificado casos particulares de tabelas no BD. Após a identificação do casos, as tabelas, colunas e restrições são convertidas para ontologia conforme seus correspondentes (classes, propriedades e relações). Esses dados correspondentes do mapeamento a partir do qual os scripts SQL são gerados, transformam as consultas em nível de instância.

A R2O (Barrasa, 2004) é uma linguagem declarativa para representar mapeamentos complexos entre banco de dados (BD) e ontologias OWL ou RDF. Objetivando extrair conteúdo, a linguagem gera um documento de descrição do mapeamento (Figura 37) automaticamente e utiliza o processador ODEMapster para atualizar os dados do BD para a ontologia.

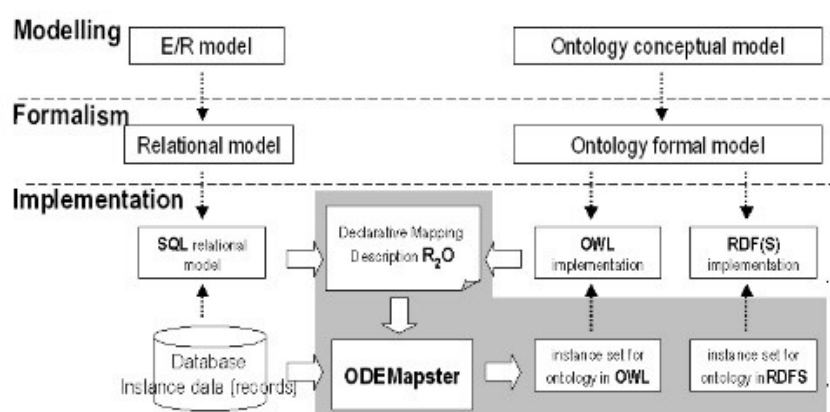


Figura 37. Mapeamento R2O

Fonte: Barrasa (2004).

Baseada em XML, a R2O fornece um conjunto extensível com semântica definida para lidar com os elementos ontológicos (atributos, conceitos e relações) e elementos relacionais (atributos e relações). Além disso, a linguagem considera as diferenças existentes entre banco de dados e ontologia; e que ambos já foram criados. A linguagem destaca-se por sua expressividade e pela independência dos sistemas de gerenciamento de banco de dados (SGBD). Dentre as características da R2O: o fluxo de mapeamento de dados geralmente é do BD para ontologia; inconcistências e ambiguidades podem ser detectadas automaticamente; permite verificar a integridade de parte do BD com ontologia, aplicando axiomas ontológicos; possibilita distribuição dinâmica de consultas.

D2R MAP (Bizer, 2003) é uma linguagem descritiva com o intuito de flexibilizar e

simplificar os mapeamentos complexos entre banco de dados e ontologia sem a necessidade de alterar o BD original. A descrição do mapeamento é baseada em XML. As instruções SQL são aplicadas nas regras de mapeamento, os resultados são agrupados e os dados são mapeados para as instâncias criadas.

O mapeamento do D2R MAP é realizado em quatro etapas, conforme apresentado em Figura 38. Primeiro é selecionado um conjunto de registros para cada classe no banco de dados; em seguida, o conjunto de registros são agrupados; as classes são instanciadas; finalizando com a criação das propriedades das instâncias a partir da ligação dos tipos de dados e objetos.

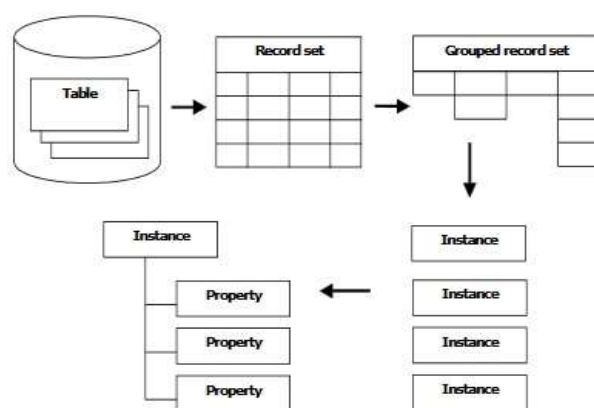


Figura 38. Etapas de Mapeamento D2R MAP

Fonte: Bizer (2003).

5.2 MAPEAMENTO ONTOLOGIA – BANCO DE DADOS RELACIONAL

Afzal (2016) relata a necessidade (em alguns casos) de uma abordagem automática para realizar o mapeamento de ontologias para banco de dados relacionais, de modo a garantir a eficiência das consultas. Assim, sugeriu uma abordagem com esta finalidade, apresentando como resultado a ferramenta OWLMap (Figura 39).

A ferramenta executa automaticamente a abordagem, realizando o mapeamento de ontologia para banco de dados sem perdas em sua transformação. Os experimentos realizados comprovaram a efetividade e rapidez da OWLMap, bem como, demonstra que não há perda de dados, dos tipos de dados e de estrutura. Além disto, foi realizado uma revisão sistemática em torno do tema; e relata resultados de um trabalho anterior, onde conduziu a transformação de conceitos RDF/OWL em BDR e obteve falhas em sua transformação completa.

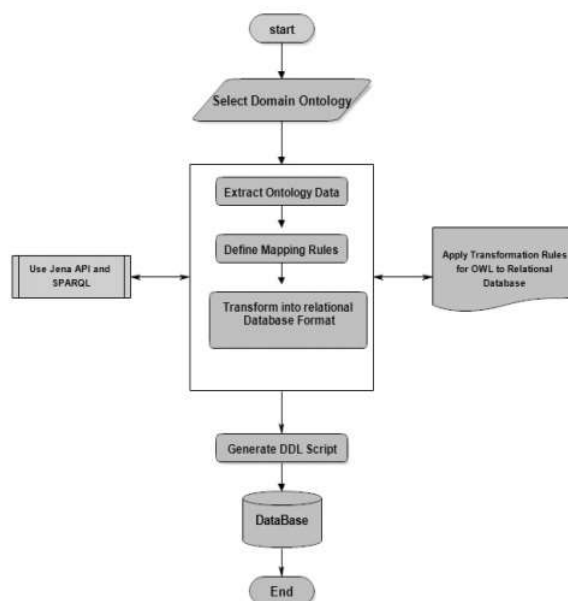


Figura 39. Mapeamento OWLMap

Fonte: Afzal (2016).

