H-KaaS: Uma arquitetura de referência baseada em conhecimento como serviço para e-saúde

Renan Gomes Barreto

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

João Pessoa, 2020

Renan Gomes Barreto

H-KaaS: Uma arquitetura de referência baseada em conhecimento como serviço para e-saúde

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba por Renan Gomes Barreto, sob a orientação da Prof.ª Dr.ª Natasha Correia Queiroz Lino, como parte dos requisitos para a obtenção do título de Mestre em Informática.

Linha de Pesquisa: Computação Distribuída

Orientadora: Prof.ª Dr.ª Natasha Correia Queiroz Lino

Coorientador: Prof. Dr. Gustavo Henrique Matos Bezerra Motta

João Pessoa - PB
Julho de 2020

Catalogação na publicação
Seção de Catalogação e Classificação

B273h Barreto, Renan Gomes.
H-KaaS : uma arquitetura de referência baseada em conhecimento como serviço para e-saúde / Renan Gomes Barreto. - João Pessoa, 2020.
167 f. : il.

Orientação: Natasha Correia Queiroz Lino.
Coorientação: Gustavo Henrique Matos Bezerra Motta.
Dissertação (Mestrado) - UFPB/CCEN.

1. Informática - Saúde. 2. Arquitetura de referência - H-KaaS. 3. Decisão clínica - Suporte - Informática. 4. Inteligência artificial. 5. Ontologias. I. Lino, Natasha Correia Queiroz. II. Motta, Gustavo Henrique Matos Bezerra. III. Título.

UFPB/BC CDU 004:614(043)

Elaborado por Larissa Silva Oliveira de Mesquita - CRB-15/746

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

Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Renan Gomes Barreto, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 29 de julho de 2020.

Aos vinte e nove dias do mês de julho do ano de dois mil e vinte, às quatorze horas, por meio de videoconferência, reuniram-se os membros da Banca Examinadora constituída para julgar o trabalho do sr. Renan Gomes Barreto, vinculado a esta Universidade sob a matrícula nº 20181000724, candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa "Computação Distribuída", do Programa de Pós-Graduação em Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores: Natasha Correia Queiroz Lino (PPGI-UFPB) Orientadora e Presidente da Banca, Clauirton de Albuquerque Siebra (PPGI-UFPB), Examinador Interno, Gustavo Henrique Matos Bezerra Motta (PPGI-UFPB), Examinador Externo ao Programa, Everaldo Marques de Aguiar Junior (Northeastern University), Examinador Externo à Instituição. Dando início aos trabalhos, a Presidente da Banca cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse a exposição oral do trabalho de dissertação intitulado: "H-KaaS: Uma Arquitetura de Referência Baseada em Conhecimento como Serviço para e-saúde". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "***aprovado***". Do ocorrido, eu, Ruy Alberto Pisani Altafim, Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da banca examinadora. João Pessoa, 29 de julho de 2020.

Prof. Dr. Ruy Alberto Pisani Altafim

Prof. Natasha Correia Queiroz Lino
Orientadora (PPGI-UFPB)

Prof. Clauirton de Albuquerque Siebra
Examinador Interno (PPGI-UFPB)

Prof. Gustavo Henrique Matos Bezerra Motta
Examinador Interno (PPGI-UFPB)

Prof. Everaldo Marques de Aguiar Junior
Examinador Externo à Instituição (Northeastern University)

RESUMO

Diante da 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 arquitetura de referência baseada no paradigma de conhecimento como serviço, que poderá ser usada para auxiliar no diagnóstico e na tomada de decisão clínica. A arquitetura de referência descrita, chamada H-KaaS, oferece, de maneira centralizada, acesso a serviços baseados em ontologias, descoberta de conhecimento em banco de dados, aprendizagem de máquina e a outros meios de representação do conhecimento e raciocínio. Para tal, descreve-se detalhadamente o funcionamento da arquitetura, fornecendo exemplos de implementação, destacando seus principais componentes e interfaces de comunicação. Como forma de validação da pesquisa, dois estudos de caso em serviços baseados em conhecimento foram realizados e instanciados de acordo com a arquitetura de referência proposta, usando diferentes meios de representação e extração de conhecimento. Desta forma, o desenvolvimento desta pesquisa colaborou para o surgimento de uma arquitetura de referência no domínio da saúde, capaz de gerenciar múltiplas fontes de dados e modelos de conhecimento, facilitando e centralizando seu acesso e contribuindo, assim, para o avanço do estado da arte da informática em saúde e de aplicações da inteligência artificial e seus sistemas baseados em conhecimento.

Palavras-chave: Conhecimento como Serviço, Arquitetura de Referência, Informática em Saúde, Inteligência Artificial, Descoberta de Conhecimento em Banco de Dados.

ABSTRACT

Facing the need to improve access to knowledge and the establishment of means for sharing and organizing data in the health domain, this research describes a reference architecture based on the paradigm Knowledge as a Service (KaaS), which can be used to assist in clinical diagnosis and decision. The proposed reference architecture, called H-KaaS, offers centralized access to services based on ontologies, knowledge discovery in databases, machine learning, and other means of representing knowledge and reasoning. For this, a detailed description of the architecture is presented, providing implementation examples, and highlighting its main components and communication interfaces. In order to validate the work, two case studies in knowledge-based services were performed, instantiating each service according to the proposed reference architecture and using different means of knowledge representation and extraction. Thus, the development of this research contributed to the creation of a new reference architecture in the health domain able to manage multiple data sources and knowledge models, facilitating its usage and sharing, thus helping to advance the state of the art of health informatics and applications of artificial intelligence and its knowledge-based systems.

Keywords: Knowledge as a Service, Reference Architecture, Health Informatics, Artificial Intelligence, Knowledge Discovery in Databases.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CE	Critérios de Exclusão
CI	Critérios de Inclusão
DaaS	Data as a Service
DataSUS	Departamento de Informática do Sistema Único de Saúde
DL	Description Logic
DRC	Doença Renal Crônica
DST	Doença Sexualmente Transmissível
ELU	Exponential linear unit
HDF	Hierarchical Data Format
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IoT	Internet das Coisas
JSON	JavaScript Object Notation
KaaS	Knowledge as a Service
KDD	Knowledge Discovery in Databases
MDRD	Modification of Diet in Renal Disease
OWL	Web Ontology Language
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QP	Questões de Pesquisa
RDF	Resource Description Framework
RELU	Rectified Linear Unit
REST	Representational State Transfer
RNA	Rede Neural Artificial
SaaS	Software as a Service
SIHSUS	Sistema de Informações Hospitalares do Sistema Único de Saúde
SINAN	Sistema de Informações de Agravos de Notificação
SISCEL	Sistema de Controle de Exames Laboratoriais e Carga Viral do HIV
SOA	Service Oriented Architecture

SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SSDC	Sistemas de Suporte à Decisão Clínica
TFG	Taxa de Filtração Glomerular
TR	Trabalho Relacionado
UML	Unified Modeling Language
WoO	Web of Objects
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

SUMÁRIO

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO

Os avanços da tecnologia mudaram significativamente muitas áreas do conhecimento, entre elas a saúde. A informática em saúde pode ser definida como o estudo da informação, processos de comunicação e sistemas aplicados na área médica (COIERA, 2015). De maneira similar, Hersh e Hoyt (2018) definem a informática em saúde como uma disciplina preocupada com o gerenciamento e a manipulação de dados e informação médicas utilizando computador. Um dos objetivos da informática em saúde é auxiliar no suporte à decisão clínica baseada em computador, que pode ser definido como o uso do computador para trazer conhecimento relevante, apoiando os cuidados de saúde e bem-estar do paciente (GREENES, 2011).

Segundo Sim et al. (2001), sistemas de suporte à decisão clínica (SSDC) possuem potencial para reduzir a quantidade de erros médicos e melhorar a qualidade e eficiência do tratamento clínico oferecido. Os autores concluem que, para aumentar 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.

A gestão e compartilhamento do conhecimento é uma área promissora, porém ainda se mostra ineficiente no domínio da saúde. De forma geral, o conhecimento gerado através das experiências dos profissionais da área não é repassado satisfatoriamente, ficando assim retido em suas próprias mentes (SILVA, R., 2005). Além disso, o acesso a certas especialidades médicas é distribuído de maneira heterogênea, principalmente em países em desenvolvimento, como o Brasil (CONSELHO FEDERAL DE MEDICINA, 2013).

Com o avanço do poder de processamento e da velocidade de obtenção de dados na internet, muitas organizações especializaram-se no desenvolvimento de ferramentas, técnicas para modelagem e estruturas dedicadas ao compartilhamento do conhecimento. A representação do conhecimento, uma subárea da inteligência artificial, tem o objetivo de desenvolver formas de representar, armazenar e manipular conhecimento utilizando algoritmos para raciocínio (BRACHMAN; LEVESQUE, 2004). Nesse contexto, uma ontologia refere-se a uma estrutura geral de conceitos representados por um vocabulário lógico, permitindo a execução de inferências de maneira automática (ALMEIDA, 2013). A descoberta de conhecimento em banco de dados (do inglês *knowledge discovery in databases* – KDD), área

que integra conceitos da inteligência artificial e de banco de dados, pode ser definida como a extração de informação implícita, anteriormente desconhecida, a partir de sistemas de bancos de dados (FRAWLEY; PIATETSKY-SHAPIRO; MATHEUS, 1992).

No contexto da engenharia de software, uma arquitetura de referência é definida como um modelo generalizado de diversos sistemas que compartilham uma ou mais características (BASS; CLEMENTS; KAZMAN, 2003). Estas características são determinadas pelo domínio da aplicação e, normalmente, são identificadas a partir do estudo de uma série de sistemas reais. De maneira similar, arquiteturas de referências podem ser definidas como uma arquitetura de software idealizada, derivada de um estudo de domínio de aplicação, que inclui todas as características que uma classe de sistemas desse domínio possa possuir (SOMMERVILLE, 2007).

As arquiteturas orientadas a serviço (do inglês *service oriented architecture* – SOA) são arquiteturas de software modulares em que serviços são fornecidos para diferentes partes da aplicação (SOMMERVILLE, 2007). Desta maneira, uma aplicação que utiliza SOA pode ser construída a partir da integração de diferentes serviços internos ou externos à organização. Esta abordagem permite aos desenvolvedores a resolução de problemas comuns a sistemas distribuídos, como a integração de aplicações, o gerenciamento de transações e implementação de políticas de segurança, além da compatibilidade com sistemas legados (ALONSO, G. et al., 2004).

A partir do modelo SOA, diversos outros paradigmas foram desenvolvidos em domínios mais específicos como, por exemplo, o software como serviço (do inglês *software as a service* – SaaS), que visa oferecer aplicativos e software entregues como um serviço através da internet (ARMBRUST et al., 2010). Outros exemplos de paradigmas baseados em SOA são: dados como serviço, infraestrutura como serviço e conhecimento como serviço.

Nesse contexto, uma arquitetura baseada no paradigma do conhecimento como serviço (do inglês *knowledge as a service* – KaaS) visa prover, de maneira centralizada, através de serviços bem definidos, conhecimento que normalmente é extraído de várias fontes de dados e que pode ser mantido por diferentes organizações, permitindo o acesso ao conhecimento por consumidores que, de outra forma, não seriam capazes de obter as respostas às suas requisições (KRISHNASWAMY; LOKE; ZASLAVSKY, 2001; XU; ZHANG, 2005).

Pretende-se assim, com este trabalho, projetar uma arquitetura de referência baseada no paradigma de conhecimento como serviço para o domínio da saúde, que poderá ser usada na área médica, visando permitir o suporte à decisão clínica, além da possibilidade de oferecer, de

forma centralizada, o acesso a diferentes ontologias e a outros meios de representação e processamento do conhecimento, como KDD e mineração de dados por meio de algoritmos de aprendizagem de máquina.

1.2. MOTIVAÇÃO

De acordo com Moreira, Alvarenga e Oliveira (2004), existe grande demanda para o desenvolvimento de sistemas computacionais que lidem com recuperação e troca de informação e conhecimento. Desta forma, surgem diariamente novos instrumentos para organização e compartilhamento de conhecimento, com o objetivo de facilitar o desenvolvimento desses sistemas.

No domínio da saúde, a quantidade de dados coletados aumenta constantemente, subsidiando o surgimento de vários avanços na medicina como, por exemplo, novos métodos de diagnóstico, novos princípios químicos e melhorias nas áreas da biologia molecular e da genética, entre outros (WECHSLER et al., 2003).

Um dos principais obstáculos para a adoção em massa de sistemas de suporte à decisão clínica está na heterogeneidade dos sistemas de informação em saúde, resultando na falta de padronização do acesso ao conhecimento e no consequente aumento dos custos para o compartilhamento do conhecimento entre organizações (KAWAMOTO, 2012). Atualmente, embora existam uma grande quantidade de padrões sendo propostos, não existe um modelo padrão de dados clínicos que seja aceito e utilizado universalmente, o que implica a limitação das trocas de conhecimento entre organizações de saúde – que, geralmente, restringem o acesso aos dados a poucos parceiros específicos (SACHDEVA; BHALLA, 2012).

A fim de suprir essa necessidade de compartilhamento de dados e de conhecimento entre organizações no domínio da saúde, têm sido propostos diferentes sistemas computacionais utilizando padrões da web semântica na área da saúde. Porém, conforme será discutido em maiores detalhes no capítulo 3, apesar de existir a proposta de padrões para semântica e interoperabilidade de dados, ainda não existem padrões para componentes de software e suas interfaces de comunicação, que torna muitas abordagens menos flexíveis e geralmente incompatíveis com outras abordagens no mesmo domínio. Desta forma, faltam-se meios para comparar sistemas existentes, baseados em serviços de conhecimento, no domínio da informática e saúde.

Fica, assim, evidente a necessidade de especificação de um padrão arquitetural capaz de servir como referência de comparação entre sistemas, de modo a facilitar o entendimento de como eles funcionam e permitir um estudo mais detalhado das escolhas feitas pelos autores.

As arquiteturas de referências (BASS; CLEMENTS; KAZMAN, 2003) podem ser utilizadas para facilitar o desenvolvimento de aplicações, diminuindo o tempo de implementação e permitindo a padronização de interfaces e comunicação entre seus componentes. Além disso, um dos principais objetivos de uma arquitetura de referência é servir como meio de discussão de arquiteturas e de comparação entre sistemas diferentes em um mesmo domínio (SOMMERVILLE, 2007). Sendo assim, fica evidente a necessidade da especificação de uma arquitetura de referência capaz de generalizar sistemas de compartilhamento de conhecimento no domínio da saúde.

Diante desses fatos, esta pesquisa propõe uma arquitetura de referência, chamada H-KaaS (Health - Knowledge as a Service), para auxiliar na tomada de decisão no domínio da saúde. A H-KaaS baseia-se no paradigma de conhecimento como serviço (XU; ZHANG, 2005), estando focada na manipulação e no compartilhamento do conhecimento.

Espera-se, dessa maneira, que a arquitetura de referência posposta padronize o compartilhamento de modelos de conhecimento e de raciocínio no domínio da saúde, contribuindo para o desenvolvimento de novos sistemas centralizados de compartilhamento de conhecimento, e facilitando o estudo de arquiteturas e sistemas existentes no domínio.

1.3. OBJETIVOS

Para a realização desta pesquisa, foram definidos objetivos geral e específicos que serão listados a seguir.

1.3.1. OBJETIVO GERAL

Este trabalho tem como objetivo geral projetar uma arquitetura de referência baseada no paradigma de conhecimento como serviço para o domínio da saúde, esperando-se assim, que a arquitetura proposta se firme como um sistema capaz de gerenciar múltiplas fontes de dados e conhecimento, centralizando seu acesso através interfaces de comunicação adaptáveis e contribuindo para o avanço do estado da arte da informática em saúde.

1.3.2. OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral, foram definidos objetivos específicos que podem ser identificados abaixo, em sequência, como:

- **Objetivo 1:** Realizar uma análise da literatura acadêmica preexistente com o objetivo de identificar arquiteturas, paradigmas e protótipos de software cuja finalidade seja o compartilhamento de conhecimento na área da saúde;
- **Objetivo 2:** Identificar os componentes que farão parte da arquitetura a ser proposta, fornecendo detalhes sobre sua utilização, apresentando exemplos de como esses componentes podem ser instanciados no domínio da saúde;
- **Objetivo 3:** Especificar os tipos de interfaces de comunicação que poderão ser utilizadas para a troca de informação com as fontes de dados e consumidores de conhecimento;
- **Objetivo 4:** Propor uma arquitetura de referência no domínio da saúde, capaz de facilitar o compartilhamento de conhecimento e sistemas de raciocínio de forma centralizada;
- **Objetivo 5:** Executar dois estudos de caso, instanciando diferentes componentes da arquitetura, de forma a validar e exemplificar seu uso em contextos distintos, implementando extratores de conhecimento capazes de acessar e raciocinar em bases de conhecimento específicas, modeladas utilizando diferentes métodos de representação de conhecimento e raciocínio.

1.4. ESTRUTURA DA MONOGRAFIA

Este trabalho está dividido em sete capítulos com os seguintes tópicos: Introdução, Fundamentação Teórica, Trabalhos Relacionados, H-KaaS: Uma arquitetura de referência baseada em conhecimento como serviço para e-saúde, Estudos de Caso, Resultados e Discussão e, por fim, Considerações Finais.

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

O Capítulo 2 apresenta os conceitos relevantes para o entendimento deste trabalho, expondo os principais conteúdos que servirão de base para a construção da arquitetura de referência proposta.

O Capítulo 3 mostra, por meio de uma revisão sistemática da literatura, trabalhos similares, já realizados pela academia, e faz uma análise comparativa entre as abordagens sugeridas.

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

O Capítulo 5 detalha os estudos de caso executados a fim de validar a arquitetura em diferentes contextos, utilizando múltiplos sistemas de representação de conhecimento e raciocínio, mostrando como, e em quais situações, os componentes da arquitetura podem ser instanciados, além de descrever o fluxo de execução dentro dos serviços provedores de conhecimento desenvolvidos.

O Capítulo 6 expõe os resultados obtidos com a realização desta pesquisa e discute sobre como a arquitetura de referência proposta pode ser utilizada, tanto na elaboração de novos sistemas quanto para a comparação entre sistemas existentes.

Para finalizar, o Capítulo 7 resume a pesquisa realizada, expõe suas limitações, apresenta suas contribuições, publicações associadas e, por fim, aponta os 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.

