

# NefroService:

## Plataforma baseada em conhecimento como serviço no domínio da nefrologia

Renan Gomes Barreto



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2017

Renan Gomes Barreto

# NefroService: Plataforma baseada em conhecimento como serviço no domínio da nefrologia

Monografia apresentada ao curso de Graduação em Ciência da Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Natasha Correia Queiroz  
Lino

Maio de 2017

Ficha Catalográfica elaborada por  
Rogério Ferreira Marques CRB15/690

B273n

Barreto, Renan Gomes.

Nefroservice: plataforma baseada em conhecimento como serviço no domínio da nefrologia / Renan Gomes Barretohfh. – João Pessoa, 2017. 44p. : il.

Monografia (Bacharelado em Ciência da Computação) – Universidade Federal da Paraíba - UFPB.

Orientador: Prof<sup>o</sup>. Dra. Natasha Correia Queiroz Lino.

1. Software. 2. Nefrologia. 3. Ontologia. 3. Plataforma de saúde. I. Título.

UFPB/BSCI

CDU: 004.62 (043.2)



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

**ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO**

Aos **05** dias do mês de **Dezembro** de **2016**, às **10:00** horas, em sessão pública em sala do Campus I da Universidade Federal da Paraíba, na presença da banca examinadora presidida pelo professor(a) orientador **Prof. Dr.ª. Natasha Correia Queiroz Lino** e pelos professores **Prof. Dr. Gustavo Henrique Matos Bezerra Motta** e **Prof. Dr. Clairton de Albuquerque Siebra**, o(a) aluno(a) **Renan Gomes Barreto**, apresentou o trabalho de conclusão de curso intitulado: **NefroService: Plataforma baseada em conhecimento como serviço no domínio da nefrologia** como requisito curricular indispensável para a integralização do Curso de **Ciência da Computação**

Após a exposição oral, o(a) candidato(a) foi arguido(a) pelos componentes da banca que reuniram-se reservadamente, e decidiram, APROVAR a monografia, com nota 10,0. Divulgando o resultado formalmente ao aluno e demais presentes, eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

*Natasha*

Prof. Dr.ª. **Natasha Correia Queiroz Lino**

*Gustavo Henrique Matos Bezerra Motta*

Prof. Dr. **Gustavo Henrique Matos Bezerra Motta**

*Clairton de Albuquerque Siebra*

Prof. Dr. **Clairton de Albuquerque Siebra**

*Renan Gomes Barreto*

**Renan Gomes Barreto**

*“A única maneira de fazer um bom trabalho é amando o que você faz.  
Se você ainda não encontrou, continue procurando. Não se desespere.  
Assim como no amor, você saberá quando tiver encontrado.”*

*Steve Jobs*

## **DEDICATÓRIA**

Dedico este trabalho a minha querida esposa Gabriela e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

## RESUMO

Devido a necessidade de melhorar o acesso ao conhecimento e da criação de meios para o compartilhamento e organização de dados na área da saúde, este trabalho propõe uma plataforma baseada no paradigma de conhecimento como serviço. Essa, poderá ser utilizada na área médica, especificamente no domínio da nefrologia, podendo oferecer, de forma centralizada, o acesso a ontologias e a outros meios de representação e processamento do conhecimento. Foi realizada uma descrição detalhada de cada parte da arquitetura e de sua implementação, destacando suas principais características e interfaces. Como fonte de conhecimento, adaptou-se uma ontologia já existente no domínio da nefrologia, possibilitando a inferência no conhecimento modelado e a adaptação de um mecanismo de suporte à decisão clínica. Além disso, foi especificado um protocolo de comunicação para ser usado entre o aplicativo consumidor e o serviço provedor de conhecimento. Diante dos fatos mencionados, o desenvolvimento desta pesquisa pôde contribuir para o surgimento de uma nova arquitetura, o NefroService, que se firmou como um sistema capaz de gerenciar múltiplas fontes de dados e conhecimento, centralizando o acesso aos mesmos através de uma API facilmente adaptável.

**Palavras-chave:** Conhecimento como serviço, Nefrologia, Ontologia, Informática em Saúde.

## ABSTRACT

Due to the need to improve access to knowledge and the establishment of means for sharing and organizing data in the health area, this research proposes a platform based on the paradigm of knowledge as a service. This can be used in the medical field, specifically in the field of Nephrology, and can offer centralized access to ontologies and other means of knowledge representation. A detailed description of each part of the architecture and its implementation was made, highlighting its main features and interfaces. As a source of knowledge, an ontology that already exists was adapted, allowing the inference in the knowledge model and the adaptation of a mechanism to support the clinical decision. In addition, a communication protocol was specified and used between the knowledge consumer and the knowledge service provider. Thus, the development of this research contributed to the creation of a new architecture, the NefroService, which established itself as a system capable of managing multiple data sources and knowledge models, centralizing access through an easily adaptable API.

**Keywords:** KaaS, Nephrology, Ontology, Health Informatics.

## LISTA DE FIGURAS

1	O paradigma conhecimento como serviço . . . . .	20
2	Arquitetura baseada em KaaS no domínio da nefrologia . . . . .	27
3	Documentação das classes Serviço e Método, especificadas pela NefroServiceAPI . . . . .	31
4	Consultas executadas durante e execução do método estadiamento . . . . .	33
5	Função auxiliar para o cálculo da Taxa de Filtração Glomerular. . . . .	33
6	Tela original para de entrada de dados do OntoDecideDRC . . . . .	34
7	Página de autenticação do NefroServiceWebsite . . . . .	35
8	Página dedicada ao serviço OntoDecideDRC . . . . .	36
9	Formulário para a entrada de dados do paciente no método estadiamento. . . . .	36
10	Resultado da execução do método estadiamento . . . . .	37
11	Documentação detalhada da NefroServiceAPI . . . . .	37
12	Resposta recebida da API utilizando a página de documentação . . . . .	38

## LISTA DE TABELAS

1	Comparativo entre as diferentes arquiteturas analisadas . . . . .	26
---	---	----

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CAD	<i>Computer-Aided Design</i>
CAM	<i>Computer-Aided Manufacturing</i>
DaaS	<i>Data as a Service</i>
DL	Lógica Descritiva
DRC	Doença Renal Crônica
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	Inteligência Artificial
JSON	<i>JavaScript Object Notation</i>
KaaS	<i>Knowledge as a Service</i>
MDRD	<i>Modification Of Diet In Renal Disease</i>
NC	<i>Nada Consta</i>
NoSQL	<i>Not Only SQL</i>
OWL	<i>Web Ontology Language</i>
RDB	<i>Relational Database</i>
RDF	<i>Resource Description Framework</i>
REST	<i>Representational State Transfer</i>
SaaS	<i>Software as a Service</i>
SOA	<i>Service Oriented Architecture</i>
SQL	<i>Structured Query Language</i>
XML	<i>Extensible Markup Language</i>
YAML	<i>YAML Ain't Markup Language</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Contextualização . . . . .	14
1.2	Motivação . . . . .	14
1.3	Objetivo geral . . . . .	15
1.4	Objetivos específicos . . . . .	15
1.5	Estrutura da monografia . . . . .	16
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	Representação do Conhecimento e Raciocínio . . . . .	17
2.1.1	Ontologias . . . . .	17
2.1.2	Raciocinadores . . . . .	18
2.2	Arquiteturas Orientadas à Serviço . . . . .	18
2.2.1	Software como Serviço (SaaS) . . . . .	19
2.2.2	Dados como Serviço (DaaS) . . . . .	19
2.2.3	Conhecimento como Serviço (KaaS) . . . . .	20
2.3	Informática na Saúde . . . . .	21
2.3.1	Nefrologia e Doença Renal Crônica . . . . .	21
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>23</b>
3.1	Domínio Geral . . . . .	23
3.2	Área da Saúde . . . . .	24
3.3	Análise Comparativa e Conclusão . . . . .	25
<b>4</b>	<b>PLATAFORMA BASEADA EM KAAS NO DOMÍNIO DA NEFROLOGIA</b>	<b>27</b>
4.1	Arquitetura . . . . .	27
4.1.1	Serviço Provedor de Conhecimento . . . . .	28
4.1.2	Consumidor de Conhecimento . . . . .	29
4.1.3	API de Comunicação . . . . .	29
4.1.4	Fontes de dados . . . . .	31

4.2	Detalhes da Implementação . . . . .	32
<b>5</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS</b>	<b>35</b>
5.1	Website consumidor de conhecimento . . . . .	35
5.2	Documentação da API de comunicação . . . . .	37
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>39</b>
6.1	Conclusões . . . . .	39
6.2	Trabalhos Futuros . . . . .	40
	<b>REFERÊNCIAS</b>	<b>41</b>

# 1 INTRODUÇÃO

## 1.1 Contextualização

Segundo Sim et al. (2001) sistemas de suporte a decisão clínica possuem potencial para reduzir a quantidade de erros médicos e melhorar a qualidade e eficiência do tratamento clínico oferecido. O autor conclui que para melhorar substancialmente a qualidade do serviço prestado, é necessário agregar o conhecimento médico adquirido baseado na prática e na literatura médica em uma base de conhecimento comum, facilmente computável e adaptável.

Com o avanço do poder de processamento e velocidade de obtenção de dados na internet, muitas organizações especializaram-se no desenvolvimento de ferramentas, técnicas para modelagem e na criação de estruturas dedicadas ao compartilhamento do conhecimento. A representação do conhecimento, subárea da inteligência artificial, busca encontrar meios de representar, armazenar e manipular de forma automática o conhecimento usando algoritmos raciocinadores (BRACHMAN; LEVESQUE, 2004).