Considerando a semântica OWL e do banco de dados relacional, Ernestas (2012) propôs o mapeamento da ontologia para BDR utilizando meta-tabelas. Para garantir a semântica, foram incluídos alguns aspectos que devem ser satisfeitos pela ontologia, como regras de normalização (requisitos), e regras de integridade para consciência dos dados. Para validação foi utilizado o plug-in do Protégé “OWL 2 To BDR”. O desempenho mostrou que a abordagem de forma automática possibilitou transformar ontologia em BDR, porém obteve como desvantagem: a falta de suporte a ontologia existente e a impossibilidade de alterações no banco de dados relacional.

Deise (2011) fornece uma estrutura para armazenar arquivos XML para banco de dados relacional chamado de implementação da ferramenta “X2Rel” e “OntoRel” que fornecem um mecanismo para transformar a ontologia de OWL para banco de dados relacional. Em sua abordagem, eles primeiro geram ontologias a partir de documentos XML com a ajuda de “OntoGen”. Em seguida, mapearam a Ontologia de OWL para modelo relacional implementando a ferramenta OntoRel, aplicando regras de mapeamento. Esta abordagem apresenta como desvantagem a transformação apenas das principais construções OWL, e a falta de um algoritmo adequado para realizar o mapeamento completo destas construções.

Irina (2007) desenvolveu uma abordagem para realizar o mapeamento de ontologia para banco de dados relacional, utilizando a técnica de regras de mapeamento. Conforme Figura 40.

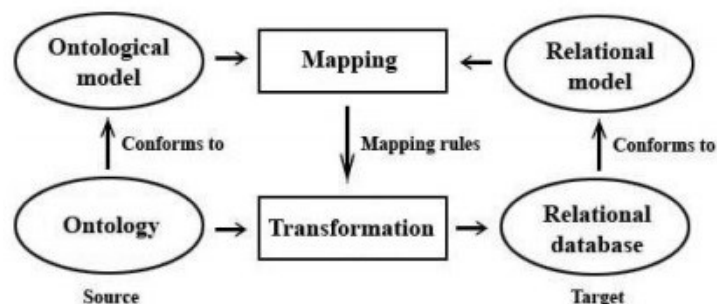


Figura 40. Abordagem com Uso de Regras para Mapeamento

Fonte: Irina (2007).

Essas regras definem o formato para transformar qualquer ontologia OWL em modelo relacional, sendo implementado através da técnica “QUALEG DB”, que executa o mapeamento de forma automática. Primeiramente é analisado um arquivo OWL e em seguida é gerado um Script SQL, que realiza a verificação de consistência e erro. Apesar das motivações apresentadas, a abordagem da autora apresentou desvantagens como desconsideração de restrições de propriedades e perda de sub-propriedades.

Guting (2003) apresenta uma álgebra para representar objetos móveis, os quais são modelados através de tipos de dados espaço-temporais. A álgebra apresentada representa modelos de dados e linguagens de consultas desenvolvidas para construir um sistema de gerenciamento de banco de dados capaz de manipular objetos móveis e permitir elaborar consultas. Após identificar a entidade de interesse e implementar um conjunto de tipos de dados e operações para dar suporte a decisão, foi criado um banco de dados para os objetos móveis, e gerado scripts de consultas visando obter o histórico de informações a partir de funções armazenados (*stored procedures*) que podem ser aplicados em diversos cenários como em mapeamento ontológico.

Tabela 3. Comparativo Abordagens Banco de Dados e Ontologia

Características	Fluxo do Mapeamento	Vantagem	Desvantagem	Alvo/ Operação	Ferramenta
Definição de regras para mapeamento	Banco de Dados - Ontologia	Mapeamento eficiente a partir da utilização de plug-in	Desvio de integridade	Desempenho/ Automático	DataMaster OntoBase (mogotlane, 2016)
Mapeamento de subpropriedade definição de regras de mapeamento	Ontologia – Banco de Dados	Eficiência das consultas; recuperação rápida de dados e esquemas de integração	Não mapeia complementos de classes, comentários e classes enumeradas ou de interseção	Semântica/ Automático	OWLMap (afzal, 2016)
Definição de regras para	Banco de Dados -	Automatização para criação de	Permite aplicação	Integração/ Automático	Não Informado

mapeamento	Ontologia	ontologia a partir de diagrama	apenas para integração de tecnologia Web Semântica com banco de dados legados		Zaremba (2015)
Representação RDF	Banco de Dados - Ontologia	Verifica possíveis dados relacionais em modelo RDF; atualiza dados; traduz consultas	Consultas com níveis de desempenho diferente devido a inclusão de predicados adicionais	Desempenho/Manual	Morph (Priyatra, 2014)
Utilização de meta-tabela; definição de regras de normalização e integridade para consciência dos dados	Ontologia – Banco de Dados	Mapeamento eficiente para otimizar operações de busca e recuperação	Falta de suporte a ontologia existente; impossibilidade de alterações no banco de dados relacional	Desempenho/Automático	OWL 2 To BDR (Ernestas, 2012)
Utilização combinatória de técnicas de nível de elemento e nível de estrutura	Banco de Dados - Ontologia	Analisa pares de elementos correspondentes de vários tipos entre ontologia e banco de dados e calcula a similaridade da estrutura	Perda de informação	Desempenho/Semi-Automático	Não Informado (Liu, 2012)
Mapeia banco de dados com estruturas normalizadas; especifica subconjuntos de dados	Banco de Dados - Ontologia	Compatível com as definições de mapeamento R2RML e D2RQ; possibilita copiar informações reais das tuplas numa série de possíveis formatos	Ausência de chave primária nas tabelas podem desencadear duplicações e falhas no processo de mapeamento	Expressividade/Manual	RDOTE (valvliakis 2012)
Regras de transformação que mapeia a estrutura XML para uma coleção de relações	Ontologia – Banco de Dados	Mecanismo para gerar o esquema relacional de um conjunto de arquivos XML integrados	Mapeia apenas as principais construções de OWL; falta de algoritmo adequado	Não Informado/Semi-Automático	OntoRel (Deise, 2011)
Mapeamento RDF/OWL baseado em correspondência de tabelas gerando instruções SQL e triplos RDF	Banco de Dados - Ontologia	Transformam as consultas em nível de instância	Não permite definir a correspondência entre a ontologia de destino e seus elementos de esquema BDR correspondentes	Semântica/Automático	DB2OWL (Bumans, 2010)