A seção 2.1 define os principais conceitos de arquiteturas de software, arquiteturas orientadas a serviço e, principalmente, o paradigma de conhecimento como serviço, paradigma usado como base da arquitetura proposta. De maneira similar, a seção 2.2 trata de temas como formas de representação do conhecimento, ontologias e raciocinadores. A seção 2.3 aborda os conceitos de descoberta de conhecimento em banco de dados, mineração de dados, aprendizagem de máquina e redes neurais. A seção 2.4 define os principais conceitos da informática em saúde. Por fim, a seção 2.5 resume o que foi apresentado e mostra sua importância em relação à presente pesquisa. As seções relativas à doença renal crônica e câncer cervical podem ser encontradas na contextualização de seus respectivos estudos de caso.

2.1. ARQUITETURAS DE SOFTWARE

Uma arquitetura de software pode ser definida como um conjunto de estruturas necessárias para a compreensão de um sistema, incluindo elementos de software, as relações e suas propriedades (BASS; CLEMENTS; KAZMAN, 2003). De maneira similar, o padrão ISO/IEC/IEEE 42010 (2011) define uma arquitetura de software como o conjunto de conceitos fundamentais de um sistema, incluindo seus elementos, seus relacionamentos e as principais características para seu projeto e evolução.

Outra definição bastante utilizada descreve uma arquitetura de software como o processo de definição de uma solução estruturada, que cumpre os requisitos operacionais e técnicos, além de otimizar atributos comuns de qualidade, performance, segurança e gerenciamento. Este processo envolve uma série de decisões baseadas em uma gama de fatores, e cada uma dessas decisões pode afetar significativamente o sucesso da aplicação (MICROSOFT, 2009).

Um dos primeiros passos para a definição de uma arquitetura de software é identificar os subsistemas que a compõem, estabelecendo um *framework* de comunicação e controle desses subsistemas (SOMMERVILLE, 2007). A utilização desses modelos possibilita ao

desenvolvedor compreender a operação da aplicação, comparar aplicações similares e avaliar seus componentes com o objetivo de reusá-los.

Nas subseções seguintes serão abordados alguns tipos de arquiteturas de software como arquiteturas orientadas a serviço e arquiteturas de referência, conceitos utilizados na proposta de arquitetura desta pesquisa.

2.1.1. ARQUITETURAS ORIENTADAS A SERVIÇO

As arquiteturas orientadas a serviço (do inglês *service oriented architecture* – 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 (PAPAZOGLOU; HEUVEL; VAN DEN, 2007).

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 e a implementação de políticas de segurança, além da compatibilidade com sistemas legados (ALONSO, G. et al., 2004).

2.1.1.1. SOFTWARE COMO SERVIÇO

O paradigma do software como serviço (do inglês *software as a service* – 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).

Devido à popularização do modelo de computação na nuvem, plataformas baseadas no paradigma de software como serviço vêm se tornando cada vez mais comuns no mercado. Nesse contexto, com a transição entre o modelo de software perpétuo e o do software como serviço, vários desafios foram criados em diversas áreas correlatas como o armazenamento distribuído de dados, escalabilidade e infraestrutura de sistemas de informação; sendo assim, pesquisas recentes nessas áreas tentam resolver tais desafios (CHEN, H., 2016; RAFIQUE et al., 2017; SOURI; ASGHARI; REZAEI, 2017).

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, 2011).

De maneira mais detalhada, Fox, Patterson e Joseph (2014) listam as vantagens de utilização do paradigma do software como serviço para os usuários e desenvolvedores:

- Os usuários não precisam instalar nenhum tipo de aplicativo, visto que o acesso à aplicação é feito, na maioria das vezes, por meio de navegadores;
- Os dados são armazenados nos servidores do provedor do serviço, permitindo acessá-los de diferentes localidades ou dispositivos;
- O uso do paradigma SaaS facilita o acesso e o compartilhamento dos dados a grupos de usuários ou funcionários de uma empresa;
- Atualizações e adição de novas funcionalidades são facilitadas, visto que os desenvolvedores não precisam pedir que os usuários atualizem seus aplicativos ou hardware.

Desta maneira, a combinação do conceito de arquiteturas orientadas a serviço com a rápida evolução da internet viabilizou o desenvolvimento do paradigma SaaS, trazendo inúmeras vantagens tanto para o desenvolvedor quanto para o usuário desse tipo de aplicação.

2.1.1.2. DADOS COMO SERVIÇO

Por definição, o termo "dados como serviço" (do inglês *data as a service* – DaaS) refere-se a dados em vários formatos e de várias fontes que podem ser acessados como um serviço por usuários na rede. Seus usuários podem manipular os dados remotamente como se estes estivessem disponíveis localmente. Também podem ser oferecidos serviços em que o acesso aos dados é feito de forma semântica, usando palavras-chave ou índices (WANG et al., 2008).

Existem várias plataformas populares hoje que oferecem esse tipo de serviço como Google Drive[1], Amazon S3[2], Mega[3] e Dropbox[4]. Estas permitem que o usuário manipule os

[1] https://www.google.com/intl/pt-br/drive/about.html
[2] https://aws.amazon.com/pt/s3/
[3] https://mega.nz/
[4] https://www.dropbox.com

dados localmente, sendo a sincronização feita de maneira transparente e invisível ao usuário. Além disso, são oferecidas APIs (*application programming interfaces*) que permitem aos desenvolvedores manipular os dados usando suas próprias aplicações (AMAZON, 2018; DROPBOX, 2018; GOOGLE, 2018; MEGA, 2019).

Do ponto de vista da aceitação da tecnologia por parte dos usuários, o trabalho de Søilen (2016) apresenta algumas das preocupações que potenciais usuários de plataformas DaaS têm a respeito de sua utilização. O trabalho mostra que estes estão preocupados com a confidencialidade, segurança e qualidade dos dados. Além disso, um dos pontos levantados pelo artigo é sobre a possível manipulação dos dados pelas empresas, visto que, em muitos casos, o acesso aos dados brutos e à fonte original dos dados não está disponível.

Devido ao rápido avanço da tecnologia da informação e da grande quantidade de dados gerados na internet, o uso de sistemas baseados no paradigma DaaS vêm se tornando cada vez mais comuns e, aos poucos, soluções para os principais problemas da área vêm sendo propostas. Por exemplo, o trabalho apresentado por Costa, Costa e Santos (2017), que trata do problema da análise em tempo real de grande quantidade de dados, analisa como o particionamento adequado dos dados pode impactar no tempo de resposta das consultas em diferentes estratégias de armazenamento. De maneira similar, em relação ao problema da preservação da privacidade nos dados e da necessidade de proteger dados sensíveis do usuário, Dagher et al. (2019) propõem um *framework* capaz de publicar um índice criptografado de dados que permita buscas nos dados sem abrir mão da confidencialidade.

Sendo assim, devido à redução de custos de hardware e melhorias na confiabilidade e velocidade das redes de computadores, o mundo vem sofrendo uma mudança de paradigma em relação ao uso de dados, partindo do armazenamento e processamento local de dados, para a coleta e disponibilização em tempo real de uma grande quantidade de dados por meio da internet.

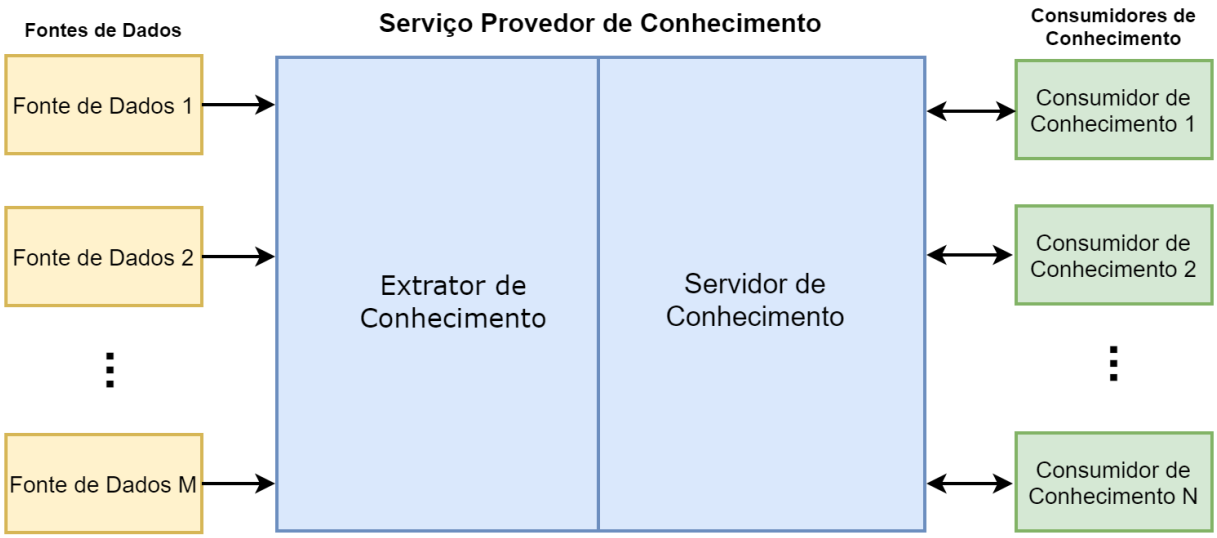
2.1.1.3. CONHECIMENTO COMO SERVIÇO

No contexto do compartilhamento e distribuição do conhecimento, um serviço provedor de conhecimento tem como objetivo fornecer, de maneira centralizada, conhecimento que normalmente é extraído de várias fontes de dados e que 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).

Segundo Xu e Zhang (2005), em uma implementação do paradigma do conhecimento como serviço (do inglês *knowledge as a service* – KaaS), como vemos na Figura 1, podemos encontrar três principais componentes: fontes de dados, serviço provedor de conhecimento e consumidores de conhecimento. Detalhadamente, eles podem ser descritos como:

- As fontes de dados são responsáveis por coletar dados de suas transações diárias e são os principais encarregados de filtrar e proteger as informações coletadas. Cada fonte de dados pode ser mantida por uma organização diferente, além de poder disponibilizar um grande volume de dados.
- O serviço provedor de conhecimento visa a centralizar e prover o conhecimento através de seu servidor de conhecimento, em que os dados são extraídos usando um algoritmo extrator que, por sua vez, é responsável pela leitura apropriada das informações de cada fonte 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.

Figura 1 - Visão geral da arquitetura KaaS destacando seus principais componentes



Fonte: Adaptado de Xu e Zhang (2005).

Também é responsabilidade do servidor de conhecimento manter um conjunto de modelos de conhecimento que ajudarão a responder às consultas dos consumidores por meio de um servidor de conhecimento. 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 às suas requisições (KRISHNASWAMY; LOKE; ZASLAVSKY, 2001).

Trabalhos recentes na área tentam aplicar o paradigma de conhecimento como serviço em diferentes domínios como internet das coisas (IoT), ciência ambiental e saúde (BARRETO; AVERSARI; et al., 2018a; KOLYVAKIS; YOO; KIRITSIS, 2017; ZHAO et al., 2019). Sendo assim, o paradigma de conhecimento como serviço se mostra promissor no que diz respeito à distribuição e acesso ao conhecimento a múltiplos aplicativos consumidores.

2.1.2. ARQUITETURAS DE REFERÊNCIA

No contexto da engenharia de software, uma arquitetura de referência é definida como um modelo generalizado de diversos sistemas que compartilham uma ou mais características (BASS; CLEMENTS; KAZMAN 2003). Estas características são determinadas pelo domínio da aplicação e, normalmente, são identificadas a partir do estudo de uma série de sistemas reais.

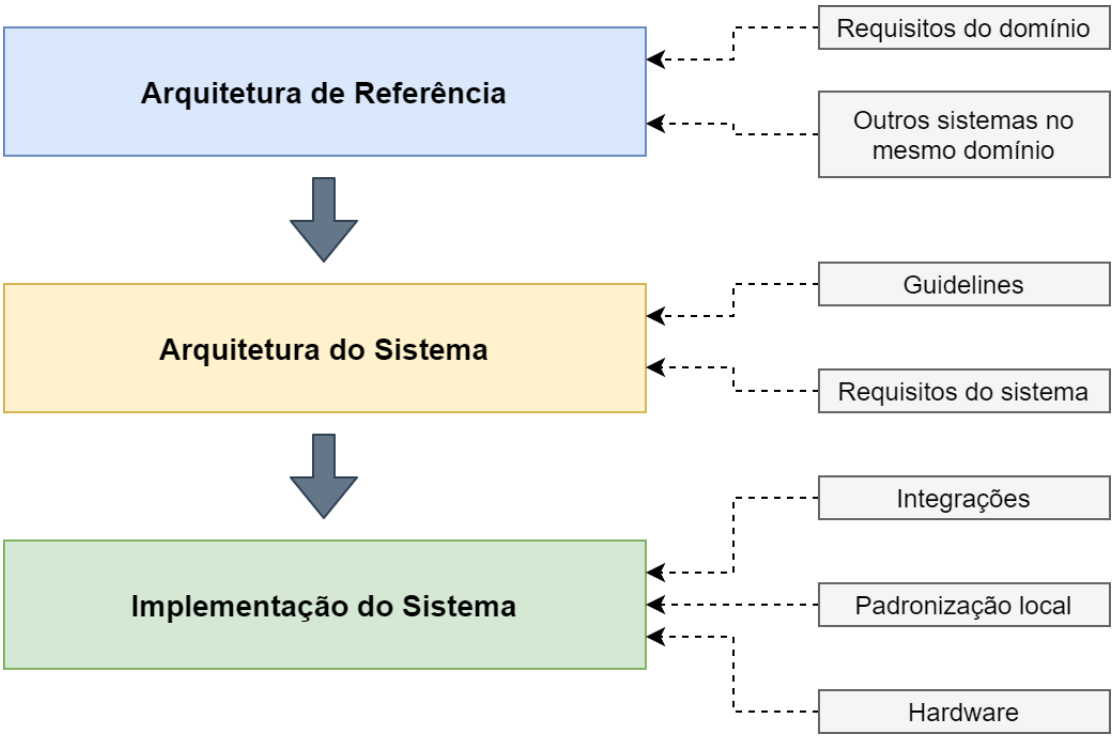
Segundo Gallagher (2000), uma arquitetura de referência é uma arquitetura generalizada criada a partir de vários sistemas existentes que compartilham um ou mais domínios. Essa arquitetura define uma infraestrutura comum aos sistemas e interfaces de componentes que serão incluídos nos sistemas em que ela é utilizada. A partir dela, desenvolvedores poderão instanciar uma arquitetura final específica para o sistema que querem criar. Desta forma, o autor conclui que uma arquitetura de referência faz um papel duplo: primeiro, ela generaliza e extrai funcionalidades e configurações comuns de uma série de aplicações; segundo, ela poderá ser utilizada como base para a instanciação de outros sistemas no mesmo domínio.

De maneira similar, arquiteturas de referências também podem ser definidas como arquiteturas de software idealizadas, derivadas de um estudo de domínio de aplicação, que inclui todas as características que uma classe de sistemas desse domínio possa possuir (SOMMERVILLE, 2007). É importante notar que uma arquitetura de referência isolada não é capaz de descrever todas as características de um sistema específico; ela é criada com o objetivo de fornecer um *framework*-base para uma série de sistemas similares.

O principal objetivo de arquiteturas de referência é a padronização de interfaces entre componentes. Essa padronização reduz o risco e os custos de implementação de novos sistemas,

por facilitar o uso de componentes. Além disso, torna o trabalho de integração entre diferentes organizações produtoras de software mais fácil (GALLAGHER, 2000). A Figura 2 mostra como uma arquitetura de referência pode ser utilizada para a implementação de um novo sistema de software. Inicialmente, tem-se a definição da arquitetura de referência, que foi desenvolvida baseada nos requisitos de domínio e em outros sistemas existentes. Como segundo passo, é criada a arquitetura específica do sistema a ser desenvolvido que, além de ser fundamentada na arquitetura de referência, tem como entrada requisitos e *guidelines* específicos do sistema. Por fim, durante a implementação do sistema são levadas em consideração detalhes como hardware e integração com sistema de terceiros.

Figura 2 - Utilização de uma arquitetura de referência para implementação de um sistema de software



Fonte: Adaptado de Gallagher (2000).

Dentre as principais vantagens da utilização de arquiteturas de referência já existentes para a implementação de novos sistemas computacionais, conforme concluído por Martínez-Fernández et al. (2013), estão:

- Redução de custos de desenvolvimento e de evolução dos sistemas de software implementados;
- Facilitação do design de novos sistemas;

- Padronização de interfaces de comunicação e componentes;
- Reuso sistemático de funcionalidades e de configurações comuns nos sistemas produzidos.

Por outro lado, é importante notar que a utilização de uma arquitetura de referência aumenta a curva de aprendizagem, diminui a flexibilidade para inovação e pode deixar o sistema final mais complexo (MARTÍNEZ-FERNÁNDEZ et al., 2013).

Diante do exposto, espera-se que uma arquitetura de referência possa facilitar o desenvolvimento de aplicações, diminuindo o tempo de implementação e permitindo a padronização de interfaces e a comunicação entre seus componentes.

2.2. REPRESENTAÇÃO DO CONHECIMENTO E RACIOCÍNIO

Na ciência da computação, dados podem ser definidos como registros puros, quantificáveis, sem qualquer análise. Por outro lado, a palavra conhecimento refere-se à um modelo capaz de descrever objetos, indicar quais ações executar ou que decisões tomar (REZENDE, 2003). No domínio da inteligência artificial (IA), o conhecimento pode ser definido como padrões aprendidos por algoritmos ou explicitamente construídos de maneira que possam ser interpretados por máquinas. Sendo assim, no contexto desta pesquisa, considera-se como conhecimento, todo o conhecimento modelado utilizando técnicas de inteligência artificial.

A representação do conhecimento (BRACHMAN; LEVESQUE, 2004) é uma área da inteligência artificial que se preocupa com a forma com que o conhecimento pode ser representado simbolicamente e manipulado de modo automático por algoritmos raciocinadores. Deste modo, uma quantidade finita de proposições é modelada, e, por isso, são necessários algoritmos raciocinadores a fim de executar formalmente inferências no conhecimento armazenado, possibilitando a criação de novas proposições.

Sendo assim, dispondo-se de uma estrutura de representação de conhecimento e de 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). O processo de raciocínio também pode ser definido como um cálculo aplicado sobre símbolos que representam proposições em vez de números (BRACHMAN; LEVESQUE, 2004).

Para entendermos como a arquitetura proposta na presente monografia será capaz de responder às consultas, serão detalhadas a seguir técnicas de representação de conhecimento e raciocinadores.

2.2.1. 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 surgiu uma série de conceitos relacionados à web semântica, trazendo a possibilidade de executar inferências automáticas na web (ALMEIDA, 2013). Com o impulsionamento da web semântica, 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; HAMOUDA; TANTAN; BOUGHZALA, 2016; PERAL et al., 2018).