No contexto do compartilhamento e distribuição do conhecimento, um serviço provedor de conhecimento (KaaS) visa prover de maneira centralizada conhecimento que normalmente é extraído de várias fontes de dados e pode ser mantido por diferentes organizações. Nele, um servidor de conhecimento responde a requisições feitas por um ou mais consumidores de conhecimento (XU; ZHANG, 2005). Também é responsabilidade do servidor de conhecimento manter um conjunto de modelos de conhecimento que ajudarão a responder as requisições dos consumidores. Nesse contexto, fica claro que o uso desse paradigma permite o acesso ao conhecimento por consumidores que, de outra forma, não seriam capazes de obter as respostas as suas requisições (KRISHNASWAMY; Wai Loke; ZASLAVSKY, 2001).

Pretende-se assim com este trabalho propor uma arquitetura baseada no paradigma de conhecimento como serviço, a fim de que seja possível criar uma base de conhecimento comum, que poderá ser usada na área médica, especificamente no domínio da nefrologia, com objetivo de facilitar o diagnóstico de pacientes, além da possibilidade de oferecer de forma centralizada o acesso a ontologias e a outros meios de representação e processamento do conhecimento.

## 1.2 Motivação

Devido a necessidade de melhorar o acesso ao conhecimento e da criação de meios de compartilhamento e organização de dados na área da saúde, a informática na saúde tem tentado encontrar soluções para resolver problemas comuns aos profissionais e especialistas de domínio (CAMPOS, 2013).

A gestão e compartilhamento do conhecimento é uma área promissora, porém ainda se mostra ineficiente no domínio da saúde (SABBATINI, 2004). O conhecimento gerado através das experiências dos profissionais da área, comumente não são repassados satisfatoriamente, ficando assim retidos em suas próprias mentes (SILVA et al., 2005).

A quantidade de dados coletados na área da saúde aumenta periodicamente, resultando no surgimento de métodos de diagnósticos, princípios químicos, além de progressos na área da biologia molecular e da genética, entre outros avanços medicinais (WECHSLER; ANÇÃO; SIGULEM, 2003).

A Doença Renal Crônica (DRC), é considerada um dos principais problemas de saúde pública no mundo, tendo um alto índice de mortalidade, tanto em países desenvolvidos, quanto em países em desenvolvimento. Uma das características mais relevantes da DRC é a perda progressiva da função renal, levando muitos pacientes a necessitarem de algum tipo de terapia renal substitutiva (SAUDE, 2014).

De acordo com o censo da Sociedade Brasileira de Nefrologia feito em 2013, embora o número de unidades de diálise no Brasil venha aumentando, a quantidade de pacientes em tratamento dialítico por ano cresce numa velocidade ainda maior (SBN, 2013). Sendo assim, o diagnóstico precoce e a prevenção da doença têm se tornado cada vez mais importante para retardar seu avanço, pois permitem a educação pré-diálise e execução de medidas preventivas que podem retardar o avanço da DRC (BASTOS; KIRSZTAJN, 2011).

Diante dos fatos mencionados, esta pesquisa optou pela criação e implementação de uma arquitetura flexível, baseada no paradigma de conhecimento como serviço, focada na manipulação e compartilhamento do conhecimento, que poderá ser usado no auxílio ao diagnóstico e encaminhamento apropriado dos pacientes.

### **1.3 Objetivo geral**

Como objetivo geral, esta pesquisa visa apresentar uma plataforma baseada no paradigma de conhecimento como serviço no domínio da nefrologia.

### **1.4 Objetivos específicos**

Os objetivos específicos dessa pesquisa podem ser identificados abaixo, em sequência, como:

1. Modelar uma arquitetura baseada no paradigma de Conhecimento como Serviço no contexto da nefrologia;

2. Especificar uma API de comunicação que será usado para transmitir conhecimento entre o Provedor de Conhecimento e os aplicativos consumidores;
3. Implementar o servidor de conhecimento, API de comunicação e aplicativo consumidor utilizando a arquitetura proposta;
4. Adaptar um protótipo existente, no domínio da nefrologia, à arquitetura proposta de maneira a permitir sua execução.

## 1.5 Estrutura da monografia

Este trabalho está dividido em seis capítulos com os seguintes tópicos: Introdução, Fundamentação Teórica, Trabalhos Relacionados, Plataforma Baseada em KaaS no domínio da Nefrologia, Resultados e Conclusões.

No Capítulo 1 é apresentada a contextualização do problema, motivação e o objetivo geral e específicos desta monografia.

O Capítulo 2 apresenta os conceitos relevantes para o entendimento da plataforma proposta neste trabalho.

O Capítulo 3 mostra trabalhos similares já realizados pela academia e faz uma análise comparativa entre as abordagens propostas em cada trabalho.

O Capítulo 4 apresenta uma proposta de arquitetura baseada no paradigma KaaS voltada para a área da saúde, bem como descreve cada um de seus módulos e detalhes de sua implementação.

O Capítulo 5 discute os resultados obtidos com a implementação e detalha o fluxo de execução dentro do serviço provedor de conhecimento.

Para finalizar, o Capítulo 6 resume a pesquisa realizada, apresenta suas contribuições e aponta possíveis trabalhos futuros relacionados ao tema.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados, com a finalidade de permitir uma maior compreensão do detalhamento técnico das atividades e resultados nos capítulos seguintes, alguns tópicos utilizados no decorrer do desenvolvimento deste trabalho. O item 2.1 trata temas como formas de representação do conhecimento e raciocinadores. O item 2.2 detalha a história das arquiteturas orientadas a serviço e principalmente o paradigma de conhecimento como serviço. Por fim, o tópico 2.3 apresenta os conceitos de informática na saúde e nefrologia, além de, explicar algumas características da doença renal crônica.

### 2.1 Representação do Conhecimento e Raciocínio

A representação do conhecimento é uma área da inteligência artificial (IA) que se preocupa com a forma que o conhecimento pode ser representado simbolicamente e manipulado de modo automático por algoritmos raciocinadores (BRACHMAN; LEVESQUE, 2004).

Dada uma estrutura de representação de conhecimento e um processo de raciocínio, é possível que se obtenha conclusões a partir do conhecimento previamente modelado. Essas conclusões podem ser utilizadas para auxiliar na tomada de decisão (LADEIRA, 1997).

Para entendermos como a arquitetura proposta na presente monografia será capaz de responder as requisições, detalharemos a seguir tanto a estrutura de representação do conhecimento, quanto a lógica de raciocínio escolhida.

#### 2.1.1 Ontologias

O termo ontologia, quando usado na área de ciências da computação, no contexto de sistemas de representação do conhecimento, refere-se a uma estrutura geral de conceitos representados por um vocabulário lógico. Esse tema ganhou notoriedade na década de 1990, quando uma série de conceitos chamados de web semântica surgiram e propuseram a possibilidade de executar inferências automáticas na web (ALMEIDA, 2013). Com isso, atualmente, as ontologias estão sendo usadas em ciências da computação para resolver problemas em diferentes domínios como medicina, biologia, entre outros (ASHBURNER et al., 2000; FAROOQ et al., 2012; DIENG-KUNTZ et al., 2006).

Uma outra definição relevante utilizada, é que uma ontologia é uma especificação de uma conceitualização. Nessas, definições são ligadas a nomes de entidades através de relações e axiomas que descrevem e restringem a interpretação e o uso desses termos (GRUBER, 1996).

É importante notar que as ontologias não conseguem descrever completamente toda a lógica do domínio e frequentemente precisam de regras semânticas adicionais baseadas nos conceitos da ontologia. Normalmente, essas regras são definidas como um conjunto de condições que implicam em uma ação (TAVARES, 2016).

### 2.1.2 Raciocinadores

Para inferir nas ontologias, necessita-se de um mecanismo de inferência chamado de algoritmos raciocinadores. Esses algoritmos permitem a comparação da sintaxe, estrutura possivelmente normalizada e conceitos expressos na ontologia (BAADER, 2003).

O Hermit é um raciocinador baseado em lógica descritiva (DL) quem tem como objetivo ser eficiente e implementar uma série de melhorias que o permite trabalhar com ontologias maiores e mais complexas (SHEARER; MOTIK; HORROCKS, 2008).

Existem também vários frameworks e interfaces para a manipulação e armazenamento de ontologias como, por exemplo, o Apache Jena e a OWL API. O Apache Jena consiste em um framework para a manipulação de dados ligados e web semântica. Possui mecanismos de serialização de grafos no formato *Resource Description Framework* (RDF) além de um mecanismo de armazenamento de triplas e uma API para manipulação de ontologias (FOUNDATION, 2011).

Uma outra *Application Programming Interface* (API) de alto nível para manipulação de ontologias é a OWL API. Uma serie de padrões criados na linguagem de programação Java que dão suporte a extração, validação e visualização das definições da sintaxe do arquivo da ontologia em formato RDF ou *Web Ontology Language* (OWL) (HORRIDGE; BECHHOFFER, 2011).