Definição de regras de mapeamento especificadas em nível de modelo	Ontologia – Banco de Dados	Analisa arquivo OWL, gera script SQL e verifica consistência e erro	Desconsidera restrições de propriedades; perda de sub-propriedades	Não Informado/ Automático	Qualeg DB (irina, 2007)
Representação de mapeamento complexos; utiliza linguagem declarativa	Banco de Dados - Ontologia	Detecta inconsistências e ambiguidades; verificar a integridade de parte do BD com consultas dinâmicas	Não compatível com outros bancos de dados para gerar e explorar descrições de mapeamento	Expressividade/ Automático	R2O (Barrasa, 2004)
Linguagem de consultas baseado em <i>stored procedures</i>	Álgebra – Banco de Dados	Gera scripts de consultas para obter histórico de informações; executa operações para suporte a decisão	Resultados mais eficientes em tipos de dados espaço-temporal	Semântica/ Manual	Não Informado (guting 2003)
Define regras de mapeamento	Banco de dados - Ontologia	Flexibiliza e simplifica os mapeamentos complexos	Não executam mapeamentos condicionais com habilidades de transformação mais sofisticados; compatível apenas com BDR que oferece acesso a JDBC e ODBC	Não Informado/ Manual	D2R MAP (Bizer, 2003)

Assim, a visão de diferentes abordagens (Tabela 3) refinam a compreensão e conhecimento desejado em torno do objeto de estudo e auxiliam na comparação com a abordagem apresentada neste trabalho.

5.3 CONCLUSÃO

Esse Capítulo apresentou as principais estruturas para execução do mapeamento entre banco de dados relacional para ontologias. Observa-se (Figura 41) que várias ferramentas, metodologias e algoritmos foram desenvolvidos para suportar a conversão de banco de dados relacionais para ontologias ou vice versa. Além disso, foram identificadas abordagens simples e híbridas (combinação de duas técnicas); pesquisas para transformar diretamente banco de dados relacional em conceitos e descrição OWL e também RDF; definição de regras para mapeamento considerando apenas classes e propriedades. O atual

trabalha considera não apenas as classes e propriedades, mas os axiomas definidos na ontologia, e utiliza regras de mapeamento em conformidade com as fases da modelagem de dados. Isto é, o processo de transformação da ontologia em modelo entidade-relacionamento e em seguida este modelo em BDR.

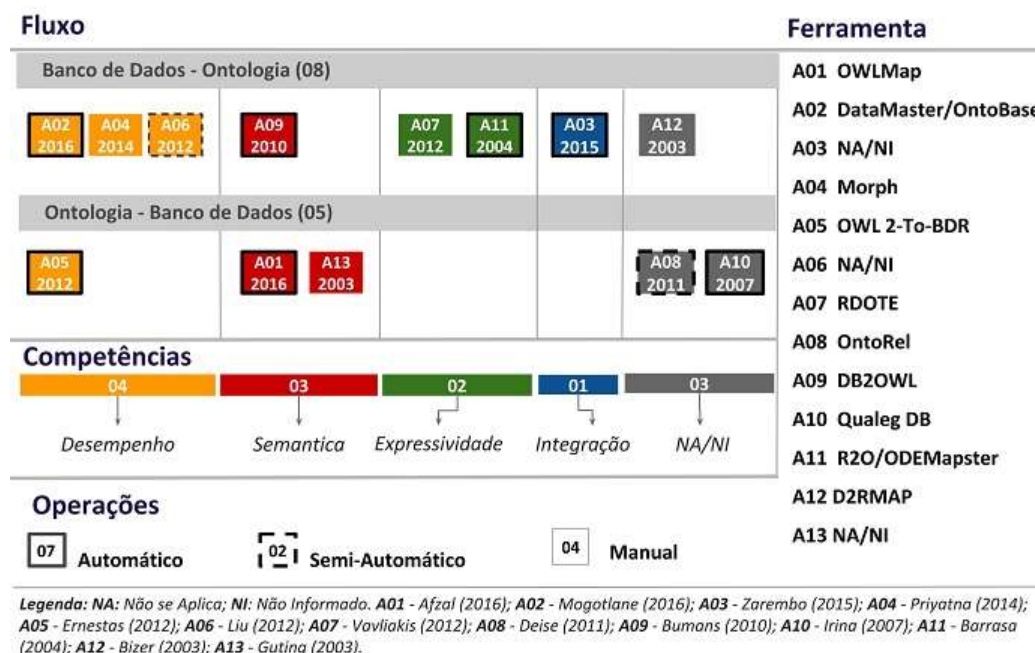


Figura 41. Visão Geral das Abordagens Identificadas

Foram identificadas abordagens com operações manuais, semi-automáticas e automáticas. Este trabalho apresenta uma abordagem manual, uma vez que, embora as abordagens dissertadas neste capítulo sejam definidas como automáticas, estas, em diferentes momentos necessitam de intervenção humana, e tem seu processo de mapeamento incompleto, sem realizar tratamento importante na construção OWL. Além disso, as automatizações das abordagens encadeiam maior esforço para desenvolvimento. Por exemplo, a abordagem com manipulação de plug-ins, demonstrou que as ferramentas e utilitários existentes são de difíceis acesso e necessitam de melhorias.