Para Gruber (1996), ontologia é uma especificação de uma conceitualização. Nela, definições são ligadas a nomes de entidades através de relações e axiomas que descrevem e restringem a interpretação e o uso desses termos. Os objetos representados e suas relações fazem parte do vocabulário representacional da ontologia que, por sua vez, pode ser utilizado por programas baseados em conhecimento para representar e inferir sobre o conhecimento do domínio.

Nesse contexto, linguagens para representação do conhecimento podem ser utilizadas a fim de descrever uma ontologia, visando o armazenamento do conhecimento do domínio e facilitando a inferência de maneira automatizada. Sendo assim, nos últimos anos, várias linguagens foram desenvolvidas e estão sendo utilizadas para implementar ontologias como FLogic, XML, RDF, entre outras (CORCHO; GÓMEZ-PÉREZ, 2000). Dentre estas, destaca-se a Resource Description Framework (RDF), criada pela W3C como forma de descrever um recurso web, tendo como principal meta criar um mecanismo para descrever recursos sem fazer suposições sobre sua aplicação ou sobre a estrutura do documento que contém a informação (WORLD WIDE WEB CONSORTIUM et al., 1999).

Com o objetivo de manipular o conhecimento serializado utilizando linguagens de representação de conhecimento, existem 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 RDF, além de um mecanismo de armazenamento de triplas e uma API para manipulação de ontologias (THE APACHE SOFTWARE FOUNDATION, 2011).

Outra API de alto nível para manipulação de ontologias é a OWL API. Trata-se de uma série de padrões criados na linguagem de programação Java que dão suporte à extração, validação e visualização das definições de sintaxe do arquivo da ontologia em formato RDF ou Web Ontology Language (OWL) (HORRIDGE; BECHHOFER, 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 (HORRIDGE; BECHHOFER; NOPPENS, 2007). Além disso, permite-se que o desenvolvedor utilize um nível de abstração mais alto ao inferir na ontologia, concentrando-se em sua manipulação em vez de tratar problemas como sintaxe e serialização de dados.

2.2.2. RACIOCINADORES

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

Segundo Khamparia e Pandey (2017), um algoritmo raciocinador é um programa de computador capaz de efetuar inferências lógicas a partir de fatos, asserções e axiomas, permitindo o raciocínio automatizado.

O HermiT (SHEARER; MOTIK; HORROCKS, 2008) é um raciocinador baseado em lógica descritiva (do inglês *description logic* – DL) que tem como objetivo ser eficiente e implementar diversas melhorias que o permitem trabalhar com ontologias maiores e mais complexas. De maneira similar, o raciocinador Pellet (SIRIN; PARSIA, 2004), escrito na linguagem de programação Java, é capaz de raciocinar em ontologias utilizando lógica descritiva e expõe uma interface de comunicação com as funções de raciocínio na linguagem Java, linha de comando e formulário web. O raciocinador Pellet pode ser usado para classificação, consultas e testes automatizados, entre outras aplicações. Além disso, ele também pode ser empregado na localização de inconsistências na descrição dos conceitos de ontologias, ajudando o engenheiro de conhecimento a encontrar erros nestas.

Raciocinadores baseados em lógica descritiva, como o HermiT e o Pellet, geralmente utilizam algoritmos que transformam a tarefa de raciocínio em uma ou mais verificações de consistência da ontologia. Estas verificações têm o objetivo de checar a consistência das asserções e cláusulas carregadas (GLIMM et al., 2014). Dessa forma, todas as outras tarefas de raciocínio, como classificação e realização, podem ser reduzidas a tarefas mais simples de verificação de consistência (SIRIN et al., 2007). Sendo assim, otimizações que reduzam o número de verificações de consistência tendem a melhorar significativamente a performance do algoritmo raciocinador.

Trabalhos recentes mostram que existe uma ampla oportunidade para pesquisas na área, visto que ainda não existem raciocinadores capazes de oferecer suporte completo ao raciocínio em linguagens para representação do conhecimento propostas recentemente (KHAMPARIA; PANDEY, 2017). Por outro lado, há um número considerável de raciocinadores implementados que utilizam diversas formas de otimização e suporte a diferences linguagens para representação do conhecimento (MISHRA; KUMAR, 2011).

2.3. DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

A descoberta de conhecimento em banco de dados (do inglês *knowledge discovery in databases* – KDD), área que integra conceitos da inteligência artificial e de banco de dados, pode ser definida como a extração de informação implícita, anteriormente desconhecida, a partir de sistemas de bancos de dados (FRAWLEY; PIATETSKY-SHAPIRO; MATHEUS, 1992).

Outra definição, presente em Maimon e Rokach (2010), diz que o processo de descoberta de conhecimento em banco de dados é um procedimento automático e exploratório, com o objetivo de modelar o conhecimento existente em grandes e complexos bancos de dados, identificando padrões válidos, úteis e compreensíveis.

Segundo Frawley, Piatetsky-Shapiro e Matheus (1992), o processo de descoberta de conhecimento possui quatro características principais:

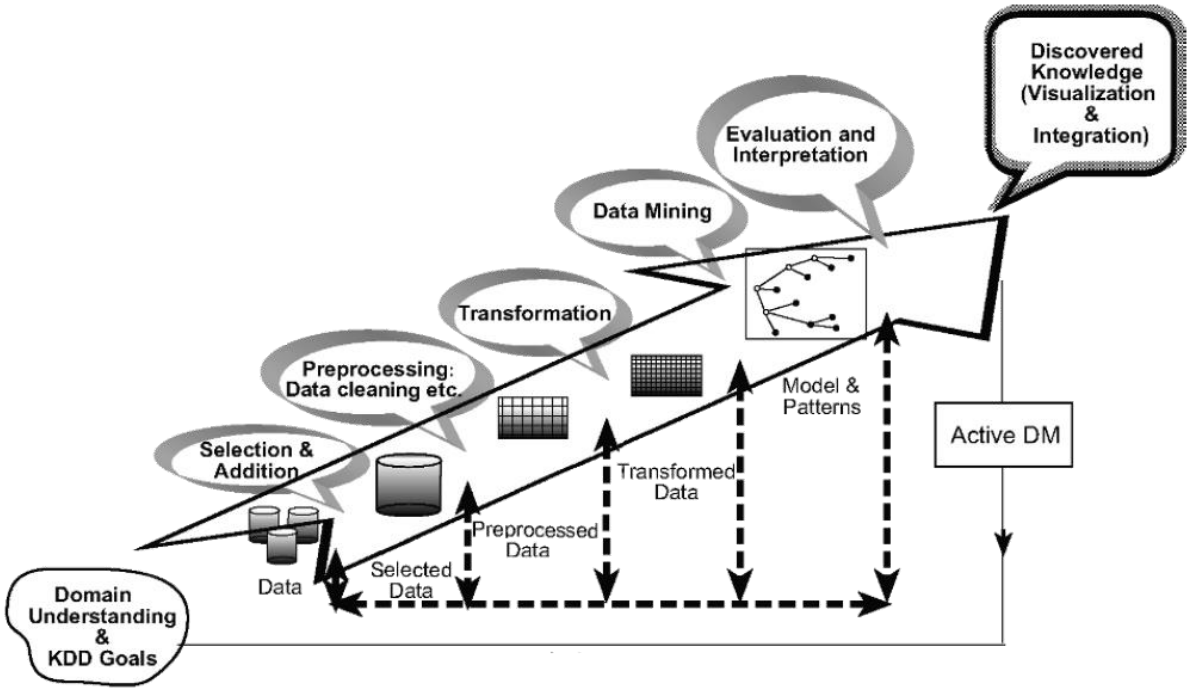
- **Eficiência:** O processo é eficiente, processado de forma automatizada por computador, e pode ser executado em grandes bancos de dados;
- **Precisão:** As descobertas representam o conteúdo do banco de dados de maneira precisa;
- **Resultados interessantes:** Os resultados obtidos através do processo de descoberta

de conhecimento seguem as regras e intenções do usuário, visto que seus interesses influenciam os padrões extraídos;

- **Extração de conhecimento em alto nível:** Geralmente, o conhecimento extraído é expresso em uma linguagem de alto nível, facilitando o entendimento por humanos.

O processo de descoberta de conhecimento em banco de dados é interativo e pode ser descrito em uma sequência de passos ordenados que começa com o estudo do domínio e termina com a utilização do conhecimento extraído. A Figura 3 mostra um resumo dos passos executados em um processo de descoberta de conhecimento em bancos de dados.

Figura 3 - Processo de descoberta de conhecimento em banco de dados



Fonte: Adaptado de Maimon e Rokach (2010).

Inicialmente, são definidos os objetivos da execução do processo de KDD através do estudo do domínio do problema. Em seguida, são selecionados os dados que serão utilizados no processo de descoberta e, assim, são aplicadas técnicas de pré-processamento, limpeza dos dados, tratamento de dados faltantes e transformações como redução de dimensionalidade, seleção de atributos, normalização e outras técnicas. A partir dos dados preparados, é definida a tarefa de mineração de dados e um algoritmo apropriado é escolhido de acordo com o

problema. Após sua escolha, o algoritmo é executado e tem seus parâmetros refinados diversas vezes até que se chegue a um resultado satisfatório. Por fim, é feita uma avaliação do conhecimento extraído, e este pode passar a ser utilizado em outros sistemas (MAIMON; ROKACH, 2010).

2.3.1. MINERAÇÃO DE DADOS

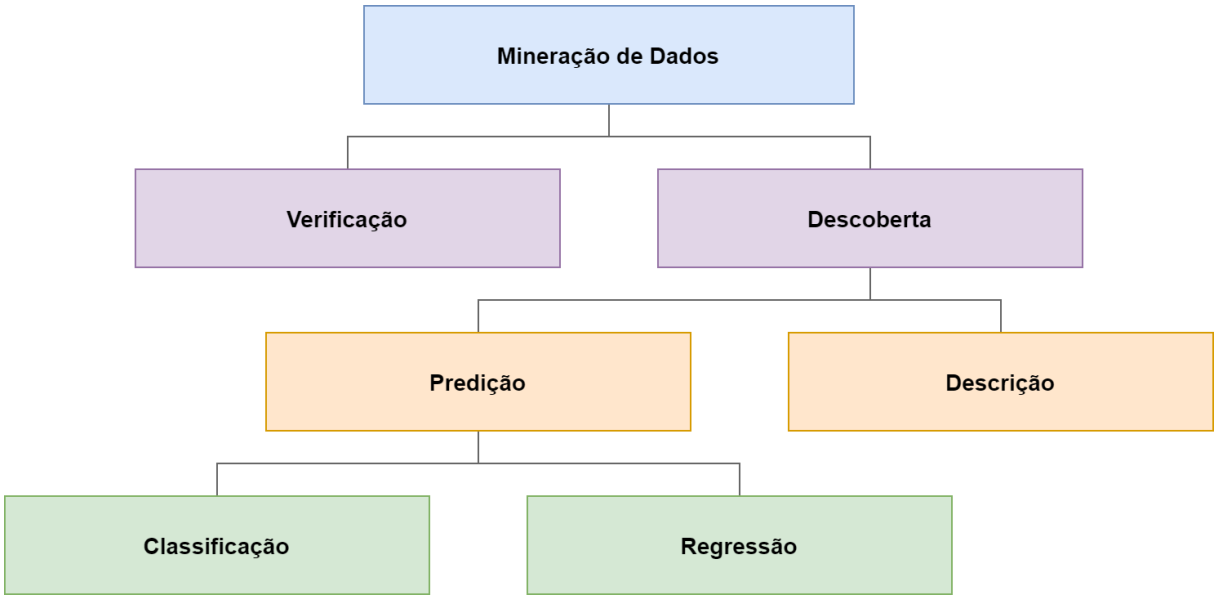
A mineração de dados, uma das principais etapas do processo de descoberta de conhecimento em banco de dados, é descrita por Maimon e Rokach (2010) como a execução de algoritmos que exploram os dados a fim de desenvolver modelos e encontrar padrões previamente desconhecidos.

De forma semelhante, uma outra característica bastante utilizada na definição do termo “mineração de dados” é que este é um processo de descoberta de padrões em dados. O processo pode ser completamente automático ou semiautomático, porém os padrões encontrados devem ter algum significado, de forma a trazer algum benefício para a organização (WITTEN et al., 2011).

Muitas vezes, o termo “mineração de dados” pode ser interpretado ou utilizado como equivalente ao termo “descoberta de conhecimento em banco de dados”, embora a mineração de dados propriamente dita é apenas uma das etapas deste processo de descoberta. Sendo assim, de forma mais abrangente, a mineração de dados é o processo de descoberta de padrões e conhecimento a partir de grandes quantidades de dados (HAN; PEI; KAMBER 2011). As fontes de dados podem ser diversas, desde bancos de dados relacionais até arquivos de textos brutos.

As tarefas de mineração de dados podem ser classificadas de maneira hierárquica de acordo com seu paradigma e objetivos. A Figura 4 mostra os principais paradigmas da mineração de dados e como eles se relacionam, segundo Maimon e Rokach (2010). Tarefas de verificação têm o objetivo de validar hipóteses. Por outro lado, tarefas de descoberta tentam identificar padrões nos dados que podem ser utilizados para sua predição ou descrição. De maneira similar, Han, Pei e Kamber (2011) dividem as tarefas de mineração de dados em duas categorias: descritivas (tarefas que descrevem as propriedades dos dados em um conjunto de dados alvo) e preditivas (tarefas que realizam induções baseadas nos dados com o objetivo de fazer predições).

Figura 4 - Taxonomia das principais tarefas de mineração de dados



Fonte: Adaptado de Maimon e Rokach (2010).

2.3.2. APRENDIZAGEM DE MÁQUINA

Com o aumento da quantidade de dados coletados em diversas áreas do conhecimento, a aprendizagem de máquina vem sendo utilizada para solucionar diversos problemas, inclusive na área da saúde, como na predição do grau de risco de doenças, em diagnósticos automatizados baseados em exames de imagem e na interpretação de dados de sensores (CHEN, M. et al., 2017; STUNTEBECK et al., 2008; ZHOU et al., 2002).

O processo de aprendizado pode ser definido como o processo de converter experiências anteriores em conhecimento. Técnicas de aprendizagem de máquina têm o objetivo de fazer com que programas de computador extraiam conhecimento útil de uma fonte de dados. A entrada do processo de aprendizagem de máquina é chamada de conjunto de treinamento (que representa a experiência), e a saída é geralmente um outro programa de computador capaz de realizar alguma tarefa (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Segundo Russell e Norvig (2013), a maioria das pesquisas atuais em aprendizagem de máquina utiliza-se de uma representação fatorada, composta por um vetor de valores e atributos, e sua saída pode ser um valor numérico contínuo ou um valor numérico discreto. Os autores contextualizam a aprendizagem de máquina dentro do paradigma de agentes inteligentes, que podem ser definidos com entidades capazes de perceber o ambiente do problema através de

sensores e agir sobre este através de atuadores. Neste contexto, a aprendizagem de máquina pode ser classificada em três grupos baseados no *feedback* fornecido:

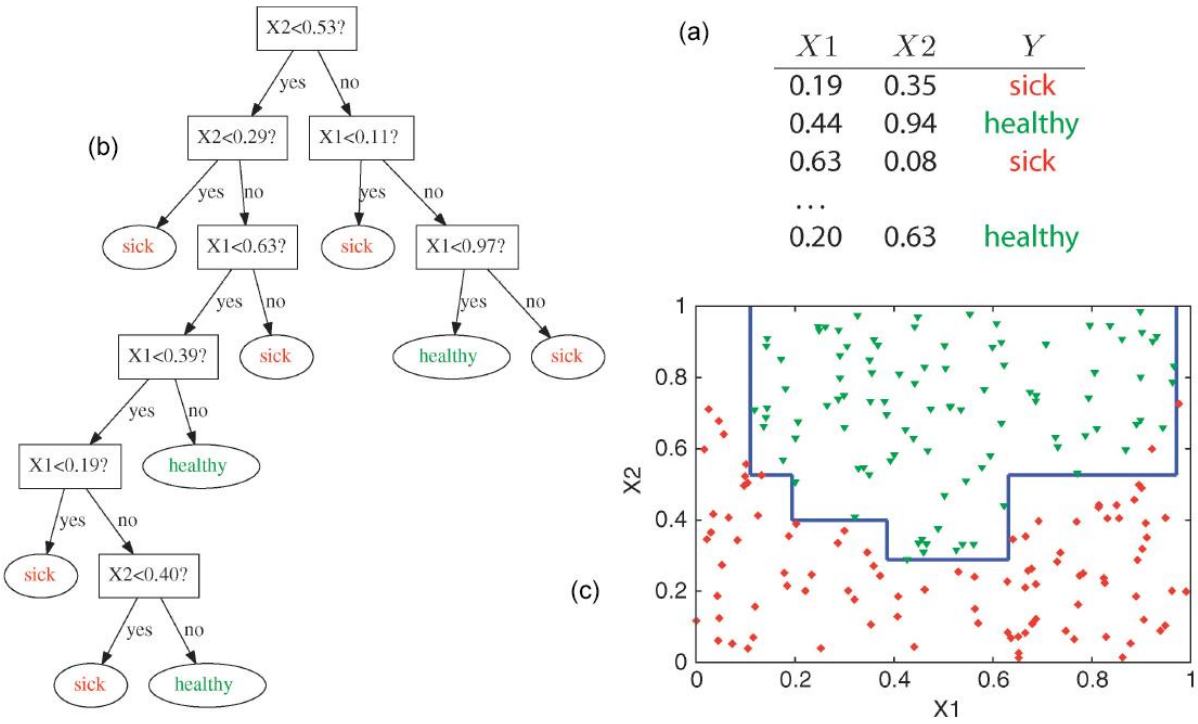
- **Aprendizagem supervisionada:** Utiliza-se exemplos de pares de entrada e saída para fazer o mapeamento dos valores de entrada para a saída, com o objetivo de encontrar uma função capaz de generalizar a solução para o problema;
- **Aprendizagem não supervisionada:** Nesse tipo de aprendizagem, tenta-se aprender padrões nos dados de entrada mesmo sem ter acesso a um *feedback* explícito. Normalmente, tarefas de agrupamento são resolvidas utilizando aprendizagem de máquina não supervisionada;
- **Aprendizagem por reforço:** Utiliza-se técnicas de tentativa e erro, baseadas em uma série de recompensas e reforços fornecidos a partir das ações executadas.

Existem diversos algoritmos de aprendizagem de máquina, sendo um dos mais populares o algoritmo de aprendizagem em árvore de decisão, que usa uma tabela de exemplos e, a partir de uma estratégia gulosa, é capaz de encontrar uma solução aproximada e representá-la com uma árvore. Nesse contexto, uma árvore de decisão representa uma função que toma como entrada uma lista de valores de atributos e retorna um valor de saída, uma "decisão". Cada nó da árvore representa um teste no valor de um dos atributos de entrada e as ramificações do nó são os possíveis resultados do teste. Um nó-folha desta árvore representa a decisão que será retornada pela função. Uma das vantagens do uso de árvores de decisão é que seu funcionamento é simples e, muitas vezes, o conhecimento modelado pode ser diretamente interpretado por humanos (RUSSELL; NORVIG, 2013).