O benefício do uso de uma API de manipulação de ontologias é permitir a utilização de um novo raciocinador em um sistema já implementado, sem a necessidade de grandes alterações do código fonte. Além disso, permite-se que o desenvolvedor utilize um nível de abstração mais alto ao inferir na ontologia, concentrando-se na manipulação da mesma em vez de tratar problemas como sintaxe e serialização de dados (HORRIDGE; BECHHOFFER; NOPPENS, 2007). Por esse motivo, escolhemos usar a OWL API em conjunto com o raciocinador Hermit durante o desenvolvimento do presente trabalho a fim diminuir a dependência entre a arquitetura proposta e o algoritmo raciocinador usado.

## 2.2 Arquiteturas Orientadas à Serviço

As arquiteturas orientadas a serviço (SOA) surgiram da necessidade de integração e automação de negócios através da internet. Elas têm como principais características o desacoplamento do código, o uso de padrões de computação distribuída e a independência

de protocolo (PAPAZOGLU, 2003).

Em SOA, os recursos são empacotados como "serviços" bem definidos, modulares e que produzem uma saída padronizada e independente do estado ou contexto de outras partes da aplicação (FREMANTLE; WEERAWARANA; KHALAF, 2002).

O uso de arquiteturas orientadas a serviço permite aos desenvolvedores a resolução de problemas comuns a sistemas distribuídos, como a integração de aplicações, o gerenciamento de transações, implementação de políticas de segurança além da compatibilidade com sistemas legados (ALONSO et al., 2004).

### **2.2.1 Software como Serviço (SaaS)**

O paradigma Software como Serviço (SaaS) descreve aplicativos e software entregues como um serviço através da internet. Essa arquitetura já se tornou um modelo importante para a venda e entrega de software em vários setores da indústria, oferecendo diversos benefícios tanto para os provedores de serviço quanto para seus usuários (ARMBRUST et al., 2010).

Do ponto de vista dos usuários, a utilização da arquitetura SaaS apresenta inúmeras vantagens, como: redução dos custos, elasticidade, atualizações automáticas e fácil implementação. Para as empresas desenvolvedoras de software, essa arquitetura oferece uma nova forma de vender funcionalidades para seus clientes e compete com modelos de negócio tradicionais (BENLIAN; KOUFARIS; HESS, 2012).

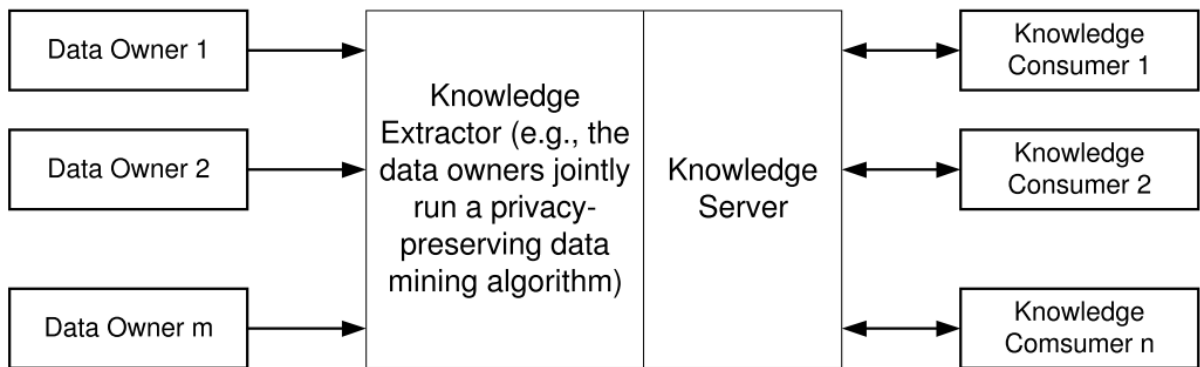
### **2.2.2 Dados como Serviço (DaaS)**

O conceito de Dados como Serviço (DaaS) pode ser definido como quando dados em vários formatos e de várias fontes podem ser acessados com um serviço por usuários na rede. Seus usuários podem manipular os dados remotamente como se os mesmos estivessem disponíveis localmente. Também podem ser oferecidos serviços onde o acesso aos dados é feito de forma semântica, usando palavras chaves ou índices (WANG et al., 2008).

Existem várias plataformas populares hoje que oferecem esse tipo de serviço, como o Google Drive, Amazon S3 e o Dropbox. Todos esses permitem que o usuário manipule os dados localmente, sendo a sincronização feita de maneira transparente e invisível ao usuário. Além disso, são oferecidas APIs que permitem desenvolvedores manipular os dados usando suas próprias aplicações (DROPBOX, 2016; GOOGLE, 2016; AMAZON, 2016).

### 2.2.3 Conhecimento como Serviço (KaaS)

No contexto do compartilhamento e distribuição do conhecimento, um serviço provedor de conhecimento (KaaS) tem como objetivo prover, de maneira centralizada, conhecimento que normalmente é extraído de várias fontes de dados e pode ser mantido por diferentes organizações. Nele, um servidor de conhecimento responde a requisições feitas por um ou mais consumidores de conhecimento (XU; ZHANG, 2005).



**Figura 1:** A figura mostra uma visão geral da arquitetura KaaS destacando seus principais componentes: detentores de dados, serviço provedor de conhecimento e os consumidores de conhecimento. Fonte: Xu; Zhang, (2005)

Segundo Xu e Zhang (2005), em uma implementação da arquitetura KaaS, como vemos na figura 1, podemos encontrar três principais componentes: detentores de dados, serviço provedor de conhecimento e os consumidores de conhecimento. Detalhadamente eles podem ser descritos como:

- Os detentores de dados são responsáveis por coletar dados de suas transações diárias e são os principais responsáveis por filtrar e proteger as informações coletadas. Cada detentor de conhecimento pode ser mantido por uma organização diferente, além de poder disponibilizar um grande volume de dados.
- O serviço provedor de conhecimento, visa centralizar e prover o conhecimento através de seu servidor de conhecimento, onde os dados são extraídos usando um algoritmo extrator que, por sua vez, é responsável pela leitura apropriada das informações de cada detentor de dados.
- Os consumidores de conhecimento são aplicações que usam o conhecimento do provedor em seu processo de tomada de decisões. A comunicação com o servidor é feita por um protocolo e sistema de autenticação previamente estabelecido e pode ocorrer de forma bidirecional. Uma requisição é enviada em um formato específico e o servidor, por sua vez, responde baseado em modelos de conhecimento e nos dados extraídos.

Também é responsabilidade do servidor de conhecimento manter um conjunto de modelos de conhecimento que ajudarão a responder às consultas dos consumidores. Nesse contexto, fica claro que o uso desse paradigma permite o acesso ao conhecimento por consumidores que, de outra forma, não seriam capazes de obter as respostas as suas requisições (KRISHNASWAMY; Wai Loke; ZASLAVSKY, 2001).

### **2.3 Informática na Saúde**

A informática na saúde é um termo abrangente que pode se referir ao prontuário eletrônico do paciente, processamento de sinais biológicos, além de sistemas de informação na saúde, inteligência artificial em saúde e até programas para treinamento especialistas (BRASIL, 2008).

Segundo Hoyt, Sutton e Yoshihashi (2008) a informática na saúde pode ser definida como o campo da ciência que lida com recursos, equipamentos e métodos formais para otimizar o armazenamento, leitura e gerenciamento de informações médicas na resolução de problemas e tomada de decisões.

Com a evolução da tecnologia e devido à dificuldade do gerenciamento de processamento da informação da área médica, a informática na saúde tem tentado desenvolver ferramentas e algoritmos com o objetivo de solucionar problemas comuns aos profissionais da área da saúde (CAMPOS, 2013).

Sendo assim, a informática na saúde visa melhorar a qualidade de serviços de saúde, reduzindo custos e permitindo a troca de informações médicas (HOYT; SUTTON; YOSHIHASHI, 2008).

#### **2.3.1 Nefrologia e Doença Renal Crônica**

A nefrologia é uma área da medicina que tem como objetivo o diagnóstico e tratamento clínico das doenças do sistema urinário, principalmente relacionadas ao rim (SBN, 2016).

A Doença Renal Crônica (DRC) é definida como lesão do parênquima renal (com função renal normal) e/ou pela diminuição funcional renal presentes por um período igual ou superior a três meses (BASTOS; BREGMAN; KIRSZTAJN, 2010). O desenvolvimento da doença no paciente é frequentemente assintomático o que dificulta o diagnóstico antes que a doença atinja seu estágio avançado (BASTOS; KIRSZTAJN, 2011).

Junior (2004) afirma que a DRC consiste em uma lesão renal, acompanhada de uma perda progressiva e irreversível da função dos rins (Glomerular, Tubular e Endócrina). Em sua fase mais avançada, os rins não conseguem mais manter a normalidade do meio interno do paciente.

A DRC é classificada em seis estágios baseados no nível de lesão renal do paciente. Dependendo do estágio em que o paciente se encontra, pode ser necessário um manejo clínico especial ou uma terapia específica com o objetivo de retardar a progressão da DRC e prevenir suas complicações (JUNIOR, 2004).

De acordo com o censo da Sociedade Brasileira de Nefrologia feito em 2013, embora o número de unidades de diálise no Brasil venha aumentando, o número de pacientes em tratamento dialítico por ano cresce numa velocidade ainda maior (SBN, 2013). Sendo assim, o diagnóstico precoce e a prevenção da doença têm se tornado cada vez mais importante a fim de que sejam tomadas medidas preventivas que podem retardar ou parar o avanço da DRC (BASTOS; KIRSZTAJN, 2011).

### 3 TRABALHOS RELACIONADOS

Com o objetivo de entender e comparar pesquisas similares, foram analisadas diversas propostas de arquiteturas e frameworks aplicados a diferentes domínios.