As abordagens desenvolvidas no processo de mapeamento da ontologia para banco de dados relacional, geralmente utilizam um banco de dados pré-definido ou criado, diferentemente desta abordagem que parte do pressuposto de não existir já um banco de dados, e contrui-lo. Além disso, os trabalhos relacionados restringiram-se em apresentar soluções para características como desempenho, integridade, expressividade e perda de informação de maneira individual, enquanto este trabalho aborda mais de uma dessas características, como perda de informação, expressividade, semântica e integridade.

Embora utilizem a técnica de mapeamento para obter benefícios diante de uma dessas

características, os trabalhos não apresentam a aplicabilidade detalhada em algum domínio destes mapeamentos. Isto é, realizam apenas o mapeamento estrutural, mas não demonstram aplicabilidade em detalhes ou apresentam apenas protótipos como validação. Dentro outras desvantagens, pode-se observar que a maioria dos trabalhos também não dissertam mecanismos para tratar os axiomas e a construção do processo de raciocínio para operações e aplicações ao final do mapeamento. Assim, após executar o mapeado da ontologia para banco de dados relacional, não se observam mecanismos que permitam construir um processo de raciocínio em consultas no BDR em correteza com as *queries* executadas na ontologia.

Assim, observou-se a necessidade de estudos que explorem também os axiomas e o processo de raciocínio da ontologia no banco de dados relacional, de modo que, fossem construídos mecanismos para atuar semelhantemente após o mapeamento ontológico. Uma alternativa observada no trabalho de Gutting (2003) foi o emprego de *stored procedures*, uma vez que, representa modelos de dados e linguagens de consultas desenvolvidos para construir um sistema de banco de dados relacional capaz de manipular objetos, elaborar consultas e garantir a persistência dos dados. O uso de *stored procedures* viabiliza o processamento de todas as informações no servidor do banco de dados, tornando o processamento mais simples para o SGBD do que seria para um Serviço ao enviar a consulta. Isto é, deixando o “servidor web” mais livre para processar de maneira eficiente e rápida as requisições.

Assim, considerando (i) a carência na literatura de abordagens para mapeamento entre ontologia e banco de dados relacional com uso de *stored procedures*; (ii) as desvantagens de determinadas abordagens que indicam vulnerabilidades no processo de mapeamento como: desvio de integridade, perda de informação, falta de suporte a ontologia, incompatibilidade com outros bancos de dados e, desprezo de estruturas como sub-classes e propriedades; e (iii) a caracterização de definição de regras de mapeamento; foi identificado a adequação para utilizar regras de mapeamento e *stored procedures* na construção de uma abordagem para mapeamento ontológico aplicada no domínio biomédico com suporte a decisão.

6 CONSIDERAÇÕES FINAIS

Um dos aspectos principais neste trabalho é lidar com ontologia e suas características de representação do conhecimento e raciocínio de modo que esse conhecimento possa ser armazenado em banco de dados relacional, possibilitando recupera-lo quando necessário para acesso àquelas informações.

Considerando o domínio biomédico, essas informações tornam-se ainda mais valiosas e carentes de armazenamento, principalmente no Sistema Único de Saúde. As limitações ainda enfrentadas com a manipulação de dados clínicos motivaram este trabalho, uma vez que, o suporte a tomada de decisões pode auxiliar médicos da atenção primária na aquisição de conhecimento suficiente para lidar com os cuidados em saúde dos pacientes. Assim, diante desta necessidade, o estudo apresentou o DB-Ontology.

O DB-Ontology surgiu do processo de mapeamento da ontologia OntoDecideDRC para um banco de dados relacional. Foi possível mapear as classes e propriedades da ontologia para tabelas de um SGBD, preservando a semântica da ontologia, e permitindo a persistência dos dados. Além disso, foram mapeados os axiomas e construído o processo de raciocínio no banco de dados utilizando funções armazenados (stored procedures), cujos códigos fontes ficam armazenados no servidor do banco de dados, evitando que esses códigos sejam replicados em cada um dos aplicativos no lado cliente.

Em resumo, foi analisada hierarquicamente a ontologia e, a partir dos axiomas definidos para ontologia voltada para DRC, foi feito o mapeamento para gerar o SGBD. Criado o banco de dados relacional do zero, foram inseridos dados fictícios, com base na tabela definida pela KDIGO, para testar os axiomas da ontologia, propriedades e classes; fazendo ajustes ao uso na Atenção Primária. A definição da OntoDecideDRC como domínio experimental pode ser justificado pelo fato dos axiomas definidos na ontologia estarem congruentes com a tabela da KDIGO para diagnóstico da DRC, que também é seguido pelas diretrizes do Ministério da Saúde do Brasil; e pelo fato da ontologia se aproximar da estrutura do banco de dados. Além disso, Tavares (2016) apresenta o ciclo completo do diagnóstico, estadiamento e encaminhamento, contribuindo para identificação do momento de ruptura da troca de informações entre profissionais da atenção primária e especialistas em nefrologia, permitindo uma visão olística nos dois sentidos do fluxo de informações.

A abordagem apresentada pode auxiliar e aprimorar o cuidado em saúde dos pacientes de DRC, a partir da gestão do conhecimento no ambiente da atenção primária. Além disso,

atua como mecanismo estratégico a ser utilizado por profissionais da área da saúde, pois, permite maximizar o poder decisório e pode facilmente ser adaptado para diferentes cenários dentro da realidade na USF. Essa adaptação pode ampliar o processo organizacional e desenvolver mecanismos cada vez mais eficientes e próximo da realidade que objetivem beneficiar os pacientes.

De uma forma geral a principal contribuição apresentada é uma abordagem capaz de executar os mesmos mecanismos que uma ontologia sendo que no próprio banco de dados relacional, com uso de subprogramas. Como pôde ser observado nos cenários de aplicação, o DB-Ontology foi capaz de (a partir dos parametros de entradas definidos e baseando-se nas diretrizes propostas pela KDIGO) diagnosticar a DRC com exatidão; sem perda de desempenho, sem falha de integridade, preservando a semântica da ontologia e garantindo a persistência dos dados.