A Figura 5 mostra um exemplo de problema de aprendizagem supervisionada, aplicado à área da saúde e resolvido por meio de uma árvore de decisão. No mundo real, esse problema poderia ter a finalidade de ajudar um profissional de saúde a distinguir entre um paciente saudável, em verde, e um doente, em vermelho, utilizando os dados coletados do próprio paciente. O objetivo do problema é encontrar uma função que diferencie a saída Y, representada pelas cores verde (saudável) e vermelho (doente), utilizando as entradas X1 e X2, que são os dados do paciente. A tabela (a) mostra os exemplos de entrada e saída, onde cada linha representa uma instância do problema. O diagrama (b) apresenta uma representação gráfica da árvore de decisão aprendida onde os retângulos mostram os testes realizados nos dados, as ligações denotam os possíveis resultados dos testes e os círculos indicam o resultado retornado. Além disso, é possível ver na imagem o gráfico de dispersão (c), uma representação alternativa

do problema em que a linha azul mostra os limites do classificador representado pela árvore (GEURTS; IRRTHUM; WEHENKEL, 2009).

Figura 5 - Problema de classificação resolvido utilizando árvore de decisão



Fonte: Adaptado de Geurts, Irrthum e Wehenkel (2009).

Em alguns casos, é possível treinar um modelo que tenha uma performance excelente para o banco de dados de treinamento, mas sua performance para exemplos nunca vistos não é satisfatória, ou seja, o modelo gerado não possui a capacidade de generalizar. Este fenômeno é chamado de superadaptação e pode ser evitado, na maioria das vezes, pelo provimento de mais dados de treinamento (SHALEV-SHWARTZ; BEN-DAVID, 2014).

Outra forma de reduzir as chances de superadaptação é empregar a técnica de *data augmentation*, que consiste em produzir novos exemplos baseados no conjunto de treinamento a fim de aumentar e, artificialmente, diversificar a quantidade de dados do conjunto de treinamento (WONG et al., 2016).

Em problemas cujo objetivo é a predição, com o intuito de validar os modelos criados, a técnica de validação cruzada tenta avaliar sua capacidade de generalização a partir de um conjunto de dados. Nesta forma de treinamento e avaliação, o conjunto de dados é dividido em subconjuntos mutuamente exclusivos e, por diversas vezes, é efetuado o treinamento e o erro é calculado. Cada execução do algoritmo é denominada dobra. Sendo assim, uma validação

cruzada com três dobras indica que o processo foi repetido três vezes, cada uma em um subconjunto de validação e treinamento diferentes. Técnicas como a validação cruzada de múltiplas dobras ajudam o projetista a entender o quão confiável está o modelo e se este é capaz de generalizar (BAKER; ISOTANI; CARVALHO, 2011).

No contexto do suporte à decisão, testes de performance indicam que regras aprendidas automaticamente por meio de algoritmos de aprendizagem de máquina muitas vezes podem superar regras criadas manualmente por especialistas (WITTEN et al., 2011). Desde a década de 1980, pesquisas relacionadas à aquisição de conhecimento de forma automática demonstram que, em alguns casos, os resultados se mostram melhores do que a representação direta do conhecimento provido por especialistas do domínio. Um exemplo clássico foi detalhado por Michalski e Chilausky (1980), no contexto do diagnóstico da doença da soja, que demonstra como uma técnica de aquisição de conhecimento permitiu a criação de regras mais precisas do que as regras utilizadas pelos especialistas na época.

Na área da saúde, muitos estudos vêm sendo desenvolvidos com o objetivo de integrar o conhecimento obtido de forma automatizada com o conhecimento de especialistas do domínio (ALONSO, F. et al., 2002; ESFANDIARI et al., 2014). Esta integração permite a criação de regras a partir da grande quantidade de dados gerados e, ao mesmo tempo, conta com o suporte de especialistas do domínio para, entre outras tarefas, verificar e melhorar o conhecimento descoberto a partir dos dados.

2.3.2.1. REDES NEURAIS ARTIFICIAIS

Uma rede neural artificial (RNA) é formada por camadas de neurônios e sinapses. É capaz de adquirir conhecimento através de exemplos, ajustando os pesos de ligações entre os neurônios, a partir de um processo de aprendizagem (HAYKIN, 1994).

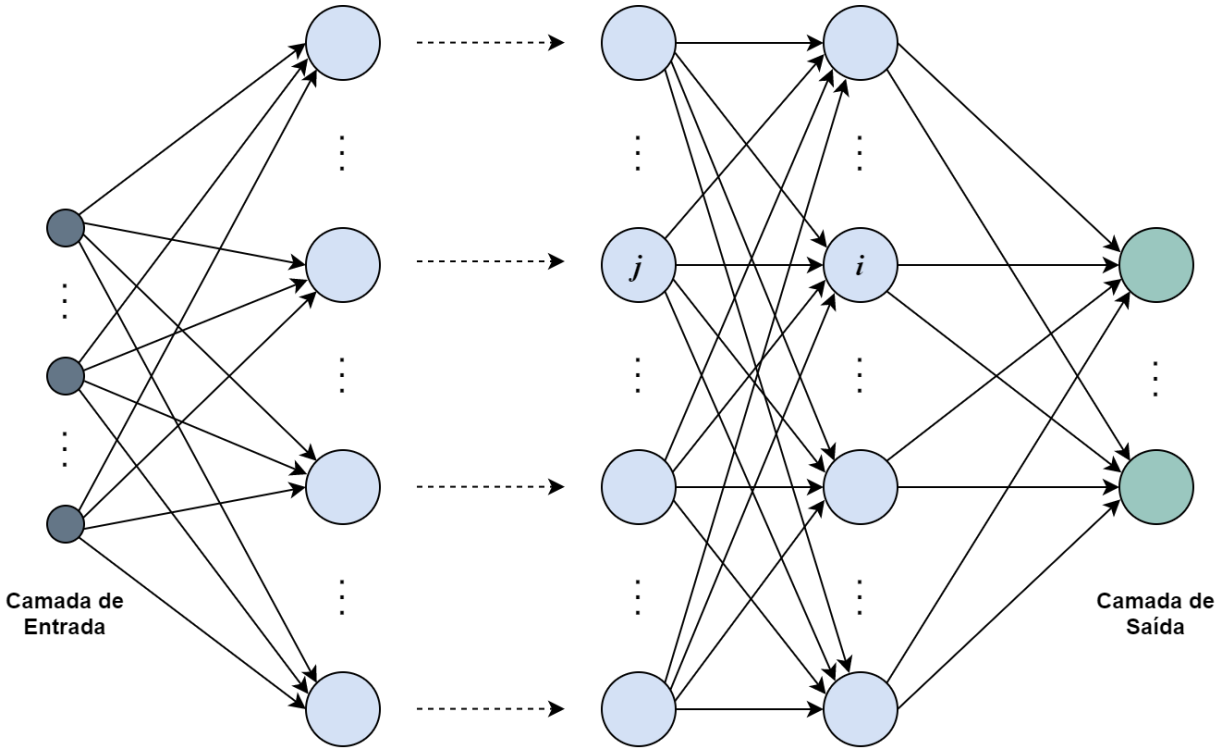
O funcionamento das redes neurais artificiais é inspirado pelo funcionamento do cérebro humano. Dessa maneira, as RNAs são compostas por nós que, por sua vez, estão dispostos em uma ou mais camadas interligadas por uma grande quantidade de conexões, normalmente unidirecionais. Na grande maioria das RNAs, as conexões são associadas a pesos que são ajustados durante o seu treinamento (BRAGA; CARVALHO; LUDERMIR, 2000).

Existem diversos tipos de arquiteturas de redes neurais artificiais, tendo cada uma delas diferentes vantagens e desvantagens ao serem aplicadas a certos tipos de problemas. Uma das arquiteturas de RNA mais utilizadas é a das redes neurais profundas. Estas podem ser definidas

como redes neurais em que os nós estão organizados em camadas, e em que pelo menos uma destas camadas não está diretamente ligada à entrada ou à saída. Em outras palavras, são RNAs que possuem pelo menos uma camada oculta.

Em RNAs com a arquitetura *feed-foward*, as conexões são unidirecionais acíclicas, partindo da camada de entrada em direção à camada de saída. Um exemplo de rede artificial profunda do tipo *feed-foward* pode ser visto na Figura 6. Os círculos representam os nós da RNA enquanto as setas representam as ligações entre eles. Durante a execução de uma instância, a camada de entrada recebe os dados da instância que está sendo executada e os propaga para que sejam processados pela camada seguinte. O processo se repete até que um resultado seja obtido na camada de saída (BEEMAN, 2000b).

Figura 6 - Exemplo de rede neural artificial profunda feed-foward



Fonte: Adaptado de Beeman (2000b).

Os principais benefícios do uso desse tipo de arquitetura são sua simplicidade e a possibilidade da utilização do algoritmo *backpropagation*, um algoritmo de aprendizagem que permite o aprendizado supervisionado dos pesos das conexões através de exemplos (BEEMAN, 2000b). O procedimento para o ajuste dos pesos das ligações da rede neural no algoritmo de *backpropagation* é descrito por Beeman (2000a) e pode ser visto na Figura 7.

Figura 7 - Passos do algoritmo backpropagation

1. Inserir a lista de entradas do primeiro item do conjunto de dados de treinamento na camada de entrada;
2. Efetuar uma soma ponderada dos valores de entrada e os propagar para a próxima camada, calculando suas funções de ativação;
3. Inserir os valores de ativação na próxima camada e repetir o passo (2) para todas as camadas até que se alcance a camada final, a camada de saída;
4. Comparar o valor final da ativação na camada de saída com o valor-alvo, o valor da supervisão para aquela instância;
5. Propagar o erro para trás, utilizando o erro da camada de saída para calcular a variação nos pesos da camada precedente;
6. Utilizar as diferenças encontradas para calcular as diferenças nas camadas anteriores, repetindo o processo até que a primeira camada, a camada de entrada, seja atingida;
7. Calcular as alterações nos pesos das ligações e bias;
8. Atualizar os pesos da rede com base nos erros. Caso o treinamento esteja sendo feito baseado em épocas, os pesos são atualizados apenas no final da época utilizando a média dos ajustes de cada instância;
9. Repetir o processo até que o erro médio quadrado seja menor que o valor do critério de parada do algoritmo.

Fonte: Adaptado de Beeman (2000a).

Redes neurais, como qualquer outra técnica de aprendizagem de máquina, podem sofrer o problema da superadaptação, principalmente se possuírem uma grande quantidade de parâmetros ou se o conjunto de dados de treinamento for desbalanceado ou pequeno. Nesse contexto, a técnica de *dropout* pode ser utilizada para diminuir a tendência de superadaptação da rede. Essa técnica consiste na eliminação aleatória de alguns neurônios da rede e suas conexões durante o treinamento (SRIVASTAVA et al., 2014).

A arquitetura da RNA escolhida tem um grande impacto no resultado do treinamento e, por isso, algoritmos de buscas automatizados por hiperparâmetros podem ser usados para encontrar a melhor configuração de rede para um problema específico. Entre essas técnicas de busca, destacam-se duas: a busca em grade (do inglês *grid-search*) e a busca aleatória de parâmetros. A técnica de busca em grade consiste na criação de várias arquiteturas de rede,

funções de ativação e outras configurações utilizando intervalos predefinidos de busca, tratando cada combinação de parâmetros igualmente durante o aprendizado (BUITINCK; LOUPPE; BLONDEL; PEDREGOSA; MUELLER; GRISEL; NICULAE; PRETTENHOFER; GRAMFORT; GROBLER; LAYTON; VANDERPLAS; JOLY; HOLT; et al., 2013). Por outro lado, o método de busca aleatória de hiperparâmetros é uma estratégia que, em casos em que existem vários conjuntos de dados e algoritmos de aprendizagem, pode ser mais eficiente do que a utilização da busca em grade, visto que nem todos os parâmetros das redes são igualmente importantes (BERGSTRA; BENGIO, 2012).

Em conclusão, as redes neurais artificiais, em conjunto com os demais algoritmos de aprendizagem, possuem a capacidade de aproximar mapeamentos não lineares complexos através de um conjunto de exemplos de entradas. Com isso, são capazes de encontrar soluções que normalmente são difíceis de modelar por meio de uma abordagem paramétrica clássica (HUANG; ZHU; SIEW, 2006). Devido a essa capacidade, as RNAs estão sendo usadas em diversas áreas do conhecimento, incluindo o suporte à decisão clínica, como abordado por Lobo (2017).

2.4. INFORMÁTICA EM SAÚDE

O termo “informática em saúde” é abrangente e pode se referir ao prontuário eletrônico do paciente, ao processamento de sinais biológicos, a sistemas de informação na saúde, à inteligência artificial em saúde e até a programas para treinamento de especialistas (BRASIL, 2008).

Segundo Hoyt (2009), a informática em saúde pode ser definida como o campo da ciência que lida com recursos, equipamentos e métodos formais para otimizar o armazenamento, a leitura e o gerenciamento de informações médicas na resolução de problemas e na 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 deste ramo (CAMPOS, 2013). Por exemplo, o trabalho apresentado por Veinot et al. (2019) discute o potencial da informática em saúde para reduzir as desigualdades existentes no acesso à saúde por certos grupos socialmente marginalizados. Outros trabalhos na área focam-se na disponibilização de meios e serviços baseados em computador para gerenciar atividades

relacionadas diretamente ao tratamento de saúde de pacientes, como, por exemplo, o trabalho publicado por Silva (2019), que propõe uma plataforma para orquestração de serviços visando a permitir que profissionais de saúde criem seus próprios *workflows* contendo as tarefas necessárias para a execução de planos de cuidados para pacientes.

Sendo assim, a informática em saúde visa a melhorar a qualidade de serviços de saúde, reduzindo custos e permitindo a troca de informações médicas (HOYT, 2009).

2.4.1. SISTEMAS DE SUPORTE À DECISÃO CLÍNICA

No contexto da informática em saúde, destacam-se os sistemas de suporte à decisão clínica (SSDC). Estes são caracterizados, conforme Greenes (2011), pelo uso do computador para a obtenção de conhecimento relevante na área da saúde e sobre o bem-estar do paciente.

Sistemas de suporte à decisão clínica desenvolvidos por meio de guias para a prática clínica como fonte de conhecimento podem ajudar durante a tomada de decisão por recomendar ações, padronizar a execução de certos tratamentos e identificar quais dados do paciente são realmente relevantes para seu diagnóstico e encaminhamento (BERNER, 2007; GREENES, 2011).

Segundo Greenes (2011), a maioria dos SSDC possui algumas características em comum, como: a) utilização de algum meio de inferência, algoritmo ou método; b) intenção de deixar os dados dos pacientes mais acessíveis ou auxiliar no processo de tomada de decisão; c) provisão do suporte à decisão para um humano, que pode ser um médico, enfermeiro ou outro indivíduo que necessite do suporte à decisão clínica; d) o resultado provido é geralmente a recomendação de uma ação a ser tomada.

Nos últimos anos, pesquisas na área mostram que sistemas de suporte à decisão clínica permitem que profissionais de saúde gerenciem grandes quantidades de informação de uma maneira efetiva e eficiente (SIU et al., 2019). Por exemplo, o trabalho apresentado por Souffront et al. (2019) demonstra a utilização de um sistema de suporte à decisão clínica capaz de alertar enfermeiras para que monitorem a pressão sanguínea de pacientes que deram entrada na emergência com valores anormais, concluindo que sistemas semelhantes podem auxiliar no exercício das atividades médicas. De maneira similar, Seneviratne et al. (2019) explora a utilização de SSDC em conjunto com o conceito de web semântica, tecnologias com potencial de representar e ligar conceitos da literatura científica à pratica médica, permitindo uma adaptação mais rápida a mudanças de informação presentes nas *guidelines* médicas.

Dessa maneira, SSDC vêm ajudando profissionais de saúde a melhorar suas práticas médicas e, consequentemente, têm contribuído para o aumento da eficiência no diagnóstico e no tratamento de seus pacientes.

2.5. CONSIDERAÇÕES FINAIS

O presente capítulo apresentou uma visão geral dos conceitos utilizados no desenvolvimento desta pesquisa. Sendo assim, foram apresentadas definições de diversas áreas do conhecimento, como, por exemplo, conceitos da área de representação do conhecimento e raciocínio, essenciais para o entendimento de como o conhecimento pode ser processado dentro de um sistema computacional.

No domínio de arquiteturas de software, foram descritos os principais conceitos da área como, por exemplo, as definições e formas de utilização de arquiteturas de referência. Além disso, foram apresentados diferentes paradigmas oriundos do conceito de arquiteturas orientadas a serviço, no qual estão inseridas as arquiteturas baseadas em conhecimento como serviço, paradigma utilizado na proposta apresentada. Tais conceitos serão fundamentais na definição da arquitetura H-KaaS, contribuindo para sua melhor padronização, de modo que possa ser utilizada por outros pesquisadores, tanto no desenvolvimento de novos sistemas de suporte à decisão clínica quanto na análise e comparação de sistemas existentes.

De maneira similar, foram abordados temas relacionados à descoberta de conhecimento em bancos de dados, permitindo uma melhor compreensão de como dados podem ser analisados no intuito de gerar conhecimento útil e capaz de ser interpretado por algoritmos raciocinadores.

Por fim, foram abordados temas necessários para o entendimento dos estudos de caso que serão utilizados durante a validação da arquitetura posposta, como informática em saúde e sistemas de suporte à decisão clínica.

Em resumo, este capítulo apresentou o contexto teórico onde esta pesquisa está inserida, com a finalidade de permitir uma maior compreensão dos capítulos seguintes e dos resultados apresentados.

3. TRABALHOS RELACIONADOS

Para o desenvolvimento da presente pesquisa, inicialmente foi feito um levantamento bibliográfico com o objetivo de analisar diversas arquiteturas e sistemas cuja finalidade é o compartilhamento de conhecimento e de dados em subdomínios da saúde. A fundamentação teórica e a análise das arquiteturas existentes certamente se mostrarão elementos imprescindíveis para a elaboração e o refinamento da arquitetura de referência proposta neste trabalho.

A partir da análise das arquiteturas preexistentes, pretende-se propor uma arquitetura de referência baseada no paradigma de conhecimento como serviço para o domínio da saúde, detalhando sistematicamente cada um de seus módulos e componentes. Os módulos da arquitetura serão especificados formalmente, sendo identificadas suas responsabilidades e interfaces de entrada e saída de dados ou conhecimento.