Os trabalhos relacionados foram divididos em dois grupos: aplicações de domínio geral e sistemas na área da saúde. Essa divisão tem por objetivo facilitar o entendimento sobre as diversas abordagens existentes e destacar os principais aspectos de cada implementação.

É importante notar que cada trabalho propõe arquiteturas com suas próprias características e, a fim de deixar a análise mais objetiva, foram selecionadas apenas as propriedades mais relevantes a esta pesquisa.

Por fim, no item 3.3, é feita uma análise comparativa das abordagens encontradas, para que seja possível criar uma relação com a arquitetura proposta neste trabalho.

#### 3.1 Domínio Geral

Venkataramana e Padmavathamma (2012) em seu artigo intitulado "*A Design of Framework for AGRI-CLOUD*" propõe a arquitetura do Agri-Cloud, um framework baseado no paradigma SaaS que oferece serviços especializados para agricultores em relação ao cultivo, preços de fertilizantes, entre outras funcionalidades. Cientistas e especialistas podem adicionar novos modelos de conhecimento, sugestões para técnicas modernas de plantação, uso de fertilizantes, além de poderem obter dados históricos de certas regiões. Da mesma forma, o framework Agri-Cloud pode beneficiar organizações privadas e governamentais permitindo a inserção e leitura de dados e estatísticas.

A interação com os aplicativos consumidores pode ser feita através da camada SaaS, de maneira barata usando sensores, dispositivos móveis e requisições personalizadas. Os principais componentes da arquitetura do Agri-Cloud são:

- **Camada de aquisição:** Contém bibliotecas para analisar os dados e convertê-los para um formato uniforme baseado no padrão *Extensible Markup Language* (XML). Além disso, contém ferramentas para o processamento de linguagem natural, a fim de extrair possíveis padrões presentes nos dados;
- **Camada de processamento:** Camada especialista que contém algoritmos de modelagem e suporte a decisões. Aqui são usados vários módulos responsáveis por responder diretamente a algumas requisições de consumidores, utilizando-se de técnicas da IA, a fim de prover soluções dinamicamente;

- **Camada de serviço de armazenamento:** Módulo responsável pelo armazenamento, leitura e inferência dos dados e modelos criados. Contém diversos submódulos responsáveis por encriptação, armazenamento do conhecimento, indexação de imagens, entre outros.

Foi proposto por Grolinger et al. (2013) o Disaster-CDM, um framework baseado em SOA, mais especificamente no paradigma de conhecimento como serviço, tendo como principais objetivos:

- Ser capaz de armazenar uma grande quantidade de dados relacionados a desastres de diversas fontes;
- Facilitar a busca e indexação de dados de desastres;
- Fornecer ferramentas de suporte para resolução de requisições e permitir a interoperabilidade com outros sistemas.

O Disaster-CDM tem como principal forma de armazenamento dos dados, o uso de Banco de Dados Relacionais (RDB) e Bancos de Dados NoSQL. A arquitetura usa regras semânticas para melhor organização e leitura da informação. A aquisição dos dados é feita a partir do processamento de várias fontes de dados distintas, como documentos, simulações, sensores e arquivos de texto semiestruturados.

A comunicação com os aplicativos consumidores pode se dar por três formas: ontologias, APIs e serviços. Essas três formas de acesso aos dados, permitem o Disaster-CDM responder às requisições de forma integrada sem depender da forma em que os dados são salvos internamente.

Também vale a pena destacar o trabalho de Yoo e Kim (2002), no qual é proposto um sistema para gerenciamento de conhecimento e compartilhamento de informações sobre produtos. Nessa arquitetura, são usadas diversas fontes de conhecimento, como arquivos do tipo *Computer-Aided Design* (CAD), XML e websites para a extração e modelagem do conhecimento armazenado. O conhecimento é representado usando ontologias, metadados e tabelas, podendo ser acessado pelos consumidores através de uma API, capaz de responder as consultas utilizando vários padrões baseado em XML.

### 3.2 Área da Saúde

No domínio da saúde, um artigo relevante é o de Yoo, Gnanasekaran e Cheng (2014) que descreve uma arquitetura orientada a serviço colaborativa, projetada para facilitar o compartilhamento e a cooperação entre prestadores de serviço de saúde, além de reduzir os custos para o paciente. A arquitetura descrita tem três componentes principais:

- **Centralizador de colaborações médicas:** Mantém uma versão atualizada dos dados de hospitais e do mecanismo de tomada de decisão. É de responsabilidade do agente centralizador, extrair informações a partir dos dados providos pelos prestadores de serviço de saúde e criar diagramas com fluxos de tratamento de saúde;
- **Aplicativos consumidores:** Aplicações que, através de consultas XML, podem acessar o conhecimento provido pelo módulo centralizador;
- **Prestadores de serviços de saúde:** São as fontes de dados da arquitetura e possuem uma descrição detalhada sobre seus próprios serviços. Essas informações serão coletadas pelo serviço centralizador, a fim de gerar modelos de conhecimento e estatísticas sobre cada fonte.

O acesso aos dados se dá por duas formas: funções de gerenciamento de subscrições e serviços de consultas. As funções para o gerenciamento de subscrições permitem aos aplicativos consumidores, se inscreverem para receber atualizações sobre uma certa categoria de informação. Já os serviços de consultas, permitem que requisições padronizadas sejam executadas a qualquer momento em que os consumidores necessitarem.

Uma outra pesquisa voltada para o domínio da saúde foi feita por Lai, Tam e Chan (2012), nela são identificadas as principais características de uma rede colaborativa para serviços médicos e, além disso, são exploradas formas de como o KaaS poderia ser usado para a criação de uma rede privada de conhecimento médico na China.

### 3.3 Análise Comparativa e Conclusão

A Tabela 1 permite visualizar cada arquitetura ou framework, citados anteriormente, de acordo com cinco características relevantes: domínio, paradigma utilizado, formato que o conhecimento é armazenado internamente, fonte de dados e o formato de dados escolhido para a comunicação com os aplicativos consumidores.

Considerando as informações vistas na Tabela 1, podemos observar que arquiteturas baseadas em serviço podem ajudar no suporte à decisão em diversos domínios.

Também podemos ver que a maioria das arquiteturas usam um formato textual para serialização dos dados e comunicação com os consumidores, com exceção de uma, que oferece o acesso direto a suas ontologias.

Por outro lado, podemos perceber que as fontes de conhecimento variam conforme o domínio e disponibilidade de serviços detentores de dados. Sendo assim, é necessário a adaptação de cada arquitetura a seu domínio, a fim de permitir um maior aproveitamento e compartilhamento do conhecimento de domínio disponível.

**Tabela 1: Comparativo entre as diferentes arquiteturas analisadas**

Pesquisa	Domínio	Paradigma	Formato do Conhecimento	Fonte de Conhecimento	Formato de Saída
(1)	Agricultura	SaaS	Multiplos	Especialistas, Sensores, Imagens, etc.	XML
(2)	Gestão de Desastres	KaaS	RDB/NoSQL	Ontologias, website e outros	Ontologias e APIs
(3)	Negócios	KaaS	Ontologias, XML e outros	Website, Arquivos CAD/CAM, XML e outros	XML
(4)	Saúde	KaaS	NC	NC	XML
(5)	Saúde	SOA	NC	Prestadores de Serviços de Saúde	XML

(1) - Venkataramana e Padmavathamma (2012); (2) - Grolinger et al. (2013); (3) - Yoo e Kim (2002); (4) - Yoo, Gnanasekaran e Cheng (2014); (5) - Lai, Tam e Chan (2012).

Tendo em vista a necessidade de gerenciamento e distribuição do conhecimento na área da saúde, o paradigma de conhecimento como serviço se mostra uma alternativa promissora, capaz de organizar e disponibilizar esse conhecimento para que possa ser consumido por diversas aplicações.

## 4 PLATAFORMA BASEADA EM KAAS NO DOMÍNIO DA NEFROLOGIA

Ao estudarmos a arquitetura proposta por Xu e Zhang (2005) para o paradigma conhecimento como serviço, ficou claro que a mesma pode ser adaptada para o domínio da nefrologia. Com isso em mente, foi projetada uma plataforma similar, o NefroService, que contém as especificidades do domínio.

No decorrer deste capítulo, será detalhada a arquitetura proposta, a API de comunicação com os aplicativos consumidores e as particularidades de sua implementação.

### 4.1 Arquitetura

O NefroService tem como objetivo centralizar o acesso ao conhecimento gerado a partir de diversas fontes distintas de dados, tornando o compartilhamento do mesmo mais eficiente. Uma visão geral da arquitetura do NefroService pode ser visto na figura 2.