O mapeamento da informação ontológica em banco de dados relacional facilita, várias operações, como busca e recuperação de informações. Além disso, é importante ressaltar que o DB-Ontology garante a persistência de dados, uma vez que, as *stored procedures* incorpora à SQL características procedurais, como os benefícios e facilidades de controle de fluxo de programas que as melhores linguagens possuem. Dessa forma, o desenvolvimento em PL/pgSQL significa ter a disposição um ambiente procedural totalmente desenvolvido para aplicações de banco de dados, favorecendo-se do controle transacional inerente das aplicações deste tipo.

Dentre outras contribuições esperadas pode-se citar:

- Amplitude do conhecimento do mundo real das unidades de saúde da família;
- Desenvolvimento do fluxo de modelagem de dados relacional;
- Obtenção eficiente de tomadas de decisões;
- Execução ágil de consultas SQL avançadas com uso de funções;
- Construção de processo de raciocínio no banco de dados relacional em correteude com a ontologia.

6.1 LIMITAÇÕES DA PESQUISA

Embora a literatura disponha de um referencial teórico robusto em torno do tema, permitindo a elaborada e execução das etapas de desenvolvimento da abordagem, ainda existem algumas limitações da pesquisa.

Assim, é importante ressaltar que o escopo desta abordagem foi restrito ao

procedimento de diagnóstico e estadiamento da DRC, de modo que, foram construídos mecanismos de raciocínio que estivessem atrelados apenas a esses tipos de procedimentos. Isto é, foi restrito a realização de testes inerentes ao processo de raciocínio a estes procedimentos, direcionando para trabalhos futuros a necessidade de uma avaliação mais abrangente, indicando que o raciocínio para outros procedimentos da ontologia também possa ser executado no banco de dados relacional. As questões temporais fragilizaram o desenvolvimento de testes para garantir que todos os mecanismos de raciocínio da ontologia fossem suportados no DB-Ontology.

Em relação aos dados, deve-se considerar as dificuldades para aquisição de variáveis estratégicas e reais, uma vez que, a burocratização existente na área de saúde ainda limita o acesso a dados clínicos para pesquisas científicas, o que justificou a utilização de dados fictícios para aplicação do estudo

6.2 TRABALHOS FUTUROS

Como trabalhos futuros, pretende-se construir um conjunto de funções que atendam não somente o processo de suporte a decisão para diagnóstico da DRC aos médicos da atenção primária, mas que sirvam como mecanismo de auxílio a médicos especialista em nefrologia; e também absorvam outros domínios de aplicação. Além disso, desenvolver mecanismos para integrar o DB-Ontology a uma arquitetura baseada no paradigma do Conhecimento como Serviço (KaaS), que seja capaz de gerenciar múltiplas fontes de dados e modelos de conhecimento, centralizando o acesso por meio de uma API facilmente adaptável.

6.3 PUBLICAÇÕES

- Artigo Completo: The Brazilian Challenge to Accessibility and Digital Inclusion for People With Autistic Spectrum Disorders – XVII Brazilian Symposium on Human Factors in Computing Systems.
Disponível em: <https://dl.acm.org/citation.cfm?id=3274192&picked=prox>
- Artigo Completo: Panorama das Publicações Nacionais sobre Autismo, Educação e Tecnologia – XXIX Simpósio Brasileiro de Informática na Educação. Disponível nos Anais do SBIE 2018: <https://www.brie.org/pub/index.php/sbie/issue/view/180>

Referências

ALMEIDA, M. B. e BAX, M. P. (2003). Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **Ci. Inf.**, Brasília.

ALVES, C. M. O., COTA, M. P. Multi-level visualization in decision support systems: Literature review. **IEEE**, Caceres, Spain. 2018.

AFZAL, H., WAQAS, M. e NAZ, T. OWLMap: Fully Automatic Mapping Of Ontology into Relational Database Schema. **The Science and Information Organization**, v. 7, n. 11, 2016.

BARRASA, J. et al. R 2 O, an extensible and semantically based database-to-ontology mapping language. In **Proceedings of the 2. Workshop on Semantic Web and Databases**, Toronto, (Can) Aug. 2004.

BRASIL. **Ministério da Saúde. Secretaria de Atenção à Saúde. Departamento de Atenção Especializada e Temática.** Diretrizes Clínicas para o Cuidado ao paciente com Doença Renal Crônica – DRC no Sistema Único de Saúde/ Ministério da Saúde. Secretaria de Atenção à Saúde. Departamento de Atenção Especializada e Temática. – Brasília: Ministério da Saúde, 2014. p.: 37 p.: il. ISBN 1. Doença Renal Crônica.

BRASIL, W. **Semantic web.** w3c.br/Padroes/WebSemantica, acesso em abril de 2018, 2011.

BIZER, C. D2RMap - a database to rdf mapping language. In **12th Intl World Wide Web Conference Budapest**, 2003.

BORDOLOI, Prantik; ISLAM, Nazrul. Knowledge management practices and healthcare delivery: a contingency framework. **The Electronic Journal of Knowledge Management**, v. 10, n. 2, p. 110-120, 2012.

BUMANS, G. Mapping Between Relational Databases and OWL Ontologies: An Example. **Computer Science and Information Technologies**, v. 756, p. 99-117, 2010.

CHAO, T.E.; MODY, G.N. The impact of intellectual property regulation on global medical technology innovation. **BMJ Innov**; 1:49–50, 2015.

DACONTA, M. C.; OBRST, L. J.; SMITH, K. T. The Semantic Web: a guide to the

Future of XML, Web Services, and Knowledge Management. Indianápolis: **Wiley Publishing**, 2003.

DARWEN, H., The Relational Model: Beginning of an Era. **IEEE Annals of the History of Computing**, v. 34, n. 4, p. 7-8, 2012.

DEISE, D.B.S., TOBIAS D.C.A. e EDUARDO K.P, Mapping OWL Ontologies To Relational schemas, In **proceeding of Information Reuse and Integration (IRI), IEEE International Conference**, (2011).

ERNESTAS, V. e LINA, N. E. Preserving Semantics of Owl 2 Ontologies in Relational Databases Using Hybrid Approach, Information Technology and Control, **IEEE Annals of the History of Computing**, v. 34, n. 4, p. 38-48, 2012.