Uma revisão sistemática da literatura é um estudo de uma ou mais perguntas bem definidas, por meio de uma metodologia clara, utilizando-se de métodos explícitos para selecionar e avaliar de forma crítica estudos relevantes incluídos na revisão. Adicionalmente, uma revisão sistemática pode valer-se de métodos estatísticos para analisar e resumir os resultados dos estudos selecionados (MOHER et al., 2009).

Sendo assim, os trabalhos relacionados nesta pesquisa serão identificados através da revisão sistemática da literatura proposta neste capítulo. Seu objetivo é, além de identificar propostas parecidas, responder às questões de pesquisa apresentadas na Tabela 1. Além disso, serão listadas as principais características de cada trabalho selecionado a fim de compará-los com a arquitetura posposta nesta pesquisa.

Tabela 1 - Questões de pesquisa definidas para a revisão sistemática da literatura

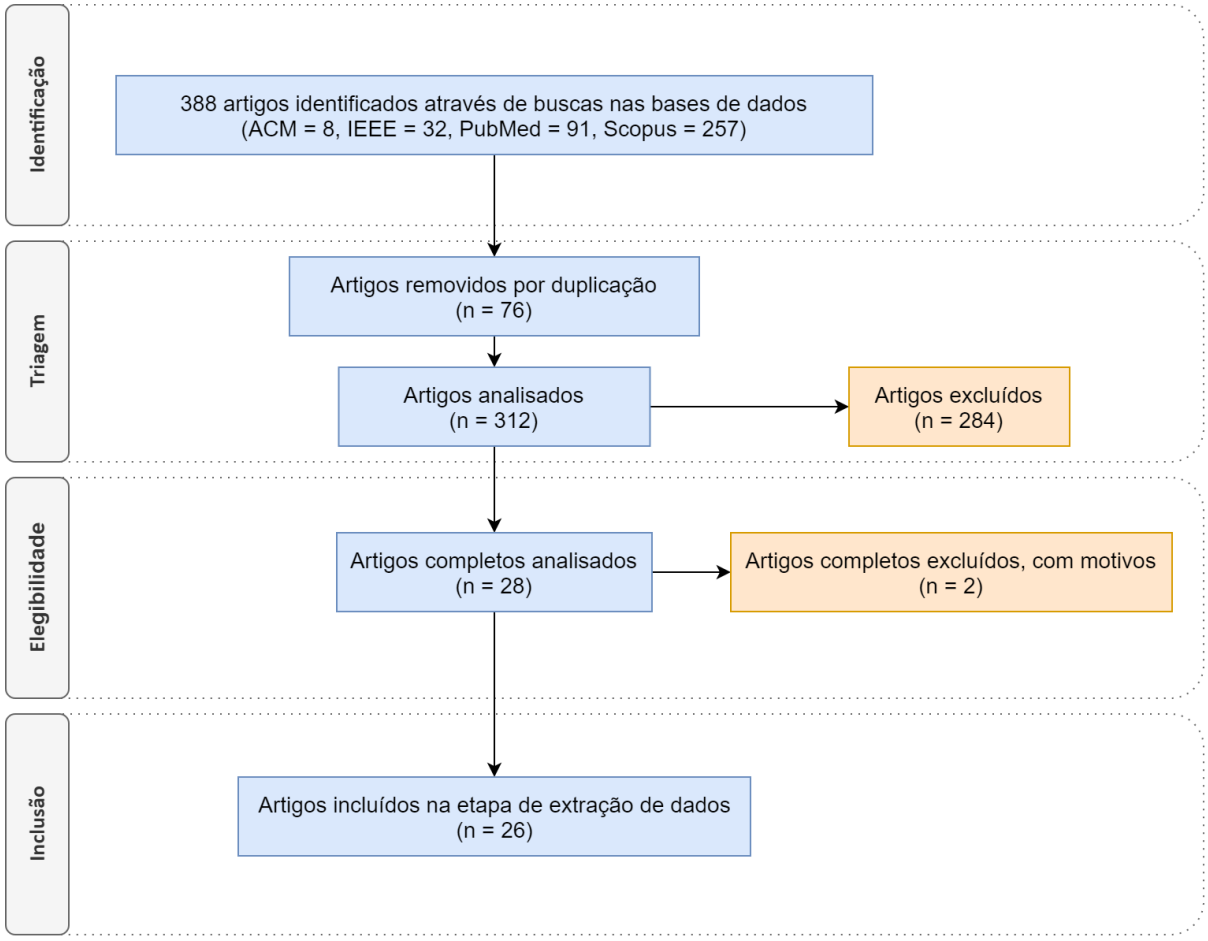
Questões de pesquisa	
QP-01	Quais são os principais componentes propostos pelas arquiteturas para compartilhamento de conhecimento na área da saúde?
QP-02	Como a comunicação entre os componentes da arquitetura ou paradigma proposto é feita?
QP-03	Quais formas de representação e descoberta de conhecimento são utilizadas em arquiteturas focadas no compartilhamento de conhecimento no domínio da saúde?
QP-04	Em quais subdomínios da saúde as arquiteturas de compartilhamento de conhecimento estão sendo utilizadas?
QP-05	Como o problema da persistência dos dados é tratado pelas arquiteturas?

QP-06	Como propostas de novas arquiteturas com foco no compartilhamento de conhecimento na área da saúde são validadas?

Fonte: Elaborado pelo autor.

A metodologia usada para a execução da revisão sistemática foi baseada nas diretrizes propostas por Kitchenham et al. (2009), tendo como principais etapas: definição das questões de pesquisa; definição dos critérios de inclusão e exclusão; extração de dados e análise; resultados e conclusões. Adicionalmente, a sumarização do processo de seleção dos trabalhos foi feita por meio do fluxograma recomendado pelo protocolo PRISMA (do inglês *preferred reporting items for systematic reviews and meta-analyses*), que consiste em uma lista de verificação com 27 recomendações e um fluxograma padronizado para a sumarização da revisão (MOHER et al., 2009). O fluxograma resultante da execução desta revisão, baseado no protocolo PRISMA, pode ser visto na Figura 8.

Figura 8 - Diagrama do processo de seleção de artigos da revisão sistemática



Fonte: Elaborado pelo autor.

3.1. QUESTÕES DE PESQUISA

Como primeira etapa de uma revisão sistemática, foram definidas as questões de pesquisa (QP), que serão respondidas ao fim do processo de revisão. A Tabela 1 mostra as questões de pesquisa desta revisão sistemática de literatura. Estas foram criadas com o objetivo de entender pontos importantes que precisam ser detalhados durante a elaboração da arquitetura de referência.

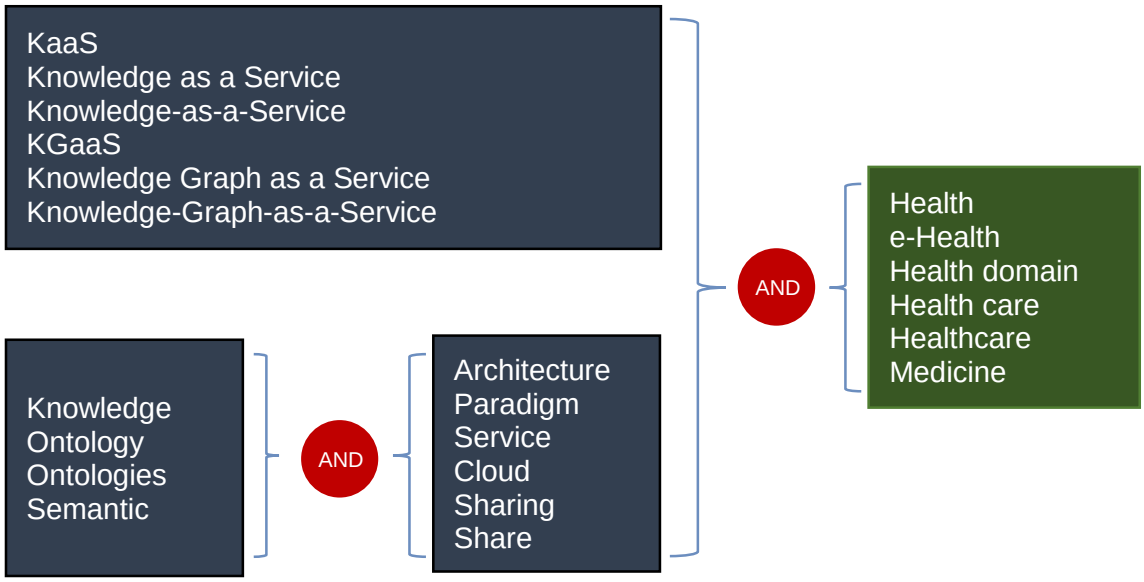
3.2. DEFINIÇÃO DAS CONSULTAS E BUSCADORES

Com o objetivo de definir as consultas a serem executadas nas bases de dados, primeiramente foram listadas, em inglês, as palavras-chave que representam os principais paradigmas de compartilhamento de conhecimento, suas siglas e sinônimos: “KaaS”; “Knowledge as a Service”; “Knowledge-as-a-Service”; “KGaaS”; “Knowledge Graph as a Service”; “Knowledge-Graph-as-a-Service”.

Em seguida, foram enumeradas expressões que, quando combinadas, pudessem representar paradigmas ou meios de compartilhamento do conhecimento, como “knowledge”, “ontology”, “ontologies” e “semantic”, em conjunto com as palavras “architecture”, “paradigm”, “service”, “cloud”, “sharing” e “share”.

Por fim, foi definida uma lista de palavras e expressões que possam representar o domínio da saúde e seus sinônimos, como: “health”; “e-health”; “health domain”; “health care”. Utilizando os operadores OR e AND, foi esboçado um diagrama que representa a *string* de busca a ser executada, apresentado na Figura 9.

Figura 9 - Agrupamento de palavras-chave usado na criação da string de busca



Fonte: Elaborado pelo autor.

Depois de várias interações decidiu-se que as palavras-chave e expressões referentes à arquitetura e paradigmas, em azul escuro, deveriam aparecer no título do trabalho, visto que a proposta ou aplicação da arquitetura deve ser o principal objetivo da pesquisa a ser analisada. Por outro lado, as palavras-chave relacionadas ao domínio da saúde, em verde, poderiam aparecer tanto no título quanto nos outros metadados da publicação, como resumo e palavras-chave. Essa medida garantiu uma maior abrangência das consultas, permitindo que publicações em subáreas da saúde também fossem retornadas durante as consultas. A *string* de busca genérica, criada a partir do diagrama, é apresentada na Figura 10.

Com o propósito de aumentar a abrangência da revisão sistemática da literatura, foram cuidadosamente escolhidos quatro engenhos de buscas, sendo dois específicos da área da computação e engenharias, um na área da saúde e um de propósito geral. Os engenhos de busca usados podem ser vistos na lista abaixo:

- **Específicos:** ACM[5], IEEE[6];
- **Saúde:** PubMed[7];
- **Abrangência geral:** Scopus[8].

[5] https://dl.acm.org/
[6] https://ieeexplore.ieee.org/
[7] https://www.ncbi.nlm.nih.gov/pubmed/
[8] https://www.scopus.com/

Figura 10 - String de busca genérica utilizada como base para as consultas

```
(("kaas" OR "knowledge as a service" OR "knowledge-as-a-service"
OR "kgaas" OR "knowledge graph as a service" OR "knowledge-graph-
as-a-service") OR (("knowledge" OR "ontology" OR "ontologies" OR
"semantic") AND ("architecture" OR "paradigm" OR "service" OR
"cloud" OR "sharing" OR "share"))) AND ("health" OR "e-health" OR
"health domain" OR "health care" OR "healthcare" OR "medicine"))
```

Fonte: Elaborado pelo autor.

O buscador Scopus foi escolhido por ser considerado o maior banco de dados de *abstracts* e citações do mundo, abrangendo mais de 5.000 editores internacionais de diferentes áreas do conhecimento como tecnologia, medicina, ciências sociais, arte, entre outras (ELSEVIER, 2017). Não há garantias de que todas as publicações de um determinado editor estejam indexadas, mesmo que este editor esteja na lista de editores rastreados pelo Scopus. Sendo assim, foi necessária a adição dos engenhos de busca específicos da área da computação e da saúde.

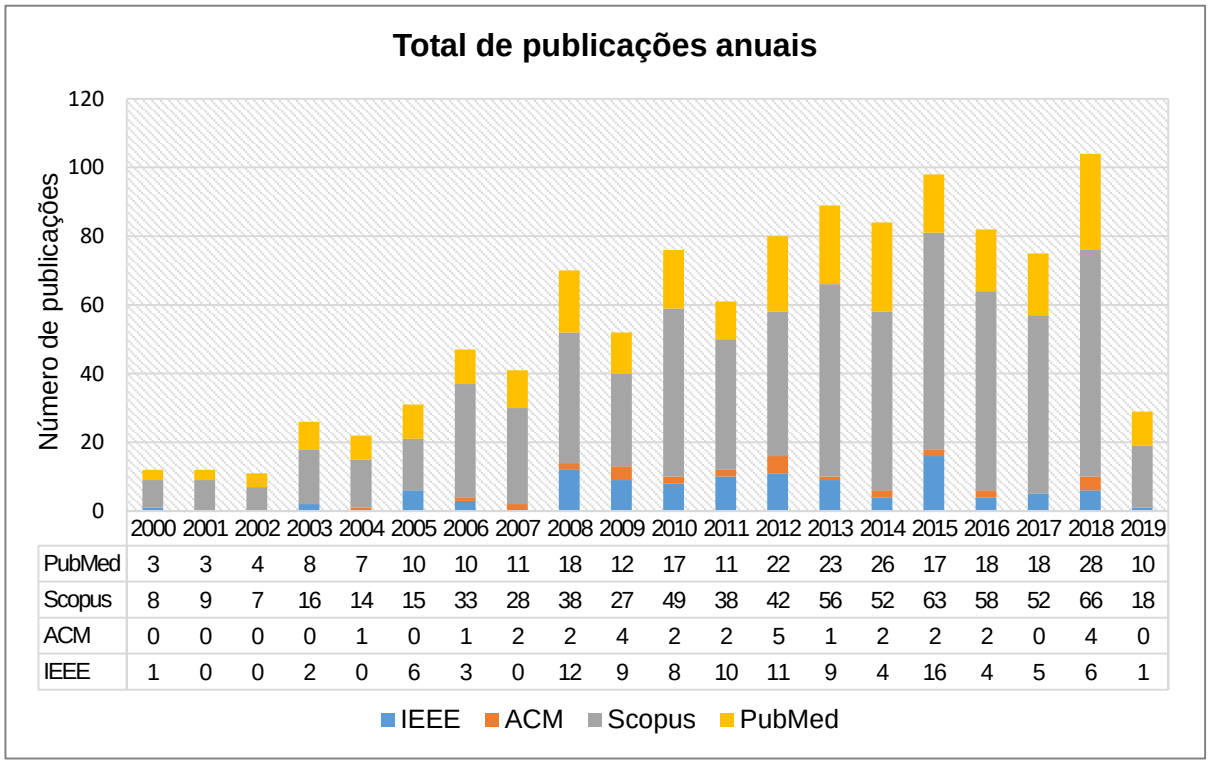
As *strings* de busca foram adaptadas para cada engenho de busca de acordo com a sintaxe aceita em sua página de busca avançada. As consultas específicas executadas em cada base de dados estão listadas no Apêndice A.

Com o intuito de entender como a área de estudo desta pesquisa foi evoluindo ao longo dos últimos anos, nenhuma data limite foi especificada inicialmente, e foram coletadas informações sobre a quantidade de artigos publicados ano a ano. Adicionalmente, devido à abrangência do banco Scopus, foi incorporado em sua *string* de busca o filtro pela língua inglesa e domínio da medicina ou computação.

A Figura 11 apresenta um comparativo anual das publicações retornadas por cada engenho de busca a partir do ano 2000, mostrando que houve um aumento significativo no total de trabalhos publicados a partir de 2006. Além disso, conforme esperado, devido ao fato de o engenho de busca Scopus indexar várias bibliotecas de publicações, incluindo ACM e IEEE, este retornou mais publicações que os demais, possuindo 62% dos trabalhos publicados entre 2000 e maio de 2019. É importante notar que a contagem dos trabalhos retornados pelos engenhos de busca foi feita antes da etapa de remoção de duplicados, sendo assim, um mesmo

trabalho poderia ser contabilizado múltiplas vezes caso esse aparecesse em múltiplo engenhos de busca.

Figura 11 - Comparativo anual de publicações retornadas pelos engenhos de busca



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
PubMed	3	3	4	8	7	10	10	11	18	12	17	11	22	23	26	17	18	18	28	10
Scopus	8	9	7	16	14	15	33	28	38	27	49	38	42	56	52	63	58	52	66	18
ACM	0	0	0	0	1	0	1	2	2	4	2	2	5	1	2	2	2	0	4	0
IEEE	1	0	0	2	0	6	3	0	12	9	8	10	11	9	4	16	4	5	6	1

Fonte: Elaborado pelo autor.

3.3. CRITÉRIOS DE INCLUSÃO E EXCLUSÃO

A partir do estudo inicial da quantidade de artigos ano a ano, foram analisados os possíveis períodos de abrangência para a revisão sistemática da literatura, sendo definido o período de 1º de janeiro de 2015 a 31 de maio de 2019 para sua execução, como pode ser visto na Tabela 2.

Desta maneira, foi utilizada uma abordagem de quatro etapas para a seleção dos trabalhos relevantes para esta pesquisa. Estas podem ser listadas, em sequência, como: a) Remoção de duplicados; b) Análise dos títulos; c) Análises dos títulos, resumos e palavras-chave; d) Leitura do texto completo dos trabalhos.

Na primeira etapa, a de remoção de duplicados, uma análise foi feita utilizando os títulos e, assim, os trabalhos com títulos repetidos foram removidos. Um total de 76 artigos foi excluído nesta etapa.

Tabela 2 - Total de publicações em cada buscador agrupadas por períodos

BUSCADOR	Número de Publicações				
	Últimos 2 anos (2018-2019)	Últimos 5 anos (2015-2019)	Últimos 10 anos (2009-2019)	Desde 2000 (2000-2019)	Todos (1970-2019)
IEEE	7	32	74	107	110
ACM	4	8	20	30	31
Scopus	84	257	494	689	761
PubMed	38	91	190	276	305
TOTAL	133	388	778	1102	1207

Fonte: Elaborado pelo autor.

A segunda e terceira etapas foram executadas em sequência, por dois revisores independentes, tendo sido utilizados os critérios de inclusão e exclusão a fim de selecionar os artigos relevantes. Para situações em que houve discordância entre os revisores, o estudo em questão foi mantido para uma análise detalhada em etapas posteriores.