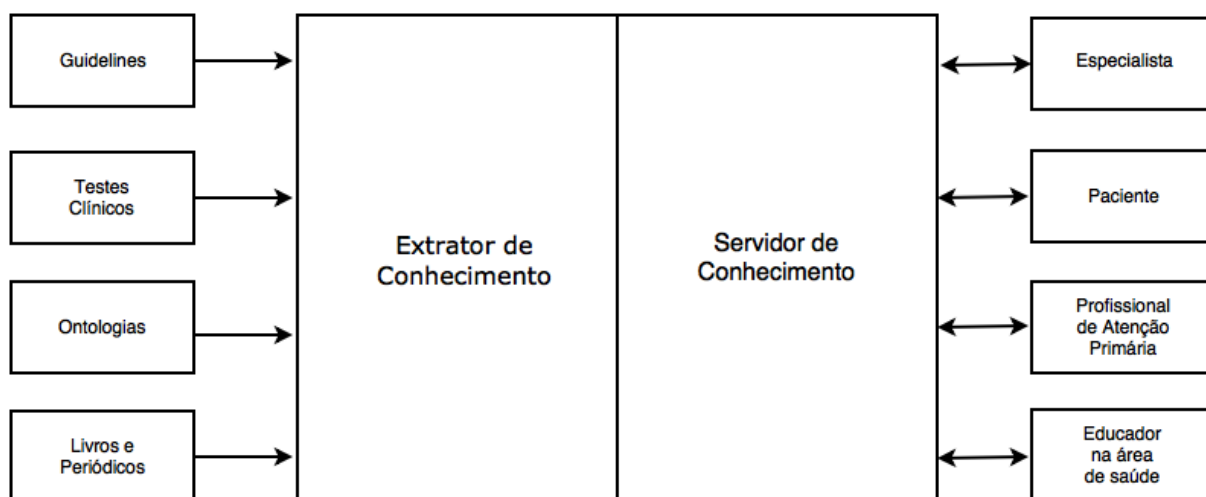


Figura 2: Uma visão geral da arquitetura KaaS baseada no domínio da nefrologia. Fonte: Autoria Própria.

No domínio da nefrologia, poderiam ser consideradas fontes de dados, por exemplo, resultados de testes clínicos, ontologias de domínio, livros, periódicos, orientações para o tratamento de doenças, entre outros. Cada fonte de dado possui suas próprias características e deve ser tratada de forma independente, visando facilitar a inclusão e alteração de novas regras de extração.

Em relação aos aplicativos consumidores, é possível a criação de diversas soluções para cada parte interessada existente no domínio. Por exemplo, podem ser criados aplicativos para o auxílio à decisão clínica, objetivando uma melhora do serviço prestado por especialistas e profissionais da atenção primária.

Além disso, é possível a criação de aplicações e modelos de conhecimento direcionados à educação de novos profissionais ou de pacientes interessados em estudar sobre o domínio da nefrologia ou doenças relacionadas.

A arquitetura do NefroService foi dividida em módulos, a fim de mais fácil ser mantida, entendida e implementada. Os principais módulos que compõem a arquitetura são: serviço provedor de conhecimento, consumidor de conhecimento, API de comunicação e fontes de dados. Nos próximos tópicos falaremos mais detalhadamente sobre cada um deles.

#### **4.1.1 Serviço Provedor de Conhecimento**

Dentro do paradigma KaaS, o módulo provedor de conhecimento tem como objetivo acessar e processar os dados da fonte de dados, gerir os modelos de conhecimento e servir consultas feitas pelos consumidores de conhecimento (XU; ZHANG, 2005).

No contexto do NefroService, o serviço provedor de conhecimento é constituído de duas partes principais: o extrator de conhecimento e a implementação do lado servidor da API de comunicação.

Com o objetivo de responder as consultas dos consumidores, o servidor implementa a API de comunicação, fornecendo respostas padronizadas, facilmente entendidas pelos aplicativos consumidores. Falaremos detalhadamente sobre a API de comunicação do NefroService no item 4.1.3.

A extração de conhecimento é feita à medida que consultas são realizadas pelos aplicativos consumidores, através de inferências a ontologia, consultas a documentos ou outras fontes de dados e conhecimento.

Cada fonte de dados possui regras de extração correspondentes dentro do extrator de conhecimento. Sendo assim, para cada nova fonte de dados adicionada, regras de extração e acesso precisam ser escritas para que seja possível a integração da mesma com o sistema existente.

Além disso, fica dentro do extrator de conhecimento, algoritmos auxiliares, responsáveis por diversas funções comuns para a leitura e manipulação das informações. Alguns exemplos desses algoritmos são: funções para leitura de arquivos de texto, leitura e conversão de imagens, filtros para garantir o anonimato dos dados, entre outros. Também é possível que o extrator de conhecimento possua um mecanismo de armazenamento de dados próprio, a fim de indexar e melhorar a velocidade de acesso para consultas específicas.

### 4.1.2 Consumidor de Conhecimento

A arquitetura do NefroService, de maneira similar ao paradigma KaaS, prevê a possibilidade de ser acessado por diferentes aplicativos consumidores de conhecimento. Esses aplicativos, se utilizam da API de comunicação para realizar consultas a base de conhecimento central.

A implementação de cada aplicativo pode variar dependendo de seu objetivo e linguagens de programação escolhidas, desde que a comunicação com a API ocorra conforme sua especificação.

No quesito segurança, cada aplicativo deverá possuir uma chave única, privada, que será usada durante a comunicação com o serviço provedor a fim de autenticar, limitar e/ou identificar as consultas realizadas. A falha no provimento da chave de segurança impossibilitará o acesso ao serviço provedor de conhecimento, sendo este responsável pela criação, armazenamento e compartilhamento das chaves.

Entre possíveis aplicativos consumidores, no domínio da nefrologia, podem ser citados: websites para o suporte à decisão clínica, aplicativos para dispositivos móveis ou sistemas embarcados, prontuário eletrônico, sistemas educacionais, entre outros.

### 4.1.3 API de Comunicação

A comunicação entre o serviço provedor de conhecimento e os aplicativos consumidores se dá através da API de comunicação, a NefroServiceAPI.

Segundo Masse (2011), uma API é um serviço web, baseados em interfaces de programação bem definidas, que permite a comunicação entre aplicativos. Em outras palavras, uma API pode ser considerada um meio no qual se permite a comunicação entre dois programas de computador.

Para a especificação da API, usou-se os princípios mencionados por Bloch (2006), que auxiliam a criação de uma boa API de comunicação, sendo os principais deles:

- Uma API deve ser pequena e simples;
- Quando ocorrerem, os erros devem ser reportados imediatamente e de forma consistente;
- Manter um número pequeno de parâmetros;
- Todas as funções e objetos devem estar bem documentados.

Recentemente, um estilo de arquitetura para a especificação de APIs foi proposto por Richardson e Ruby (2008), o *Representational State Transfer* (REST). APIs que

seguem esse paradigma são chamadas de RESTful APIs ou APIs REST. Essa arquitetura utiliza-se de padrões similares a um protocolo bastante utilizado na internet, o *Hypertext Transfer Protocol* (HTTP).

Em APIs REST, existem quatro tipos principais de comandos: *GET*, *POST*, *PUT* e *DELETE*. Cada comando é responsável por uma operação nos dados. O comando *POST*, por exemplo, é responsável pela atualização de informações. De forma similar, o comando *DELETE* é capaz de remover informações do servidor. Já os comandos *GET* e *PUT* são responsáveis, respectivamente, por ler e adicionar dados no servidor (IETF, 2014).

Uma outra característica importante de APIs REST é seu formato de saída de dados, onde são usados padrões semiestruturados de serialização de objetos em formato textual, como o *JavaScript Object Notation* (JSON) e XML (RICHARDSON; RUBY, 2008).

A NefroServiceAPI é uma API baseada na arquitetura REST e utiliza-se do formato de dados JSON, que é um formato leve, baseado em texto, independente de linguagem, para a transferência de informações. Esse formato define algumas regras de formatação que possibilitam a representação de dados de forma estruturada (CROCKFORD, 2006).

Ao projetar a NefroServiceAPI, observou-se a necessidade de especificação de dois comandos:

- ***/servico***

Comando que recebe requisições do tipo *GET* e lista os serviços atualmente implementados na plataforma. Cada serviço possui suas descrições, código de identificação e, principalmente, uma lista com os métodos disponíveis.

Os métodos são funções que podem ser executadas a qualquer momento pelo aplicativo consumidor, possibilitando o acesso ao conhecimento disponível. A documentação de cada objeto pode ser vista na figura 3;

- ***/servico/id/método***

Comando responsável pela execução de um método de um serviço específico. Recebe requisições no formato *POST* e, por isso, para executá-lo, o aplicativo precisará fornecer informações serializadas baseadas no formulário de entrada presente na chave *input*. A resposta desse comando é um objeto serializado em JSON, contendo algumas informações sobre a consulta realizada e o resultado de sua execução.

Em relação ao tratamento e formato dos erros, a NefroServiceAPI possui um objeto especial da classe *Erro* que é instanciado e retornado quando algum problema fatal

```

Servico {
  id (string, optional): Identificador único do serviço.,
  nome (string, optional): Um nome amigável para o serviço.,
  descricaoCurta (string, optional): Um descrição curta do serviço e de sua ontologia base.,
  descricaoLonga (string, optional): Um descrição longa do serviço. Pode conter tags HTML básicas como B, BR, P e outras.,
  metodos (Array[Metodo], optional)
}
Metodo {
  id (string, optional): O código identificador do método.,
  nome (string, optional): Um nome amigável para o serviço.,
  descricao (string, optional): Uma descrição para o método. Pode conter HTML.,
  input (string, optional): O formulário base de entrada.
}

```

**Figura 3: Documentação escrita para as classes Serviço e Método, usadas na NefroServiceAPI como modelo de resposta para seus comandos. Fonte: Autoria Própria.**

acontece durante a execução da consulta. Alguns possíveis erros são: erro de autenticação, comando HTTP incompatível, erro interno do servidor, serviço indisponível, método inexistente, parâmetros inválidos, entre outros.