FRADE, S.; FREIRE, S. M.; SUNDVALL, E.; PATRIARCA-ALMEIDA, J.; CRUZCORREIA, R. Survey of openehr storage implementations. In: **Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems**, 2013.

GREENES, Robert A; MIDDLETON, Blackford.; MUSEN, Mark A. Clinical decision-support systems. In: **Biomedical informatics**. Springer, London, 2014. p. 643-674.

GOMES, C. N. A. P. ONTODRC Rompendo o Silêncio da Doença Renal Crônica. 188 f.Monografia (Doutorado) - **Curso de Modelo de Decisão e Saúde**, Universidade Federal Da Paraíba, João Pessoa, 2018.

GUARINO, N. Formal ontology in information systems: **Proceedings of the first international conference (FOIS'98)**, June 6-8, Trento, Italy. [S.l.]: IOS press, 1998.

GUTING, R.H. , M.H. BOHLEN, M. ERWIG, C.S. JENSEN, N.A. LORENTZOS, E. NARDELLI, M. SCHNEIDER, and J.R. RIOS VIQUEIRA, Spatio-Temporal Models and Languages: An Approach Based on Data Types In: Koubarakis et al. (eds.), Spatio-Temporal Databases: **The CHOROCHRONOS Approach**, Springer-Verlag, 2003, 117-176.

GRUBER, Thomas R..Toward Principles for the Design of Ontologies Used for (1993).

HARTUNG M; GROß, A. e RAHM, E (2013). COnto-Diff: Generation of complex evolution mappings for life science ontologies. **Journal of Biomedical Informatics**. v(46), p. 15-32, 2013.

HEINZLE, R., GAUTHIER, F. A. O. e FIALHO, F. A. P. Semantic in Decision Support

Systems: **The State of the Art. Revista Unifebe**, v.1, n.8, 2010.

HENRIQUE, T.; FADIGAS, I. d. S.; ROSA, M. G.; PEREIRA, H. B. d. B. Mathematics education semantic networks. **Social Network Analysis and Mining**, Springer Vienna, v. 4, n. 1, 2014. ISSN 1869-5450.

HUMAIRA. A.; TABASSUM, N.; AYESHA, S. A Survey on Automatic Mapping of Ontology to Relational Database Schema. **Research Joraul of Recent Sciences**, v. 4, p. 66-70, 2015.

IRINA, A., KORDA. N. e KALJA, A. Storing OWL Ontologies in SQL Relational Databases. World Academy of Science, Engineering and Technology International **Journal of Computer, Electrical, Automation, Control and Information Engineering** Vol:1, No:5, 2007.

ISOTANI, S., BITTENCOURT, C. e IG, IBERT. Dados Abertos Conectados. Núcleo de Informática e Coordenação do Ponto Br – NIC.br, **Centro de Estudos sobre Tecnologia Web** – CeWeb.br.[http://http://ceweb.br/livros/dados-abertos-conectados/capitulo-3/#sh3.3](http://ceweb.br/livros/dados-abertos-conectados/capitulo-3/#sh3.3). Acesso em dezembro de 2018.

KDIGO, Kidney Disease: Improving Global Outcomes (KDIGO) CKD Work Group. **KDIGO 2012 Clinical Practice Guideline for the Evaluation and Management of Chronic Kidney Disease**. Kidney inter., Suppl. 2013; 3: 1-150.

KIRSZTAJN, G. M., FILHO, N. S., DRAIBE, S. A., NETTO, M. V. P., THOMÉ, F. S., SOUZA, E., BASTOS, M. G. Fast Reading Of The KDIGO 2012: Guidelines For Evaluation And Management Of Chronic Kidney Disease In **Clinical Practice. J. Bras Nefrol**, 36 (1), pp. 63-73, 2014.

LADEIRA, Marcelo. Representação de Conhecimento e Redes de Decisão.1997. 150 f.Monografia (Doutorado) - **Curso de Ciência da Computação**, Universidade Federal Do Rio Grande Do Sul, Porto Alegre, 1997.

LAGE, M. C. Utilização do software NVivo em pesquisa qualitativa: uma experiência em EaD. **Educação Temática Digital**, São Paulo. p. 198 – 226, 2011.

LAUDON, K.C.; LAUDON, J.P. Gerenciamento de Sistemas de Informação. 3 ed. Rio de Janeiro: **LTC**, p. 433. 2001.

LI, C. Extending SQL Functions - Project in Database Management Winter. **Department of Computer Science**. UC Irvine, 2007.

LIU, H.; NING, H.; WANG, T. A Hybrid Approach of Mapping Relational Database

Schema to Ontology. **International Conference on Computer Science and Service System**, 2012.

LOAN, T. T. H., CHI, P. T. T. e QUANG, H. Na Approach of Transforming Ontologies Into Relational Databases. **Intelligent Information and Database Systems**, pp 149-158, 2015.

MAEYAMA, M. A., JASPER, C. H., NILSON, L. G., DOLNY, L. L. and CUTOLO, L. R. A. Health promotion as technology for social transformation. **Revista Brasileira de Tecnologias Sociais**, v.2, n.2, p. 129-143 2015.

MANOVICH, LEV. The Database. **Revista Eco Pós – Arte, Tecnologia e Mediação**, v. 8, n. 1, 2015.

MARAGOUidakis, M.; MAGLOGIANNIS, I.; LYMBEROPOULOS, D. **A Medical, Description Logic based, Ontology for Skin Lesion Images**. Conference on Bio-Informatics and Bio-Engineering 1 – 6. Grécia, Outubro, 2008.

MCGUINNESS, D. L.; VAN HARMELEN, F. OWL web ontology language overview. W3C recommendation, **W3C**, Feb. 2008.

MOGOTLANE, K. D. e DOMBEU, V. F. Automatic Conversion Of Relational Databases Into Ontologies: A Comparative Analysis Of Protégé Plug-Ins Performances. **Internacional Journal of Web & Semantic Technology (IJWest)**, vol. 7, n. ¾, 2016.