Como critérios de inclusão (CI) desta revisão de literatura, foram definidos os seguintes itens:

- **CI-01:** Artigo publicado no intervalo de tempo escolhido para a revisão;
- **CI-02:** Artigos que propõem uma arquitetura, paradigma ou sistema cujo objetivo é o compartilhamento de conhecimento ou de dados no domínio da saúde;
- **CI-03:** Trabalhos escritos na língua inglesa;
- **CI-04:** Artigos que tenham o potencial para responder pelo menos a uma das questões de pesquisa da análise.

Os critérios de exclusão (CE) utilizados nas etapas de seleção de artigos desta revisão foram:

- **CE-01:** Artigos fora do período escolhido para a revisão;
- **CE-02:** Trabalhos de revisão de literatura ou *survey*;
- **CE-03:** Trabalhos cujo texto completo não está disponível e não possa ser obtido no portal de periódicos Capes;

Dessa maneira, 312 artigos foram selecionados após a primeira etapa, sendo 284 deles excluídos na segunda e terceira etapas, resultando em 28 artigos completos analisados na quarta etapa, como pode ser visto no Apêndice B. Nela, os artigos foram lidos integralmente com o objetivo de verificar se todos estão de acordo com os critérios de inclusão e exclusão desta pesquisa. Como consequência, dois artigos foram excluídos, um por não ter seu texto completo

disponível (CE-03) e outro por não propor uma arquitetura, paradigma ou sistema de compartilhamento de conhecimento na área da saúde (CI-02).

Em conclusão, ao fim das quatro etapas, 26 artigos foram incluídos neste estudo, mostrando que existe uma quantidade significativa de pesquisas que tentam resolver o problema de compartilhamento de conhecimento no domínio da saúde publicadas nos últimos anos.

3.4. EXTRAÇÃO DE DADOS

Além de responder às questões de pesquisa, esta revisão sistemática da literatura tem o objetivo de extrair as principais características das arquiteturas e sistemas propostos nos trabalhos relacionados a fim de compará-los com a proposta desta pesquisa.

As variáveis a serem extraídas foram divididas em nove categorias: metadados da publicação; paradigma; componentes da arquitetura; domínio; fontes de conhecimento; persistência; consumidores de conhecimento; segurança; validação.

Sendo assim, foram escolhidas 20 variáveis e cada uma delas foi estipulada para responder a uma ou mais questões de pesquisa. As variáveis relacionadas aos metadados da publicação e paradigma permitem uma melhor contextualização dos trabalhos analisados. Por outro lado, variáveis relacionadas a fontes de conhecimento e consumidores de conhecimento foram pensadas com o objetivo de responder a questões de pesquisa específicas como QP-02 e QP-03. A lista de variáveis, uma descrição e sua categoria podem ser vistas na Tabela 3.

Tabela 3 - Variáveis a serem extraídas dos artigos selecionados

Id	Variável	Descrição	Categoria
VR-01	Ano de publicação	O ano de publicação da pesquisa.	Metadados da publicação
VR-02	Buscador	Engenho de busca onde a publicação foi encontrada.	
VR-03	Local de publicação	Nome do periódico ou conferência em que o artigo foi publicado.	
VR-04	Tipo de publicação	Detalha se o trabalho foi publicado em periódico ou em conferência.	
VR-05	*Qualis* na computação	*Qualis* do periódico em que o artigo foi publicado de acordo com a classificação de periódicos no quadriênio 2013-2016.	
VR-06	Paradigma ou arquitetura	Nome do paradigma ou arquitetura de referência utilizada na proposta.	Paradigma
VR-07	Principais componentes	Lista com os principais componentes da arquitetura proposta.	Componentes da arquitetura

VR-08	Domínio	Domínio para o qual a arquitetura/*framework* foi projetado ou testado.	Domínio
VR-09	Reutilizável em outros domínios	Descreve se a arquitetura poderá ser adaptada para outros domínios fora da área da saúde.	
VR-10	Fontes de conhecimento	Quais são as fontes de conhecimento, sistema de representação de conhecimento ou técnicas de descoberta de conhecimento compatíveis com a arquitetura.	Fontes de conhecimento
VR-11	Múltiplas fontes de conhecimento	Descreve se a arquitetura ou *framework* possui a capacidade de extrair conhecimento de múltiplas fontes de conhecimento simultaneamente.	
VR-12	Comunicação com a fonte de conhecimento	Quais tecnologias são empregadas para a comunicação com as fontes de conhecimento.	
VR-13	Fontes de conhecimento externas	Descreve se as fontes de conhecimento devem estar sob o controle de uma mesma organização ou se múltiplas organizações podem manter distintas fontes de conhecimento.	
VR-14	Como pode ser adicionada uma nova fonte de conhecimento?	Detalha se a arquitetura prevê a adição de outras fontes de conhecimento além das propostas pelos autores.	
VR-15	Mecanismo de persistência	Quais são os mecanismos utilizados para a persistência das consultas e dados como: dados do paciente, *queries* de busca, etc.	Persistência
VR-16	Histórico de consultas e resultados	Como é tratado o aspecto temporal das consultas e seus resultados.	
VR-17	Consumidores de conhecimento	Quais são os consumidores de conhecimento, ou equivalente, propostos pelos autores.	Consumidores de conhecimento
VR-18	Comunicação com os consumidores de conhecimento	Quais tecnologias são utilizadas para comunicação com os consumidores de conhecimento.	
VR-19	Controle de acesso	Como é feito o controle de acesso ao conhecimento e quais técnicas são utilizadas para prevenir o roubo de conhecimento.	Segurança
VR-20	Forma de validação	Como é feita a validação da proposta/modelo.	Validação

Fonte: Elaborado pelo autor.

Os 26 trabalhos incluídos nesta pesquisa foram lidos de maneira que fosse possível a extração das variáveis, gerando dados suficientes para responder às questões de pesquisa desta revisão. Os artigos analisados na etapa de extração de dados e seus respectivos códigos e referências podem ser vistos no Apêndice E.

Tabelas com os dados detalhados de cada valor extraído das publicações para cada variável foram adicionadas como apêndices nesta pesquisa. No Apêndice F, estão listados os metadados das publicações como ano de publicação, local de publicação, tipo e *qualis* do evento

ou periódico. De maneira similar, o Apêndice G mostra uma tabela com os dados extraídos para as variáveis das seguintes categorias: paradigma, componentes da arquitetura e domínio. Detalhadas no Apêndice H estão as variáveis relacionadas às fontes de conhecimento. Para concluir, estão listados no Apêndice I os dados das variáveis das categorias persistência, consumidores de conhecimento, segurança e validação.

3.5. RESULTADOS

Nesta revisão sistemática da literatura foram incluídos 26 trabalhos, os quais passaram por uma etapa de extração de dados que obteve os valores para as 20 variáveis definidas na etapa de extração de dados. Os dados extraídos foram utilizados para responder às questões de pesquisa, objetivo desta revisão.

A Tabela 4 mostra a lista de variáveis utilizadas diretamente na resposta de cada questão de pesquisa. As variáveis VR-01, VR-02, VR-03, VR-04 e VR-05, que fazem parte da categoria de metadados da publicação, foram empregadas em diferentes etapas desta pesquisa com o objetivo de contextualizar cada trabalho incluído.

Tabela 4 - Variáveis utilizadas para responder a cada questão de pesquisa.

Questão de pesquisa	Variáveis utilizadas
QP-01	VR-06; VR-07
QP-02	VR-10; VR-12; VR-17; VR-18
QP-03	VR-10; VR-11; VR-12; VR-13; VR-14
QP-04	VR-08; VR-09
QP-05	VR-15; VR-16; VR-19
QP-06	VR-20

Fonte: Elaborado pelo autor.

Os trabalhados relacionados que foram identificados também são utilizados como inspiração para a definição dos componentes da arquitetura proposta nesta pesquisa, a H-KaaS. Dessa forma, nesta seção serão apresentadas as respostas das questões de pesquisa propostas nesta revisão sistemática.

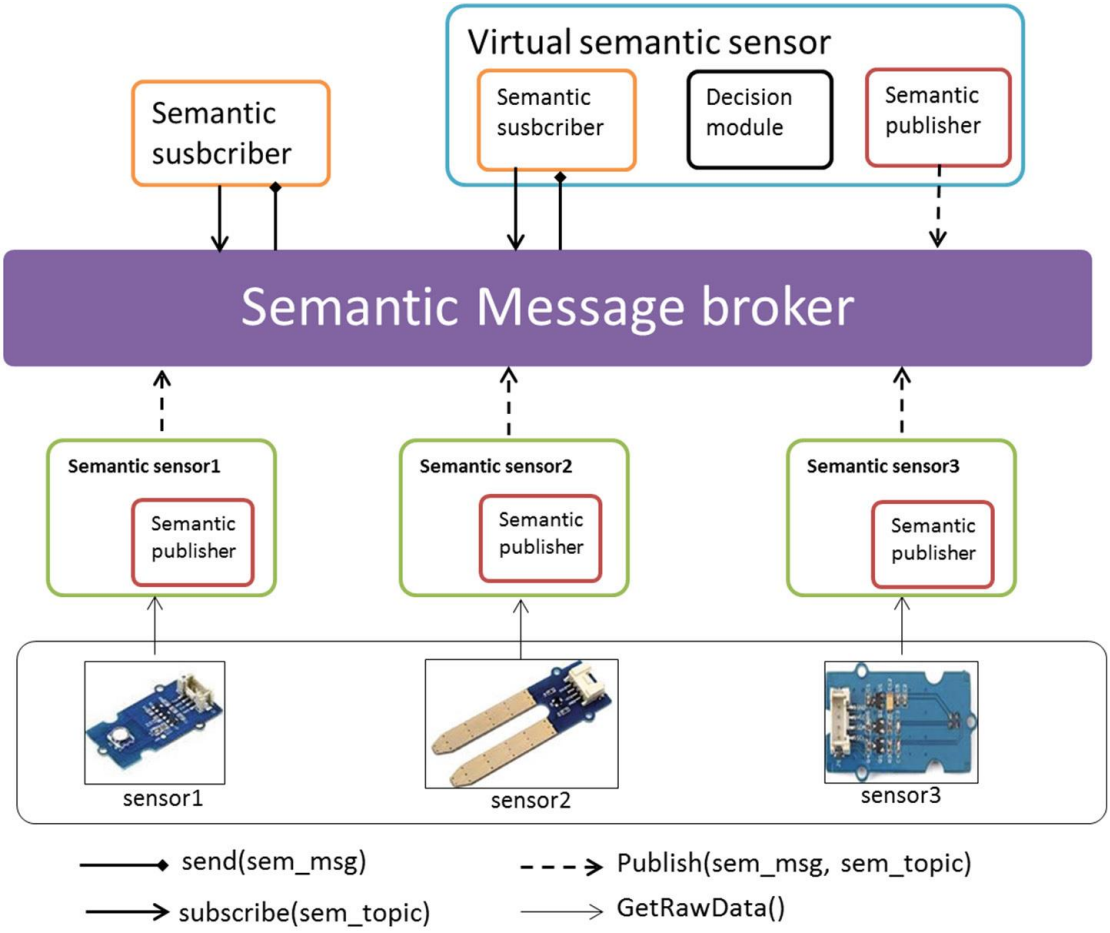
QP-01: Quais são os principais componentes propostos pelas arquiteturas para compartilhamento de conhecimento na área da saúde?

Através da análise das variáveis “paradigma ou arquitetura” e “principais componentes”, é possível ver que não existe uma padronização nos componentes das arquiteturas, mesmo para propostas que utilizam o mesmo paradigma. Os dados extraídos das descrições de cada arquitetura foram organizados e disponibilizados no Apêndice G.

Para exemplificar os componentes encontrados e sua variabilidade, pode-se destacar dois trabalhos em subáreas distintas, o artigo publicado por Zghei et al. (2017) e o de Fattah e Chong (2018).

No artigo de Zghei et al. (2017), intitulado “Engineering IoT healthcare applications: Towards a semantic data driven sustainable architecture”, é detalhada uma arquitetura baseada em serviços de mensagens que se utiliza de um componente centralizador chamado pelos autores de *semantic message broker*, capaz de intermediar a comunicação entre os *semantic publishers* e os *semantic subscribers* (Figura 12). Essa centralização do acesso permite a padronização das trocas de dados, que, neste caso, é feita por mensagens no formato OWL, utilizadas como principal forma de comunicação entre os componentes internos da arquitetura. Além disso, também pode ser considerado como vantagem da centralização um melhor controle do acesso aos dados e dos modelos de conhecimento disponibilizados. Por outro lado, como desvantagem da centralização do acesso ao conhecimento em certas arquiteturas, a adequação a alguns requisitos não funcionais pode ser mais custosa, como escalabilidade, alta disponibilidade, etc.

Figura 12 - Exemplo de uma arquitetura para o domínio da IoT aplicado à saúde

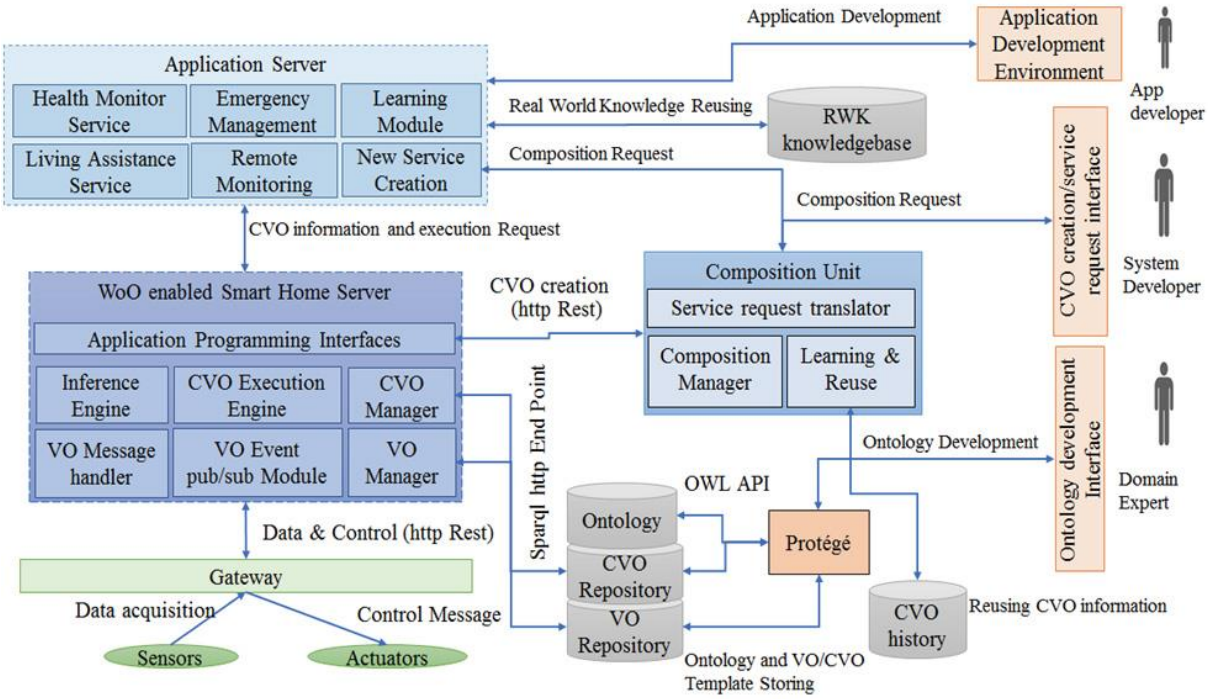


Fonte: Zghei et al. (2017).

No entanto, outra arquitetura baseada em serviços, também no domínio da saúde e internet das coisas, é proposta por Fattah e Chong (2018) no trabalho “Restful Web services composition using semantic ontology for elderly living assistance services”, que utiliza objetos virtuais e objetos virtuais compostos disponíveis através de um componente servidor para casas inteligentes, permitindo a comunicação entre dispositivos com a ajuda de um servidor de aplicação.

A Figura 13 apresenta uma visão geral da arquitetura proposta por Fattah e Chong (2018), baseada no conceito de *web of objects* (WoO), que possibilita a integração de vários objetos do mundo real em uma aplicação web centralizada, permitindo a criação de serviços inteligentes. Esta arquitetura é composta por um *gateway*, chamado pelos autores de *application server*, que funciona como um *middleware* entre o serviço para casas inteligentes baseado em *web of objects* e os objetos físicos.

Figura 13 - Arquitetura proposta por Fattah e Chong (2018) baseada no conceito de WoO



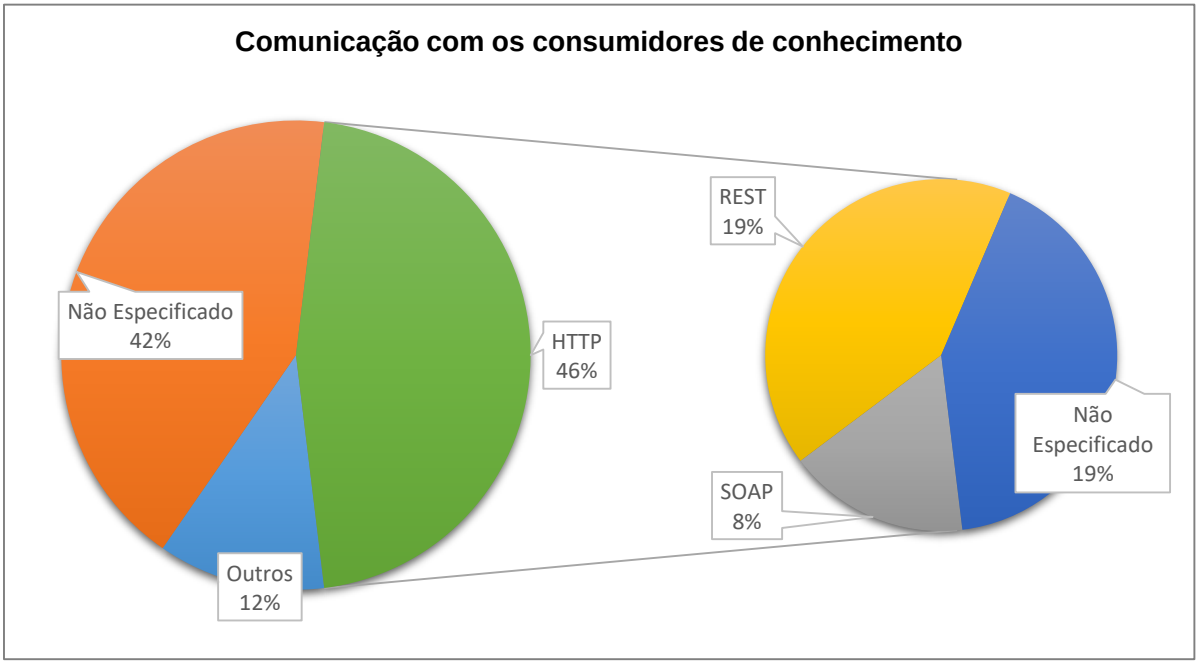
Fonte: Fattah e Chong (2018).