No contexto da segurança, para a execução de cada chamada a API, é necessário que o aplicativo consumidor envie uma chave de segurança obrigatória. Essa chave garante que o acesso ao conhecimento estará limitado apenas a aplicações autorizadas.

#### 4.1.4 Fontes de dados

Conforme explicado por Xu e Zhang (2005), embora o extrator de conhecimento possa manipular e filtrar os dados de cada fonte de dados, não é responsabilidade da arquitetura a anonimização dos dados.

O NefroService prevê a possibilidade de múltiplas fontes de dados independentes, permitindo uma maior flexibilidade na obtenção das informações e/ou conhecimento provido por serviços externos.

O sistema de acesso aos dados pode variar de acordo sua fonte e, por isso, requer-se do extrator de conhecimento a implementação dos meios necessários para a leitura e extração das informações relevantes. O acesso aos dados pode se dar de duas maneiras:

- **Local:** Os dados estão organizados no mesmo sistema de arquivos onde é executado o serviço;
- **Remoto:** Dados são transferidos a partir de fontes remotas, ocasionalmente indisponíveis.

Devido ao acesso remoto ser menos confiável, recomenda-se a utilização de me-

canismos de indexação e cache, a fim de melhorar a performance global do sistema em momentos de instabilidade.

Para algumas fontes de dados remotas, poderá ser feita a obtenção em massa das informações, com o propósito de que seja executado o processamento local em lote. Desta forma, mesmo que a fonte de dados pare de funcionar depois da obtenção dos dados, o NefroService ainda seria capaz de executar a extração e criação dos modelos de conhecimento a partir dos dados previamente obtidos.

## 4.2 Detalhes da Implementação

Um dos objetivos desta pesquisa, é a implementação da arquitetura proposta e a adaptação de um protótipo já existente no domínio da nefrologia.

Foram estudados dois protótipos a fim de escolher um para que fosse adaptado a nova arquitetura.

A OntoDecideDRC, uma plataforma baseada em *web services*, tem como objetivo dá suporte à decisão clínica aos médicos especialistas da área da nefrologia e os que trabalham na atenção primária da saúde (TAVARES, 2016).

Uma outra pesquisa similar foi feita por (CAMPOS, 2016) onde foi criado o EducaDRC, um referatório semântico de objetos de aprendizagem sobre DRC.

Ambas as pesquisas possuem, como sistema de representação de conhecimento, ontologias únicas, criadas com a ajuda de especialista no domínio da nefrologia.

A principal diferença quanto as abordagens é o fato da EducaDRC, por ser um referatório semântico, necessitar de um banco de dados adicional para armazenamento de triplas e metadados, usados para descrever os recursos externos. Sendo assim, devido a independência de banco de dados, foi escolhida a OntoDecideDRC como ontologia e protótipo base para ser adaptado ao NefroService.

O serviço provedor de conhecimento foi implementado na linguagem PHP e, além de conter os comandos necessários para a extração e inferência do conhecimento, também implementa os comandos descritos na especificação da API.

Para a adaptação da ontologia presente na OntoDecideDRC ao módulo extrator de conhecimento, fez-se necessário a escrita de um aplicativo de linha de comando, capaz de executar o raciocinador HermiT e inferir em uma ontologia do tipo OWL, utilizando consultas baseadas na lógica descritiva. Esse aplicativo foi chamado de NefroServiceWrapper, implementado na linguagem de programação Java e faz parte do módulo extrator de conhecimento. A figura 4, mostra dois exemplos de consultas e seus respectivos resultados.

Também foram criadas, dentro do extrator de conhecimento, algumas funções auxi-

## Resultado:

Paciente: Josefa Silva Possui some (T2 and A2 and Alteração\_da\_Imagem)  
Possui DRC: Sim  
Classificado no: Estagio 2 É\_Classificada\_por some (T2)

Figura 4: Consultas executadas durante e execução do método estadiamento. Fonte: Autoria Própria.

liares para a execução das consultas a ontologia. Uma delas é a implementação do cálculo da Taxa de Filtração Glomerular (TFG). Esse cálculo usa a fórmula MDRD simplificada especificada por Levey et al. (2000) e, através das informações do paciente, chega-se a um valor numérico que será utilizado durante a inferência ao modelo de conhecimento.

A implementação do cálculo da TFG pode ser vista na figura 5. O método recebe atributos do paciente como creatinina, sexo, cor da pele e idade e retorna o valor aproximado da TFG.

```
/**  
 * MDRD_simplificada  
 * Função auxiliar no calculo da TFG.  
 */  
function mdrd_simplificada($creatinina, $mulher, $negro, $idade) {  
    $creatinina = (float) $creatinina;  
  
    $resultado = 186;  
    if ($mulher) {  
        $resultado *= 0.742;  
    }  
    if ($negro) {  
        $resultado *= 1.212;  
    }  
    $resultado *= pow ( $creatinina, - 1.154 );  
    $resultado *= pow ( $idade, - 0.203 );  
    return $resultado;  
}
```

Figura 5: Função auxiliar para o cálculo da Taxa de Filtração Glomerular, presente no módulo extrator de dados. Fonte: Autoria Própria.

O módulo consumidor, que chamamos de NefroServiceWebsite, teve como objetivo testar a API e implementar a interface gráfica do protótipo escolhido.

A linguagem de programação utilizada para a implementação do NefroServiceWebsite foi PHP e, em conjunto com o framework Wordpress, possibilitou a implementação de várias páginas como: login, registro, recuperar senha, listar serviços, executar método, contato, entre outras.

Ao adaptar a OntoDecideDRC, percebeu-se que o código fonte de sua interface

gráfica não estava disponível e, por isso, foi necessário a criação de um novo formulário para a entrada de dados baseado na especificação original do protótipo. A figura 6 mostra a tela original do protótipo para a entrada de dados. Essa, serviu como base para o método estadiamento do serviço OntoDecideDRC no NefroService. Todos os campos do formulário foram reescritos e organizados de maneira similar a sua versão original. Os resultados serão apresentados no capítulo seguinte, na figura 9.

The screenshot shows a web form with the following sections:

- Informações Básicas do Paciente:** Fields for Name, Age, Height (cm), Skin Color (radio buttons: Afro-Americano, Pardo, Branco), Gender (radio buttons: Feminino, Masculino), Blood Pressure (mmHg), and Weight (kg). A button labeled 'Aplicar Conhecimento' is on the right.
- Vacinação:** A question 'O Paciente está com a vacinação em dia?' with 'Sim' and 'Não' radio buttons. Below is a 'Lista de Vacinas' text area.
- Exames:** Fields for 'Relação Albuminúria Creatinúria (RAC)' and 'Creatinina'. Below is a 'Presença de:' section with checkboxes for 'Albuminúria', 'Alteração da Imagem', and 'Proteinúria'. At the bottom, a 'Fórmula para Cálculo da Taxa de Filtração Glomerular' section has radio buttons for 'CKD-EPI' and 'MDRD Simplificada'.
- Fatores de Risco:** Two columns of checkboxes. The first column lists diseases: Doença Cística, Doença Vascular, Doença Congênita, Doença Glomerular, Doença Túbulo Intersticial. The second column lists risk groups: Afro-Americanos, Baixo Peso ao Nascer, Crianças com Menos de 5 Anos, Diminuição d Massa do Rim, Obstrução do Trato Urinário, Infecções do Trato Urinários, Cálculos Urinários, Doenças Auto-Imunes, Hipertensão, Histórico Familiar, Idoso, Mulheres Grávidas, Neoplasia, Obesidade, Diabetes, Tabagismo, and Usuário de Drogas.
- Sintomas Incomum:** A grid of checkboxes for symptoms: Acidose Metabólica, Adinamia, Anemia, Asterixis, Cefaléias, Convulsões, Dano Renal Parenquimatoso, Deficiência de Hormônios Gonadotróficos, Deficiência de vitamina D, Dor no meio das Costas, Edema, Escurecimento da Pele, Gastrintestinais, Hipervolemia, Hipopotassemia, Hálito Urêmico, Imunodepressão, Insuficiência Renal Aguda, Lesão da Estrutura Renal, Letargia, Monoparesias, Perda da Concentração, Pericardite, Peritonite, Pleurite, Problemas Urinários, Purido, Sangramentos, Sintomas Urêmicos, and Torpor.

**Figura 6:** Tela original para de entrada de dados do OntoDecideDRC antes da adaptação ao NefroService. Fonte: Tavares (2016).

Em relação a documentação técnica da NefroServiceAPI, usou-se o framework *Swagger Editor*. Este framework, permite a escrita e especificações de APIs REST de maneira robusta, utilizando linguagens semiestruturadas como *YAML Ain't Markup Language* (YAML) ou JSON. O benefício de utilização desse framework é a criação quase automática de páginas capazes de descrever de forma simplificada todos os aspectos da API. Também é possível a execução de certos métodos, diretamente da página de documentação, criando uma maneira fácil para execução de testes durante o desenvolvimento de novos aplicativos (SOFTWARE, 2016).

## 5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Com o desenvolvimento deste trabalho, realizado por intermédio de pesquisas sobre o paradigma de conhecimento como serviço e da análise de frameworks similares em vários domínios, obteve-se a criação de uma plataforma robusta, facilmente adaptável e capaz de oferecer acesso a modelos de conhecimento já existentes.

Este capítulo apresenta os resultados alcançados com o desenvolvimento do NefroService, demonstrando seu fluxo de execução e interfaces.