MORAES, J. L. C, SOUZA, W. L, PIRES, L. F, CAVALINI, L. C. e PRADO, A. F. “An Approach to Developing Applications in the Pervasive Healthcare Environment Through the Use of Archetypes”, 2013.

MOTA, M. R. A. Mapeamento sistemático sobre o uso de ontologias em informática médica. 132 f. **Dissertação (Mestrado)**. Universidade Federal da Paraíba, João Pessoa, 2013.

NEUMANN, T.; MUHLBAUER, T.; KEMPER, A. Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems. **ACM SIGMOD International Conference on Management of Data**, p. 677-689, 2015.

NOY, NATALYA F.; MCGUINNES, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. **Stanford University**, Stanford, 2012.

OKPECHI, Ikechi G. et al. Global capacity for clinical research in nephrology: a survey by the International Society of Nephrology. **Kidney International Supplements**, v. 8, p. 80-82, 2018.

PRIYATNA, F.; CORCHO, O.; SEQUEDA, J. Formalisation and experiences of

r2rmlbased sparql to sql query translation using morph. In: **Proceedings of the 23rd International Conference on World Wide Web**, p. 479–490, Republic and Canton of Geneva, Switzerland, 2014.

RAMALHO, R. A. S. Representação do Conhecimento e Ontologias: Reflexões Interdisciplinares. **Encontro Nacional de Pesquisa em Ciência da Informação**, 2008.

RUSSELL, S.; NORVING, P. Inteligência Artificial. **Elsevier**, 2. Ed.: Rio de Janeiro, 2009.

SAHOO, S. S.; HALB, W.; HELLMANN, S.; IDEHEN, K.; JR, T. T.; AUER, S.; SEQUEDA, J.; EZZAT, A. **A survey of current approaches for mapping of relational databases to rdf**, 01, 2009.

SARKAR, I. N. **Biomedical informatics and translational medicine**. Journal of Translational Medicine, 2010. p. 1 – 12.

SERGIO, M. C., SILVA, T. N. GONÇALVES, A. L. Representação do Domínio de Conhecimento Acadêmico e Uma Teoria de Representação: A Ontologia de Fundamentação Unificada. **International Journal of Knowledge Engineering and Management**, v. 5, n. 11, p. 51-69, 2016.

SILACHAN, K.; TANTATSANAWONG, P. Domain Ontology Health Informatics Service From Text Medical Data Classification. Department of Computing, Faculty of Science, Silpakorn University. Nakornpratom, Thailand. **Annual SRII Global Conference**. 2011.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: **Elsevier**, 2006.

SMITH, B. **Ontology Blackwell. Guide to the Philosophy of Computing and Information**, Oxford, 2003.

SOCIEDADE BRASILEIRA DE NEFROLOGIA (SBN). Censo Brasileiro de Diálise: 1999-2015. São Paulo: **SBN**; 2015 [citado em 2017 Jul 4]. Disponível em: [www.sbn.org.brhttps://sbn.org.br/o-censo-2017-foi-publicado-confira/](https://sbn.org.br/o-censo-2017-foi-publicado-confira/)

SOUZA, C. A. Uma Metodologia para Construção de Redes Bayesianas com Base em Ontologias de Domínio na Área da Saúde para Suporte à Decisão Clínica. 131 f. Monografia (Mestrado) - **Curso de Informática**, Universidade Federal Da Paraíba, João Pessoa, 2018.

TAVARES, ESAÚ A. Uma abordagem para suporte à decisão clínica baseada em semântica o domínio da nefrologia. 2016. 211 f. **Dissertação (Mestrado)**. Universidade

Federal da Paraíba, João Pessoa. 2016.

UIC – University of Illinois at Chicago – Health Informatics and Health Information Management. Pesquisa geral no site. **UIC**, 2011. Disponível em: <<http://healthinformatics.uic.edu/history-of-health-informatics>>. Acesso em: 15 nov. 2018.

VAVLIAKIS, K. N.; GROLLIOS, T. K.; MITKAS, P. A. Rdote - publishing relational databases into the semantic web. **Journal of Systems and Software**, 86(1):89–99, 2013.

VIANNA, J. L.; ATAIDE, C. A.; FERREIRA, C. M. Educação a Distância no Brasil: Cotidiano, Prática, Avanços e Perspectivas. **Encontro Internacional de Formação de Professores**, v.8, n.1 (2015).

WADE, B.; CHAMBERLIN, D.; IBM Relational Database Systems: The Early Years.

WANZELLER, D. A. P. Investigando o Uso de Banco de Dados Orientados a Documentos para Gerenciar Informações da Administração Pública. **Monografia de Graduação**. Universidade Federal de Brasília, 2013.

ZAREMBO, I. Automatic Transformation of Relational Database Schema into OWL Ontologies. In 10th **International Scientific and Practical Conference**. Volume III, 217-222, 2015.

APÊNDICE A – Inserção de dados no DB-Ontology

- **Inserção de dados na TB_Paciente**

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

```
VALUES (1, 'Erick Sutilly', 'R. Projetada, 250', '1940-09-12',1);
```

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

```
VALUES (2, 'Juan Huck', 'Av. Brasília, 10', '1980-10-12', 1);
```

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

```
VALUES (3, 'Simon Durin', 'Av. Rebouças, 180', '1991-11-08', 2);
```

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

```
VALUES (4, 'Willian Santini', 'R. da Gamileira, 87', '1993-05-07', 2);
```

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

```
VALUES (5, 'Bruno Muller', 'R. dos Eucarios, 100', '1970-12-11', 1);
```

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

```
VALUES (6, 'Suraia Rezente', 'Av. Epitacio Pessoa, s/n', '1985-05-05', 1);
```

```
INSERT INTO TB_Paciente (nu_SUS, nm_Paciente, nm_Endereco, ts_DataNasc, nu_CorPele)
```

- **Inserção de dados na TB_Medico**

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (1, 'Vilma Cruz ');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (2, 'Marta Sampaio');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (3, 'Roberta Aléria');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (4, 'Rebeca Sata');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (5, 'Ricardo Mendes');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (6, 'Marcelo Santa');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (7, 'Joaquim Silva');
```