QP-02: Como a comunicação entre os componentes da arquitetura ou paradigma proposto é feita?

A partir da análise da variável “comunicação com a fonte de conhecimento”, pode-se concluir que a comunicação interna entre os componentes da arquitetura é variada e geralmente depende das fontes de conhecimento escolhidas, porém, como a maioria das propostas utiliza-se de algum tipo de raciocínio semântico ou ontologias, os principais métodos de acesso às fontes de conhecimento são a utilização de APIs, de biblioteca de manipulação ou de consultas a grafos RDF, como a Jena API, manipulações OWL e consultas SPARQL.

Por outro lado, em relação à variável “comunicação com os consumidores de conhecimento”, dos 15 trabalhos que especificam sua forma de comunicação com os aplicativos consumidores, 12 utilizam o HTTP como principal protocolo para a comunicação com os aplicativos consumidores de dados e/ou conhecimento. Destes, cinco artigos informam a utilização do padrão REST e dois o padrão SOAP. A Figura 14 mostra os dados coletados para a variável “comunicação com os consumidores de conhecimento”. Sendo assim, o emprego de uma API de comunicação REST em conjunto com o protocolo HTTP parece ser o método mais usado para a comunicação com os aplicativos consumidores.

Figura 14 - Estatísticas coletadas através da variável “comunicação com os consumidores de conhecimento”



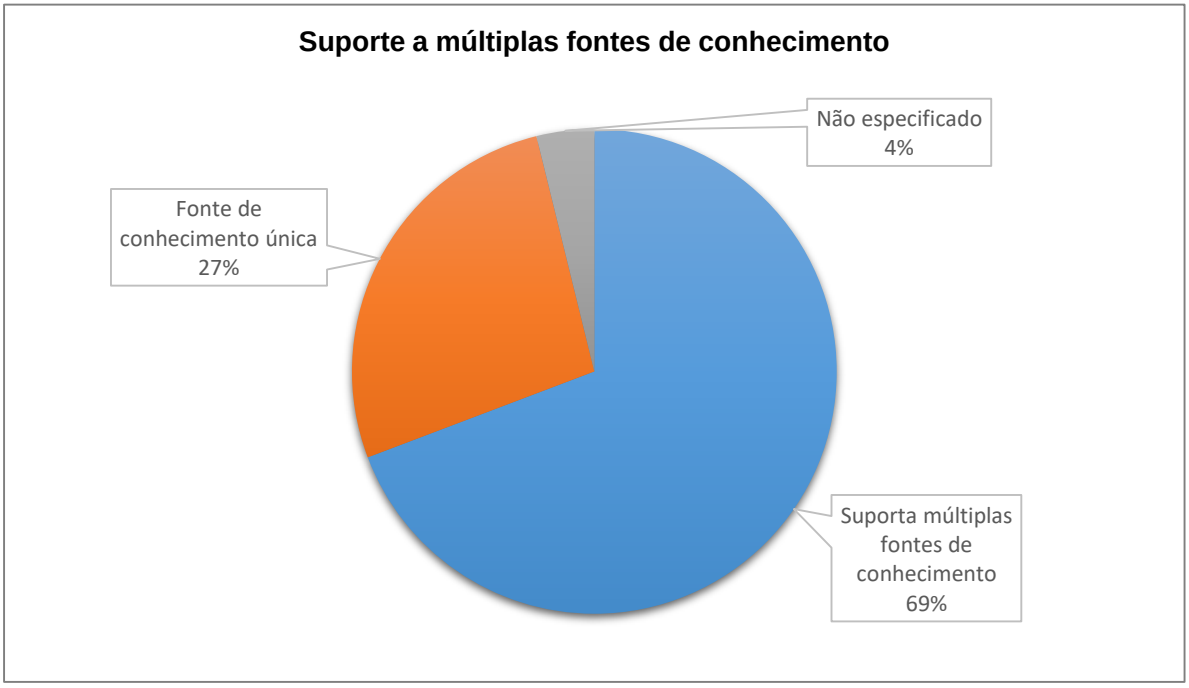
Fonte: Elaborado pelo autor.

QP-03: Quais formas de representação e descoberta de conhecimento são utilizadas em arquiteturas focadas no compartilhamento de conhecimento no domínio da saúde?

Com base no estudo dos dados extraídos para a variável “fontes de conhecimento”, percebe-se que ontologias e grafos RDF são as fontes de conhecimento mais comuns em serviços de compartilhamento de dados e conhecimento no domínio da saúde. Cerca de 65% dos serviços têm suporte a ontologias como principal método de representação de conhecimento.

Além disso, como pode ser visto na Figura 15, 18 publicações possuem a capacidade de lidar com múltiplas fontes de conhecimento distintas, 7 são limitadas a apenas uma fonte de conhecimento e uma única publicação não esclarece esta questão.

Figura 15 - Estatística sobre a quantidade de trabalhos que são capazes de operar com múltiplas fontes de conhecimento



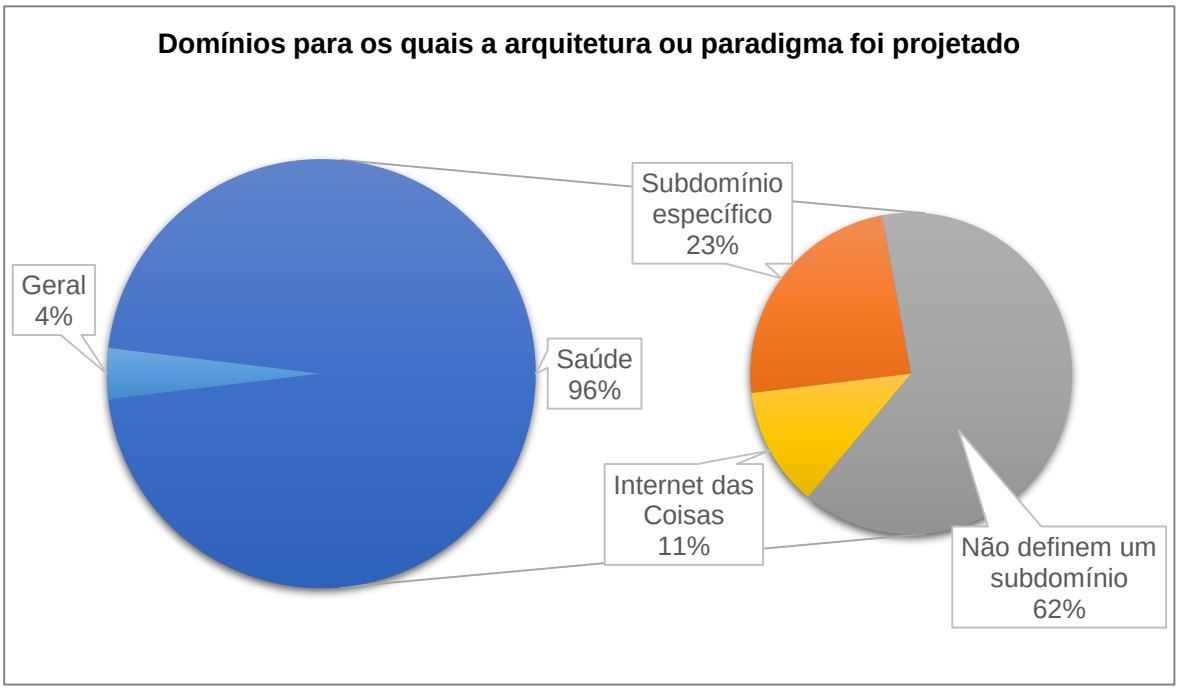
Fonte: Elaborado pelo autor.

Em relação às fontes de conhecimento, em 14 dos 26 trabalhos é possível notar que sua localização pode ser externa ao serviço, aumentado a flexibilidade da proposta e, geralmente, possibilitando a distribuição dos componentes do sistema entre diferentes organizações.

QP-04: Em quais subdomínios da saúde as arquiteturas de compartilhamento de conhecimento estão sendo utilizadas?

Dentre os trabalhos analisados, a partir dos dados da variável “domínio”, é possível concluir que 25 dos trabalhos propõem arquiteturas ou sistemas para o domínio da saúde e apenas um se identifica como sendo de domínio geral. Nesse contexto, considerando apenas os trabalhos voltados para o domínio da saúde, 16 não definem subdomínios específicos, 3 tentam resolver problemas do domínio da internet das coisas e os outros 6 tratam das áreas de: medicina personalizada, medicina tradicional chinesa, sistemas de decisão clínica, sistemas para o diagnóstico de depressão e robôs para auxílio à reabilitação (Figura 16).

Figura 16 - Domínios para os quais a arquitetura ou paradigma foi projetado para ser utilizado de acordo com seus respectivos autores



Fonte: Elaborado pelo autor.

Como pode ser visto na Figura 17, outra conclusão que pode ser obtida a partir da análise das variáveis da categoria “domínio” é que a maior parte dos trabalhos apresenta soluções voltadas para a área da saúde, sem levantar discussões sobre a possibilidade da sua reutilização em outros domínios: 17 dos trabalhos apresentam propostas específicas para o domínio para os quais foram construídos, 5 mostram que podem ser usados em outros domínios e 4 não especificam essa característica.

Figura 17 - Estatísticas sobre a possibilidade de reutilização da arquitetura ou paradigma em outros domínios

Fonte: Elaborado pelo autor.

QP-05: Como o problema da persistência dos dados é tratado pelas arquiteturas?

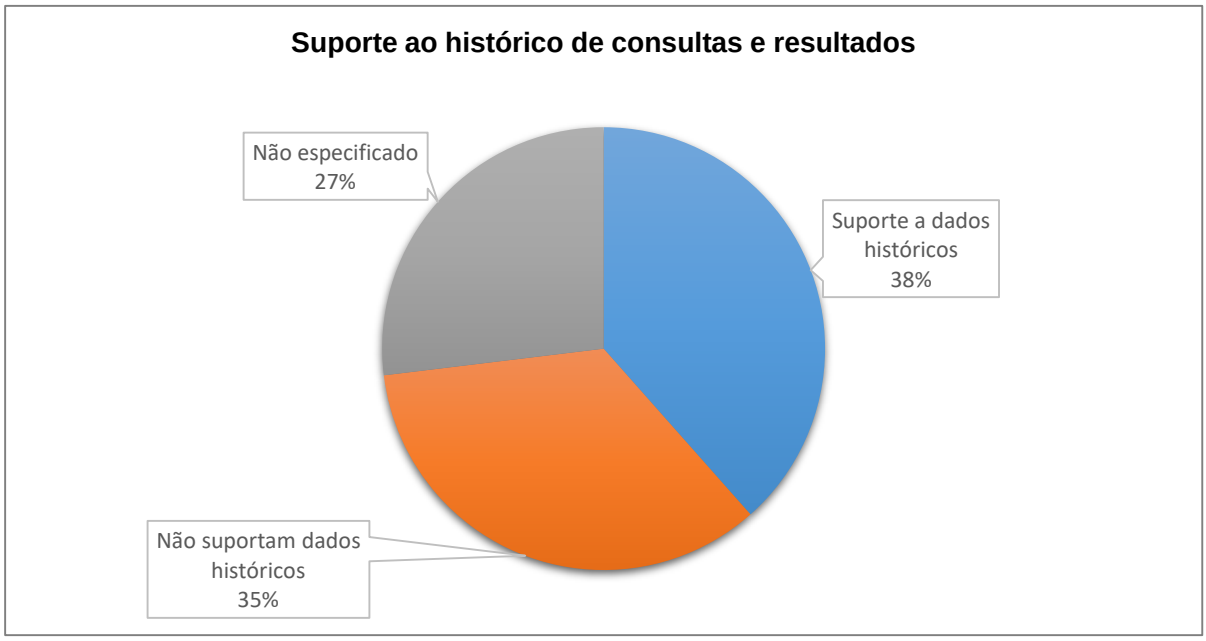
A partir da análise dos dados coletados, de acordo com a variável "mecanismo de persistência", é possível notar que, dentre os 19 trabalhos que especificam os métodos utilizados para a persistência de dados, existe uma grande variação em relação às tecnologias empregadas: 14 propostas valem-se de alguma variação do formato RDF, como ontologias OWL e suas instâncias, como mecanismo de persistência para o conhecimento. Além disso, seis dos trabalhos especificam múltiplos formatos para o armazenamento dos dados, sendo estes mais flexíveis no que diz respeito ao suporte a mecanismos de persistência. Neste contexto, é importante notar que o problema da persistência de dados é bastante relevante na área da saúde devido a vários fatores específicos do domínio como, por exemplo, mudanças periódicas nas *guidelines* para tratamento de doenças e a necessidade do armazenamento de dados dos pacientes e do histórico de consultas.

De maneira similar, destaca-se o trabalho de Peral et al. (2018) com título "An ontology-oriented architecture for dealing with heterogeneous data applied to telemedicine systems", que propõe uma arquitetura capaz de processar grandes quantidades de dados a partir de fontes de conhecimento heterogêneas como redes de sensores, redes sociais e dados textuais não estruturados. A proposta usa uma arquitetura baseada em ontologias que possibilitem a

integração das fontes de dados, por servirem como sistemas intermediários de integração. Cada fonte de dados possui sua própria ontologia de domínio, que é utilizada para enriquecer ou associar os dados das fontes com a ontologia universal central.

Em relação à persistência e acesso ao histórico de consultas e seus resultados, 10 dos trabalhos analisados possuem suporte a esta funcionalidade, 9 não possuem e 7 publicações não especificam essa característica. A Figura 18 mostra a possibilidade de armazenamento de dados históricos relacionados às consultas e seus resultados. Desta forma, a persistência das consultas e resultados parece depender apenas do domínio e das escolhas dos projetistas, visto que, em certos subdomínios da saúde, os dados e resultados das consultas não precisam ser persistidos. Por outro lado, em outros subdomínios e aplicações na saúde, os dados históricos são fundamentais para a execução das consultas.

Figura 18 - Dados coletados para a variável "histórico de consultas e resultados"



Fonte: Elaborado pelo autor.

Por outro lado, na proposta de Dogmus, Erdem e Patoglu (2015), em seu artigo intitulado "RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud", pode ser encontrada uma solução, baseada em serviços e ontologia, para a integração de fontes de dados heterogêneas no domínio da saúde, como bancos de dados de pacientes ou ontologias de domínio. O sistema utiliza-se de várias ontologias de domínio e raciocinadores para oferecer respostas às consultas executadas. Além disso, foi definida a possibilidade do uso de um banco de dados relacional capaz de armazenar

informações sobre os usuários e permissões. Os autores também especificam um módulo de manutenção e backup cujo objetivo é armazenar versões anteriores das asserções criadas pelos usuários do sistema. Sendo assim, o trabalho se destaca pelos detalhes providos em relação à persistência de dados que, através do uso de diferentes paradigmas e tecnologias de persistência, torna-se capaz de armazenar informações históricas do conhecimento modelado.

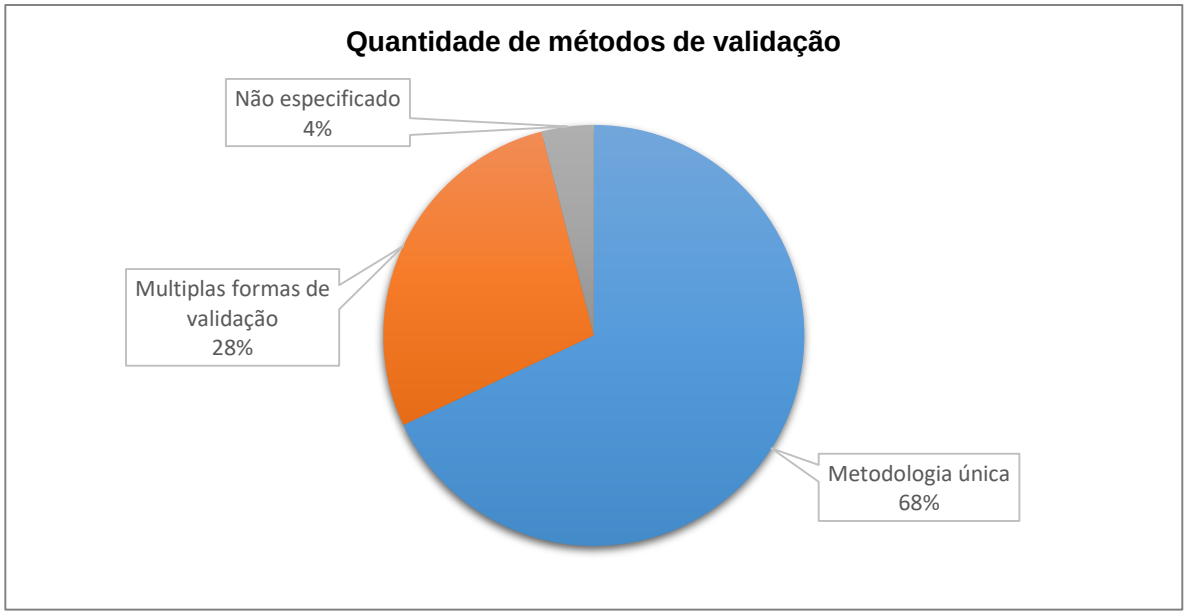
QP-06: Como propostas de novas arquiteturas com foco no compartilhamento de conhecimento na área da saúde são validadas?

É possível notar, através da análise da variável "forma de validação", que a maioria dos estudos implementa algum tipo de protótipo ou descreve aplicações implementadas usando a abordagem proposta na pesquisa. Protótipos são apresentados por 21 estudos, sendo 3 deles validados por especialistas. Seis pesquisas possuem validação utilizando medidas de performance como teste de latência e acurácia aplicado aos protótipos descritos. É importante notar que 12 trabalhos empregam apenas o desenvolvimento e descrição de protótipos como forma de validação, enquanto 2 trabalhos não descrevem como a sua proposta foi validada.

Por outro lado, sete estudos valem-se de múltiplas formas de validação (Figura 19), e alguns deles testam diretamente suas fontes de conhecimento avaliando métricas como precisão, *recall* e *F-measure* dos resultados das consultas, que são técnicas da área de recuperação da informação (do inglês *information retrieval*), uma subárea da ciência da computação que lida com problemas de otimização e avaliação da busca pela informação (FRAKES; BAEZA-YATES, 1992).