### 5.1 Website consumidor de conhecimento

Ao acessar o NefroServiceWebsite, o usuário se depara com um formulário para a inserção de suas credenciais (figura 7). Essas credenciais são de responsabilidade do aplicativo consumidor, que possibilita a criação de diferentes níveis de acesso à aplicação, além de um melhor controle de como as informações são compartilhadas.



**Figura 7: Página de autenticação do NefroServiceWebsite, website consumidor de conhecimento. Fonte: Autoria Própria.**

Depois de autenticado, o usuário verá uma lista com os serviços disponíveis, tal como, suas descrições e métodos. Também encontrará um menu lateral dinâmico, que se adapta baseado na resposta da API ao comando para listar serviços.

Cada serviço possui uma página dedicada, que poderá auxiliar o usuário na escolha de um método adequado (figura 8).

Depois da escolha do método, será mostrado ao usuário o formulário de entrada de dados e uma breve descrição de como preenche-lo. A figura 9 mostra o formulário



Figura 8: Página dedicada ao serviço OntoDecideDRC adaptado para a plataforma NefroService. Fonte: Autoria Própria.

do método estadiamento, similar ao implementado no protótipo original da plataforma OntoDecideDRC.

Figura 9: Formulário para a entrada de dados do paciente no método estadiamento. Fonte: Autoria Própria.

Esse formulário é gerado dinamicamente, a partir dos dados recebidos da API de comunicação. Quando submetido, uma requisição é enviada a API e, a partir dela, é criado o código HTML com a resposta que será mostrada ao usuário (figura 10). Caso, durante o processamento ou envio da requisição, um erro ocorra, a mensagem de erro e uma possível solução será mostrada ao usuário.

A implementação do website consumidor de conhecimento, possibilitou ao usuário da plataforma realizar consultas utilizando formulários similares aos providos original-

## Resultado:

Paciente: Josefa Silva  
Possui DRC: Sim  
Classificado no: Estagio 2

**Figura 10: Resultado da execução do método estadiamento. Fonte: Autoria Própria.**

mente pela plataforma OntoDecideDRC, protótipo escolhido para servir de fonte de conhecimento e de base para a criação da interface gráfica.

## 5.2 Documentação da API de comunicação

Como resultado da criação da especificação da API utilizando o framework *Swagger Editor*, foi possível o desenvolvimento de uma página dedicada para a descrição detalhada de suas funções, objetos e respostas. A página inicial deste website pode ser vista na figura 11.

**NefroServiceAPI**

API REST baseada no paradigma de conhecimento como serviço que provê uma forma unificada de acessar o NefroService.

Essa página é uma referência técnica para desenvolvedores que estão interessados em utilizar a API.

Código fonte  
Você poderá acessar o código fonte da API através deste [link](#).

Suporte  
Se você encontrou algum problema ou precisa de alguma funcionalidade específica, sinta-se livre para acessar nosso [sistema de suporte](#) ou nossa [wiki](#).

Licença  
A licença completa de uso está disponível em nosso repositório que pode ser acessado [aqui](#).

**Serviço** Mostrar/Esconder | Listar operações | Expandir operações

GET	/servico
POST	/servico/{id}/{metodo}

[ BASE URL: / , VERSÃO API: 1.0.0 ]

**Figura 11: Página de documentação da NefroServiceAPI. Fonte: Autoria Própria.**

Caso o desenvolvedor, usuário da página da documentação da API, queira testar algum dos comandos, o mesmo pode clicar em cada um dos caminhos e executar o comando usando sua chave de segurança. O resultado da execução de um dos comandos pode ser visto na figura 12. O comando para listar serviços, por exemplo, retorna uma lista de serviços que, por sua vez, possuem uma lista de métodos que podem ser executados.

```
URL requisição
http://127.0.0.1/servico?apikey=chavedeseguranca

Corpo da resposta

[
  {
    "id": "ontodecidedrc",
    "nome": "OntoDecideDRC",
    "descricaoCurta": "<b>Ontologia que da suporte na identificação e o manejo nefrológico adequado de pacientes com Doença Renal",
    "descricaoLonga": "<b>Ontologia que da suporte na identificação e o manejo nefrológico adequado de pacientes com Doença Renal",
    "metodos": [
      {
        "id": "estadiamento",
        "nome": "Estadiamento",
        "descricao": "Baseado nas informações do paciente, retorna informações sobre a DRC e seu estágio.",
        "input": "\t\r\n\t<div class=\"span6\">\r\n\t\t<h3>Informações do Paciente:</h3>\r\n\t\t<label for=\"nome\"><b>Nome:</b><
      ]
    ]
  }
]
```

**Figura 12:** Resposta recebida da API utilizando a página de documentação.  
**Fonte:** Autoria Própria.

O acesso a API ocorre de acordo com a especificação estabelecida no capítulo 4, utilizando o protocolo HTTP, através dos comandos para a listagem de serviços e execução dos métodos.

## 6 CONSIDERAÇÕES FINAIS

Neste capítulo, são apresentadas as conclusões do presente trabalho e as oportunidades de futuras pesquisas nesta área de estudo.

### 6.1 Conclusões

A presente monografia teve o objetivo apresentar o NefroService, uma plataforma baseada em conhecimento como serviço, no domínio da nefrologia.

Foi realizada uma descrição detalhada de cada parte da arquitetura e, junto da implementação, foram apresentadas suas principais características e interfaces.

Como fonte de conhecimento, adaptou-se uma ontologia no domínio da nefrologia, que permitiu a inferência no conhecimento e a criação do mecanismo de suporte à decisão clínica, similar ao protótipo original. Além disso, foi especificado um protocolo de comunicação, usado entre o aplicativo consumidor e o serviço provedor de conhecimento.

Por fim, foi feita uma análise de como o fluxo de informações e consultas ocorrem dentro da arquitetura, com o objetivo de esclarecer e reforçar a descrição da mesma.

Com relação as dificuldades para o desenvolvimento do trabalho, é importante destacar que, o protótipo escolhido como base para a criação do módulo consumidor, havia sido originalmente escrito na linguagem Flex, atualmente pouco utilizada. Sendo assim, fez-se necessário um estudo minucioso da ontologia, visando reescrever as consultas fazendo uso da OWL API e da linguagem de programação Java. Além disso, a interface gráfica do protótipo não estava mais disponível e, com isso, foi observada a necessidade do desenvolvimento de novos formulários e páginas para a visualização do resultado.

Quanto à revisão bibliográfica, foram abordados diversos temas necessários para o entendimento da arquitetura proposta, bem como para o esclarecimento de sua implementação. Realizou-se também uma análise dos trabalhos relacionados que propunham uma arquitetura ou framework similar, com objetivo de comparar e relacionar suas características com o NefroService.

O paradigma KaaS, embora relativamente novo, se mostra promissor no que se diz respeito a distribuição e acesso ao conhecimento, possibilitando um melhor uso do mesmo em domínios onde, embora exista uma grande quantidade de dados sendo coletados diariamente, ainda não há formas eficientes para que o conhecimento seja repassado de maneira satisfatória.

O presente trabalho abre caminho para que novas fontes de dados e conhecimento sejam desenvolvidas e disponibilizadas de modo mais eficaz. Além disso, podem ser desenvolvidos novos aplicativos que usem esses dados de formas interessantes, através do

uso da API de comunicação. Nesse contexto, as principais contribuições dessa pesquisa foram:

- Em relação ao aspecto computacional, propõe uma arquitetura baseada em conhecimento como serviço, capaz de facilitar o compartilhamento de modelos de conhecimento já existentes;
- Na área da Nefrologia, o desenvolvimento da arquitetura abre novos caminhos para um melhor aproveitamento do conhecimento gerado através das experiências dos profissionais da área, facilitando a criação de novas aplicações capazes de utilizar esses dados.

Em conclusão, o desenvolvimento desta pesquisa contribuiu para o surgimento de uma nova arquitetura, o NefroService, que se firmou como um sistema capaz de gerenciar múltiplas fontes de dados e conhecimento, centralizando o acesso aos mesmos através de uma API facilmente adaptável. Além da criação do website, que pode ser utilizado na área da nefrologia, com o objetivo de ajudar profissionais da área no diagnóstico e encaminhamento eficiente dos pacientes.

## 6.2 Trabalhos Futuros

Durante o desenvolvimento desta pesquisa, foram identificados possíveis temas para a realização de trabalhos futuros relacionados à esta pesquisa. Entre esses, está a modelagem de padrões para criação de interfaces dinâmicas, de entrada e saída de dados, a partir de ontologias e outros meios de representação do conhecimento. Esses padrões facilitariam a integração de novas fontes de dados a arquitetura, permitindo também um melhor aproveitamento do conhecimento.

Adicionalmente, podem ser criados mecanismos para a incorporação de consultas na própria ontologia, de forma a deixa-la mais robusta e facilitar a integração com algoritmos raciocinadores.

Por fim, também podem ser realizadas pesquisas com o objetivo de refinar a arquitetura proposta, adicionando suporte a novas fontes de dados e o desenvolvimento de aplicações e ferramentas capazes de melhor utilizar o conhecimento existente.