```
INSERT INTO TB_Medico (nu_CRM, nm_Medico) VALUES (8, 'Tobias Rabelo')
```

- **Inserção de dados na TB_Requisicao**

```
INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa, ts_DataRequisicao)
```

```
VALUES (1,1,1,'Dor no meio das costas','2018-11-19');
```

```
INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa, ts_DataRequisicao)
```

```
VALUES (2,1,1,'Sintomas urêmicos','2018-11-18');
```

```
INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa, ts_DataRequisicao)
```

```
VALUES (3,2,2,'Sangramentos','2018-11-17');
INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa,
ts_DataRequisicao)
VALUES (4,2,6,'Edema agudo de pulmão','2018-11-16');
INSERT INTO TB_Requisicao (nu_Requisicao, nu_CRM, nu_SUS, nm_Justificativa,
ts_DataRequisicao)
VALUES (5,3,7,'Exame de rotina','2018-11-15');
```

- **Inserção de dados na TB_Exame**

```
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (1, 'Taxa de Filtração Glomerular (TFG)');
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (2, 'Relação Albuminúria Creatinúria (RAC)');
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (3, 'Exame Sumário de Urina');
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (4, 'Aletação de Imagem');
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (5, 'Creatinina');
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (6, 'Proteinúria');
INSERT INTO TB_Exame (nu_Exame, nm_Exame)
VALUES (7, 'Albuminúria');
```

- **Inserção de dados na TB_FatorDeRisco**

```
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (1, 'Doença','D', 'Doença Túbulo Intersticial','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (2, 'Doença','D', 'Doença Congênita','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (3, 'Grupo de Risco','D', 'Diabetes','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (4, 'Doença','D', 'Doença Vascular','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (5, 'Doença','D', 'Doença Cística','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (6, 'Grupo de Risco','G', 'Obesidade','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
```

```

nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (7, 'Grupo de Risco','G', 'Tabagismo','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (8, 'Grupo de Risco','G', 'Diabetes','U','Unico');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (9, 'Sintomas','S', 'Problemas Urinários','P', 'Urina Espumosa');
INSERT INTO TB_FatorDeRisco (nu_FatorRisco, nm_FatorRisco, nu_fl_FatorRisco1,
nm_fl_FatorRisco1, nu_fl_FatorRisco2, nm_fl_FatorRisco2)
VALUES (10, 'Sintomas','S', 'Edema','E', 'Inchaço nos Pés');

```

- **Inserção de dados na tabela TB_Medicamento**

```

INSERT INTO TB_Medicamento (nu_Medicamento, nm_Medicamento, nm_Descricao)
VALUES (1, 'Anti-Hipertensivo','2x após refeição');
INSERT INTO TB_Medicamento (nu_Medicamento, nm_Medicamento, nm_Descricao)
VALUES (2, 'Anti-Hipertensivo','2x após refeição');
INSERT INTO TB_Medicamento (nu_Medicamento, nm_Medicamento, nm_Descricao)
VALUES (3, 'Medicamento para diabetes','1x ao dia');
INSERT INTO TB_Medicamento (nu_Medicamento, nm_Medicamento, nm_Descricao)
VALUES (4, 'Analgésico','3x ao dia')

```

- **Inserção de dados na TB_CorPele**

```

INSERT INTO TB_CorPele (nu_Cor, nm_Cor) VALUES (1,'Branca');
INSERT INTO TB_CorPele (nu_Cor, nm_Cor) VALUES (2,'Parda');
INSERT INTO TB_CorPele (nu_Cor, nm_Cor) VALUES (3,'Afro-Americana')

```

- **Inserção de dados na TB_ExameRequisitado**

```

INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (1,1,'2018-12-12');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (1,2,'2018-12-12');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (2,1,'2018-12-11');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (2,2,'2018-12-12');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (2,3,'2018-12-11');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (3,1,'2018-12-10');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (3,2,'2018-12-11');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (4,1,'2018-12-12');
INSERT INTO TB_ExameRequisitado (nu_Requisicao, nu_Exame, ts_Data)
VALUES (4,2,'2018-12-12')

```

- **Inserção de dados na tabela TB_FatorRequisicao**

```
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (0,4);
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (1,8);
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (2,1);
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (3,3);
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (4,5);
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (4,1);
INSERT INTO TB_FatorRequisicao (nu_FatorRisco, nu_Requisicao) VALUES (3,2)
```

- **Inserção de dados na tabela TB_MedicamentoRequisicao**

```
INSERT INTO TB_MedicamentoRequisicao (nu_Medicamento, nu_Requisicao)
VALUES (1,1);
INSERT INTO TB_MedicamentoRequisicao (nu_Medicamento, nu_Requisicao)
VALUES (2,2);
INSERT INTO TB_MedicamentoRequisicao (nu_Medicamento, nu_Requisicao)
VALUES (3,2);
INSERT INTO TB_MedicamentoRequisicao (nu_Medicamento, nu_Requisicao)
VALUES (4,3);
INSERT INTO TB_MedicamentoRequisicao (nu_Medicamento, nu_Requisicao)
VALUES (3,3)
```

- **Inserção de dados na TB_HistoricoDRC**

```
INSERT INTO TB_HistoricoDRC (nu_SUS, nu_Requisicao, nu_ValorTFG,
nu_EstagioTFG, nu_valorRAC, nu_EstagioRAC, nu_NivelRisco,ts_DataExameTFG,
ts_DataExameRAC, nu_Diagnostico)
VALUES (1,1,100,1,25,1,1,'2018-12-12','2018-12-12',1);
INSERT INTO TB_HistoricoDRC (nu_SUS, nu_Requisicao, nu_ValorTFG,
nu_EstagioTFG, nu_valorRAC, nu_EstagioRAC, nu_NivelRisco,ts_DataExameTFG,
ts_DataExameRAC, nu_Diagnostico)
VALUES (2,3,50,3,500,3,4,'2018-12-10','2018-12-11',6)
```