## REFERÊNCIAS

- ALMEIDA, M. B. Revisiting ontologies: a necessary clarification. *Journal of the American Society for Information Science and Technology*, Wiley Online Library, v. 64, n. 8, p. 1682–1693, 2013.
- ALONSO, G. et al. Web services: concepts, architectures and applications. In: *Web Services*. [S.l.]: Springer, 2004. p. 123–149.
- AMAZON. *Amazon S3: Armazenamento de objetos simples, resiliente e massivamente escalável*. [S.l.]: Amazon Web Services Inc., 2016. Disponível em: <https://aws.amazon.com/pt/s3/>. Acesso em: 26 nov. 2016.
- ARMBRUST, M. et al. A view of cloud computing. *Communications of the ACM*, ACM, v. 53, n. 4, p. 50–58, 2010.
- ASHBURNER, M. et al. Gene ontology: tool for the unification of biology. *Nature genetics*, Nature Publishing Group, v. 25, n. 1, p. 25–29, 2000.
- BAADER, F. *The description logic handbook: Theory, implementation and applications*. [S.l.]: Cambridge university press, 2003.
- BASTOS, M. G.; BREGMAN, R.; KIRSZTAJN, G. M. Doença renal crônica: frequente e grave, mas também prevenível e tratável. *Rev Assoc Med Bras*, SciELO Brasil, v. 56, n. 2, p. 248–53, 2010.
- BASTOS, M. G.; KIRSZTAJN, G. M. Doença renal crônica: importância do diagnóstico precoce, encaminhamento imediato e abordagem interdisciplinar estruturada para melhora do desfecho em pacientes ainda não submetidos à diálise. *J. bras. nefrol*, v. 33, n. 1, p. 93–108, 2011.
- BENLIAN, A.; KOUFARIS, M.; HESS, T. Service Quality in Software-as-a-Service: Developing the SaaS-Qual Measure and Examining Its Role in Usage Continuance. *Journal of Management Information Systems*, v. 28, n. 3, p. 85–126, 2012. ISSN 0742-1222.
- BLOCH, J. How to design a good api and why it matters. In: ACM. *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*. [S.l.], 2006. p. 506–507.
- BRACHMAN, R.; LEVESQUE, H. Knowledge Representation and Reasoning - Book. *New York*, v. 1, p. 381, 2004. ISSN 87567016. Disponível em: <http://www.amazon.com/dp/1558609326>.
- BRASIL, L. M. Informática em saúde. In: *Informática em Saúde*. [S.l.]: EDUEL, 2008.
- CAMPOS, S. Sistema para raciocínio semântico no domínio dos hospitais de João Pessoa. 2013.
- CAMPOS, S. *EDUCADRC: Referatório Semântico De Objetos De Aprendizagem Sobre DRC*. 2016.

- CROCKFORD, D. The application/json media type for javascript object notation (json). 2006.
- DIENG-KUNTZ, R. et al. Building and using a medical ontology for knowledge management and cooperative work in a health care network. *Computers in Biology and Medicine*, Elsevier, v. 36, n. 7, p. 871–892, 2006.
- DROPBOX. *Dropbox: Seus documentos em qualquer lugar*. [S.l.]: Dropbox Inc., 2016. Disponível em: [https://www.dropbox.com/pt\\_BR/](https://www.dropbox.com/pt_BR/). Acesso em: 26 nov. 2016.
- FAROOQ, K. et al. An Ontology Driven and Bayesian Network Based Cardiovascular Decision Support Framework. *Advances in Brain Inspired Cognitive Systems SE - 4*, v. 7366, p. 31–41, 2012. Disponível em: <http://dx.doi.org/10.1007/978-3-642-31561-9>{\} .}
- FOUNDATION, T. A. S. *Apache Jena*. 2011. Disponível em: <https://jena.apache.org/>. Acesso em: 24 nov. 2016.
- FREMANTLE, P.; WEERAWARANA, S.; KHALAF, R. Enterprise services. *Communications of the ACM*, v. 45, n. 10, p. 77–82, 2002.
- GOOGLE. *Google Drive: Todos os seus arquivos, sempre que você precisar*. [S.l.]: Google Inc., 2016. Disponível em: <https://www.google.com/intl/pt-BR/drive/>. Acesso em: 26 nov. 2016.
- GROLINGER, K. et al. Knowledge as a service framework for disaster data management. *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, p. 313–318, 2013. ISSN 15244547.
- GRUBER, T. *What is an ontology?* 1996. Disponível em: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. Acesso em: 23 nov. 2016.
- HORRIDGE, M.; BECHHOFFER, S. The OWL API: A Java API for OWL ontologies. *Semantic Web*, v. 2, n. 1, p. 11–21, 2011. ISSN 15700844.
- HORRIDGE, M.; BECHHOFFER, S.; NOPPENS, O. Igniting the OWL 1.1 touch paper: The OWL API. *CEUR Workshop Proceedings*, v. 258, 2007. ISSN 16130073.
- HOYT, R. E.; SUTTON, M.; YOSHIHASHI, A. *Medical Informatics: Practical Guide for the Healthcare Professional 2008*. [S.l.]: Lulu. com, 2008.
- IETF. *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. 2014. Disponível em: <https://tools.ietf.org/html/rfc7230>. Acesso em: 30 nov. 2016.
- JUNIOR, J. E. R. Doença Renal Crônica: Definição, Epidemiologia e Classificação. *Jornal Brasileiro de Nefrologia*, v. 26, n. 1, p. 1–3, 2004. ISSN 2175-8239.
- KRISHNASWAMY, S.; Wai Loke, S.; ZASLAVSKY, A. Knowledge elicitation through web-based data mining services. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 2176, p. 120–134, 2001. ISSN 16113349.
- LADEIRA, M. *Representação de Conhecimento e Redes de Decisão*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 1997.

- LAI, I. K.; TAM, S. K.; CHAN, M. F. Knowledge cloud system for network collaboration: A case study in medical service industry in china. *Expert Systems with applications*, Elsevier, v. 39, n. 15, p. 12205–12212, 2012.
- LEVEY, A. et al. A simplified equation to predict glomerular filtration rate from serum creatinine. *JOURNAL OF THE AMERICAN SOCIETY OF NEPHROLOGY (JASN)*, 2000.
- MASSE, M. *REST API design rulebook*. [S.l.]: "O'Reilly Media, Inc.", 2011.
- PAPAZOGLU, M. P. Service-oriented computing: Concepts, characteristics and directions. In: IEEE. *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. [S.l.], 2003. p. 3–12.
- RICHARDSON, L.; RUBY, S. *RESTful web services*. [S.l.]: "O'Reilly Media, Inc.", 2008.
- SABBATINI, R. *Internet e Educação Médica*. [S.l.]: Informática Médica, Campinas, v. 1, n. 3, 2004. Disponível em: <http://www.informaticamedica.org.br/informaticamedica/n0103/editorial.htm>. Acesso em: 27 nov. 2016.
- SAUDE, M. da. *DIRETRIZES CLÍNICAS PARA O CUIDADO AO PACIENTE COM DOENÇA RENAL CRÔNICA – DRC NO SISTEMA ÚNICO DE SAÚDE*. 2014. Disponível em: <http://sonerj.org.br/wp-content/uploads/2014/03/diretriz-cl-nica-drc-versao-final2.pdf>. Acesso em: 26 nov. 2016.
- SBN, S. B. d. N. *Censo de Diálise*. 2013. Disponível em: [http://arquivos.sbn.org.br/pdf/censo\\_2013\\_publico\\_leigo.pdf](http://arquivos.sbn.org.br/pdf/censo_2013_publico_leigo.pdf). Acesso em: 26 nov. 2016.
- SBN, S. B. d. N. *O que é nefrologia?* 2016. Disponível em: <http://sbn.org.br/publico/institucional/o-que-e-nefrologia/>. Acesso em: 25 nov. 2016.
- SHEARER, R.; MOTIK, B.; HORROCKS, I. Hermit: A highly-efficient owl reasoner. In: *OWLED*. [S.l.: s.n.], 2008. v. 432, p. 91.
- SILVA, R. d. P. et al. Modelo de apoio ao diagnóstico no domínio médico: aplicando raciocínio baseado em casos. Universidade Católica de Brasília, 2005.
- SIM, I. et al. Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, The Oxford University Press, v. 8, n. 6, p. 527–534, 2001.
- SOFTWARE, S. *Swagger Editor*. [S.l.]: SmartBear Software, 2016. Disponível em: <http://swagger.io/swagger-editor/>. Acesso em: 28 nov. 2016.
- TAVARES, E. A. *Uma Abordagem Para Suporte À Decisão Clínica Baseada Em Semântica No Domínio Da Nefrologia*. 2016.
- VENKATARAMANA, K.; PADMAVATHAMMA, M. A design of framework for agri-cloud. *IOSR Journal of Computer Engineering*, v. 4, n. 5, p. 1–6, 2012.
- WANG, L. et al. Scientific cloud computing: Early definition and experience. In: *HPCC*. [S.l.: s.n.], 2008. v. 8, p. 825–830.

WECHSLER, R.; ANÇÃO, M. S.; SIGULEM, D. A informática no consultório médico. SciELO Brasil, 2003.

XU, S.; ZHANG, W. Knowledge as a service and knowledge breaching. *Proceedings - 2005 IEEE International Conference on Services Computing, SCC 2005*, I, p. 87–94, 2005.

YOO, J. J.-W.; GNANASEKARAN, K.; CHENG, C.-Y. A collaborative healthcare service framework and performance evaluation. *American Journal of Industrial and Business Management*, Scientific Research Publishing, v. 2014, 2014.

YOO, S. B.; KIM, Y. Web-based knowledge management for sharing product data in virtual enterprises. *International journal of production economics*, Elsevier, v. 75, n. 1, p. 173–183, 2002.