Nesse contexto, destaca-se o trabalho de Arch-Int et al. (2017), que tem o objetivo de propor um serviço capaz de descobrir, selecionar, compor e monitorar serviços baseados em web semântica de maneira automática. Durante o desenvolvimento da pesquisa, com o objetivo de validar o serviço proposto, além de implementar um protótipo como a maioria dos trabalhos, os autores fazem uma análise de corretude, por meio da métrica de precisão, *recall* e *F-measure*, e também uma análise do tempo de resposta do serviço web presente no protótipo.

Figura 19 - Número de formas de validação distintas usadas pelos autores dos trabalhos



Fonte: Elaborado pelo autor.

3.6. CONCLUSÕES

Esta revisão sistemática teve o propósito de analisar os trabalhos relacionados a esta pesquisa, cujo objetivo é propor uma arquitetura de referência para o compartilhamento do conhecimento no domínio da saúde. Foram incluídos na revisão 26 artigos que, por sua vez, passaram por uma etapa de extração de dados que avaliou o total de 20 variáveis distintas a fim de responder às cinco questões de pesquisa, definidas durante o processo de planejamento da revisão sistemática.

Como formas de representação de conhecimento, ontologias e grafos RDF mostraram-se a solução mais utilizada entre os trabalhos identificados. Além disso, a maioria das propostas é capaz de lidar com múltiplas fontes de conhecimento. Ainda nesse contexto, cerca de metade das publicações apresentam soluções compatíveis com fontes de conhecimentos externas ao serviço principal.

Os componentes das arquiteturas voltadas para o compartilhamento de conhecimento na área da saúde são variados, sendo projetados de acordo com o problema que se deseja resolver. Outra conclusão da análise dos componentes das arquiteturas foi que a maioria das propostas visa, de alguma forma, a centralizar o acesso aos dados, permitindo a padronização e o controle de acesso ao conhecimento.

A comunicação com as fontes de conhecimento normalmente é feita de acordo com as fontes de conhecimento suportadas e, como a maioria tem a capacidade de lidar com ontologias e grafos RDF, consultas OWL e SPARQL destacam-se como as principais formas de acesso ao conhecimento.

Por outro lado, em relação à comunicação com os consumidores de conhecimento, as soluções propostas, em sua grande maioria, utilizam-se do protocolo HTTP e, em geral, em combinação com o modelo REST ou SOAP.

Em relação às áreas de utilização, a maioria dos trabalhos apresenta propostas de sistemas para o domínio da saúde como um todo, e apenas alguns deles foram projetados para resolver problemas específicos de subdomínios dessa área.

No contexto da persistência do conhecimento e consultas, pode-se concluir que existem várias soluções para a persistência do conhecimento extraído das fontes de conhecimento, sendo o formato RDF e suas variações, como ontologias OWL, o mecanismo mais utilizado. Por outro lado, percebe-se que a persistência das consultas e seus resultados variam de acordo com o problema a ser resolvido, sendo, em alguns casos, até mesmo dispensável.

Para a validação, pode-se concluir que a implementação de protótipos é a forma mais comum de validação de arquiteturas cujo objetivo é o compartilhamento do conhecimento no domínio da saúde. Adicionalmente, podem ser feitas validações complementares como a execução de diferentes estudos de caso, validação dos resultados por especialistas, validação direta do retorno das fontes de conhecimento e *benchmarks* focados na qualidade de serviço, como o teste de latência, e avaliação das fontes de conhecimento utilizando métricas da área de recuperação da informação.

Em conclusão, diante dos resultados obtidos pela revisão sistemática da literatura, nota-se uma falta de padronização e de meios para comparar os componentes e estruturas de comunicação propostos pelos autores dos trabalhos analisados. Sendo assim, fica evidente a necessidade da especificação de uma arquitetura de referência capaz de generalizar sistemas de compartilhamento de conhecimento no domínio da saúde. Essa arquitetura de referência deverá refletir os conceitos fundamentais do domínio, identificando e descrevendo como cada subsistema poderá ser instanciado e executado, além de fornecer uma estrutura apropriada e uma base para comparação entre sistemas existentes. Desta forma, no próximo capítulo, será proposta a H-KaaS, uma arquitetura de referência para o domínio da saúde, que visa a facilitar o desenvolvimento de novos sistemas de compartilhamento de conhecimento e dados, além de servir como um modelo generalista que poderá ser empregado na análise de sistemas existentes.

4. H-KAAS: UMA ARQUITETURA DE REFERÊNCIA BASEADA EM CONHECIMENTO COMO SERVIÇO PARA E-SAÚDE

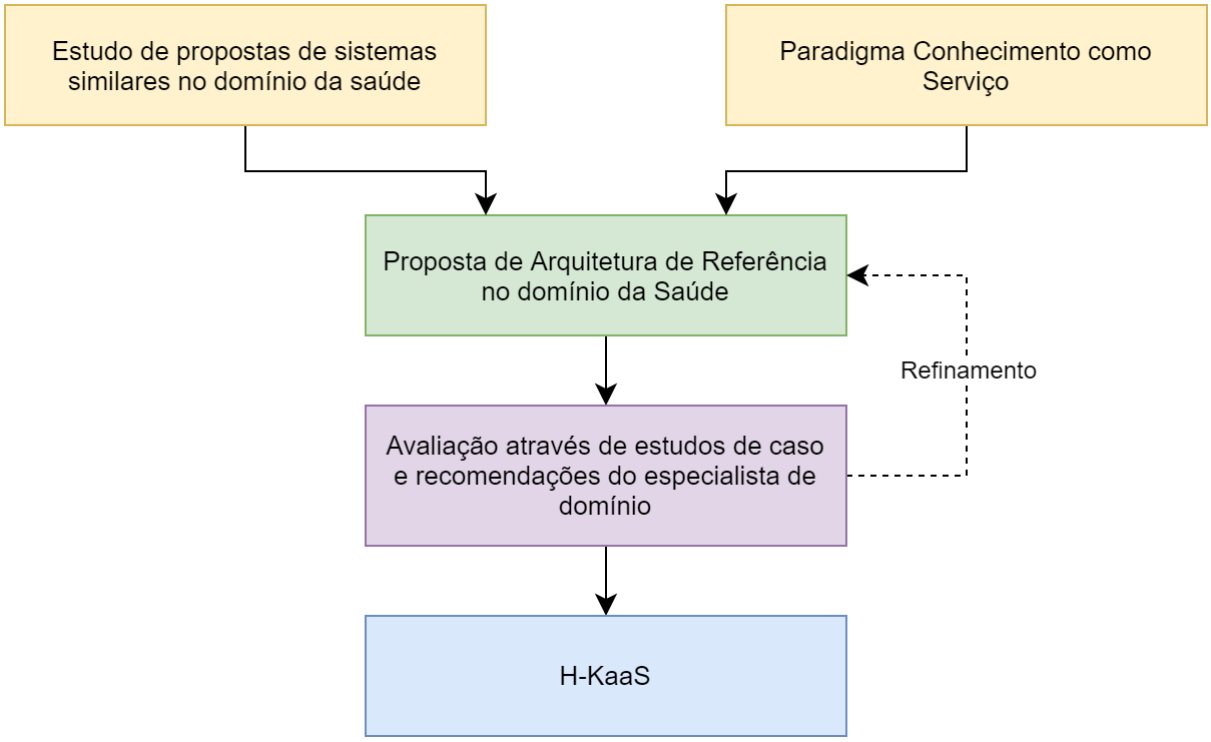
No domínio da saúde, embora estejam sendo propostos muitos sistemas e arquiteturas de software com o objetivo de compartilhar dados e conhecimento, ainda existe uma grande dificuldade referente à falta de padronização de seus componentes e interfaces de comunicação, prejudicando a interoperabilidade e análise comparativa desses sistemas. Além disso, existe uma dificuldade ainda maior em encontrar arquiteturas de referência baseadas no paradigma KaaS, que poderiam trazer inúmeras vantagens para a área.

Como visto nos capítulos anteriores, uma arquitetura de referência (BASS; CLEMENTS; KAZMAN, 2003) visa ser usada como um modelo generalizado de diversos sistemas reais de certo domínio, a fim de facilitar o desenvolvimento de novas aplicações similares e de servir como *framework*-base para comparação entre sistemas existentes (SOMMERVILLE, 2007). Sendo assim, diante dos fatos mencionados, como principal objetivo deste trabalho, será proposta a H-KaaS, uma arquitetura de referências baseada no paradigma de conhecimento como serviço para a área da saúde.

Para tal, a metodologia utilizada no desenvolvimento da arquitetura de referência proposta, apresentada na Figura 20, considerou as definições apontadas por Gallagher (2000) sobre o desenvolvimento e objetivos de uma arquitetura de referência, e teve como principais inspirações o estudo de sistemas similares no mesmo domínio com o objetivo de compartilhar conhecimento, em conjunto com os princípios apontados por Xu e Zhang (2005) através das definições do paradigma de conhecimento como serviço.

Portanto, como primeiro passo para o desenvolvimento da arquitetura, realizou-se o estudo de propostas de sistemas de compartilhamento de conhecimento no domínio da saúde por meio de uma revisão sistemática da literatura, apresentada no capítulo 3, que forneceu respostas a questões de pesquisa fundamentais para o desenvolvimento da proposta apresentada. Além disso, a análise sistemática dos trabalhos produziu dados sobre os componentes e tecnologias utilizados por cada proposta, dados estes que estão disponíveis nos apêndices F, G, H e I desta pesquisa. Os dados que tratavam sobre arquiteturas distribuídas, embora incompatíveis com o paradigma KaaS, foram utilizados parcialmente para definição de alguns conceitos e componentes da H-KaaS.

Figura 20 - Metodologia utilizada para o desenvolvimento da arquitetura H-KaaS

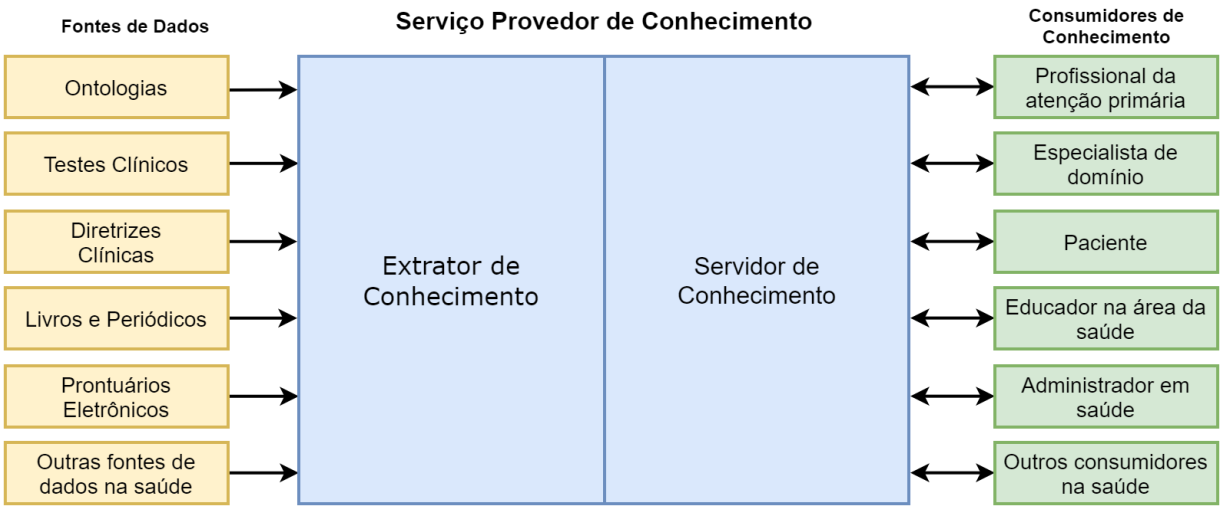


Fonte: Elaborado pelo autor.

Em seguida, ao estudar o modelo conceitual proposto por Xu e Zhang (2005) para o paradigma de conhecimento como serviço, devido à sua flexibilidade e seu foco na centralização do acesso ao conhecimento, ficou evidente a possibilidade de sua adaptação para o domínio da saúde. Com isso em mente, nas fases iniciais do desenvolvimento da arquitetura, foi conceitualizada uma instância do paradigma de conhecimento como serviço que pudesse ser aplicada nesse domínio, apresentada na Figura 21.

Essa versão simplificada da arquitetura inicial, de maneira similar ao paradigma KaaS, possui componentes referentes à: fontes de dados, consumidores de conhecimento, extrator de conhecimento e, por fim, servidor de conhecimento.

Figura 21 - Arquitetura conceitual inicial, adaptada a partir paradigma KaaS para a saúde



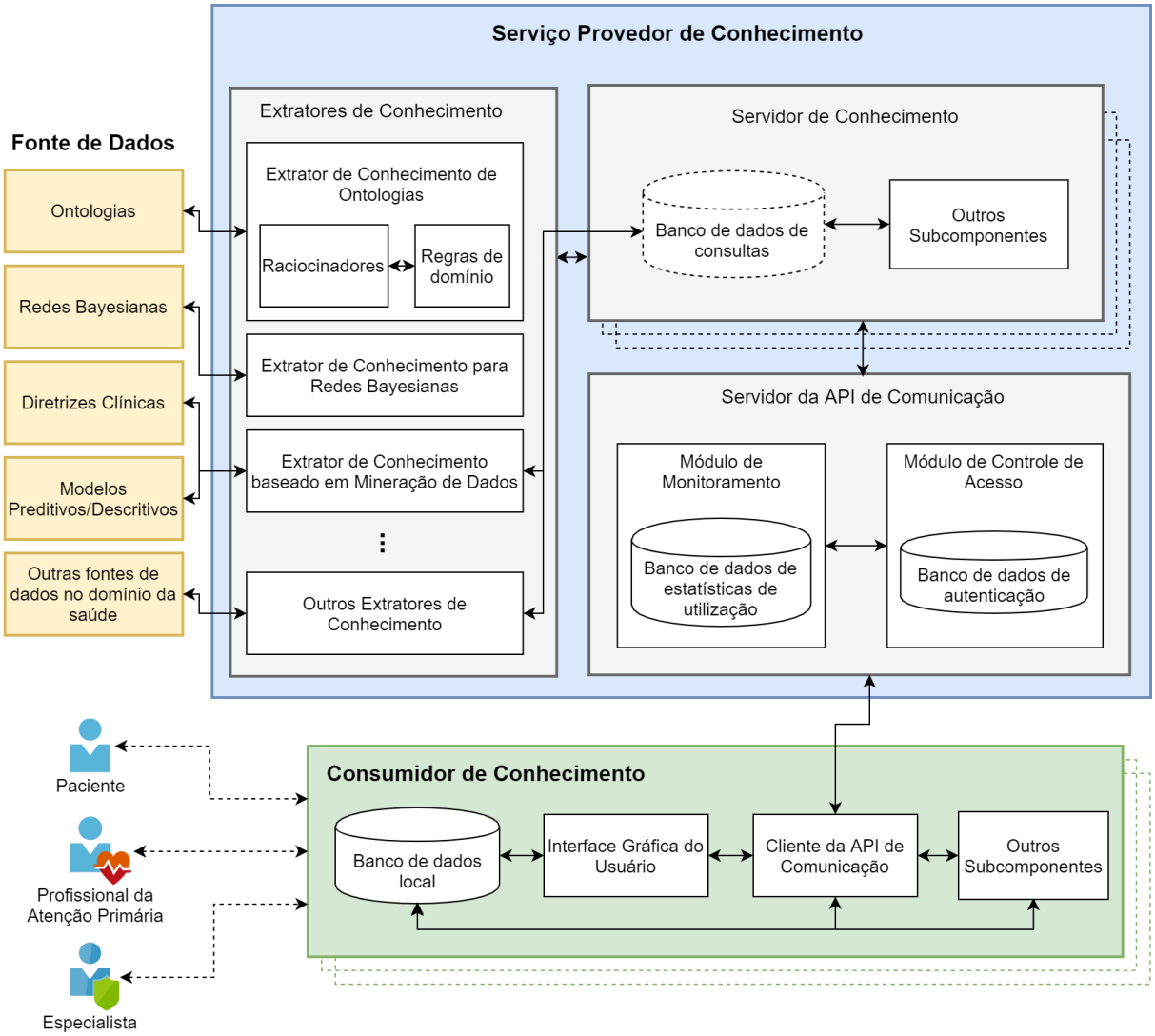
Fonte: Adaptado para a área da saúde com base em Xu e Zhang (2005).

Dessa forma, no domínio da saúde, podem ser consideradas fontes de dados, por exemplo, resultados de testes clínicos, ontologias de domínio, livros, periódicos, redes bayesianas para o suporte à decisão clínica, orientações para o tratamento de doenças, entre outros. Em relação aos aplicativos consumidores de conhecimento, 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, visando a uma melhora do serviço prestado por especialistas e profissionais da atenção primária, dentre outros. Além disso, é possível o desenvolvimento de aplicações e fontes de dados direcionados à educação de novos profissionais ou de pacientes interessados em estudar sobre sua condição médica (a fim de ajudar no tratamento, por exemplo) e o domínio da saúde em geral.

Posteriormente, a partir do estudo realizado sobre o paradigma de conhecimento como serviço e por meio da análise dos componentes dos sistemas identificados na revisão sistemática, foi possível o desenvolvimento da arquitetura de referência no domínio da saúde H-KaaS, que visa facilitar a elaboração de novas aplicações centralizadas de compartilhamento de conhecimento e oferecer meios para comparação e análise de sistemas similares existentes no domínio. Assim, os componentes descritos pela arquitetura conceitual inicial foram detalhados por meio da especificação de seus subcomponentes, formas de utilização e interfaces de comunicação, resultando na especificação da arquitetura de referência H-KaaS, apresentada nesta pesquisa.

A arquitetura H-KaaS, incluindo os seus subcomponentes, é apresentada na Figura 22. Nesta, as setas representam a direção em que os dados e, consequentemente, o conhecimento, pode fluir entre os diversos componentes da arquitetura.

Figura 22 - Arquitetura H-KaaS, uma arquitetura de referência baseada em conhecimento como serviço para o domínio da saúde



Fonte: Elaborado pelo autor.

As fontes de dados, em amarelo, são responsáveis por fornecer dados e/ou conhecimento para o serviço provedor de conhecimento que, através de extratores de dados e de servidores de conhecimento, fornece respostas às consultas feitas por um ou mais aplicativos consumidores de conhecimento. Além dos componentes herdados do paradigma de conhecimento como serviço, neste trabalho foram detalhados seus subcomponentes e interfaces de comunicação,

comuns a sistemas reais existentes no domínio, de acordo com os dados coletados pela revisão sistemática de literatura incluída nesta pesquisa.

Em conclusão, a arquitetura H-KaaS foi projetada visando a facilitar o compartilhamento de conhecimento adquirido de múltiplas fontes de dados para um ou mais aplicativos consumidores, tornando este compartilhamento mais organizado e eficiente. Além disso, a H-KaaS foi dividida em subcomponentes bem definidos a fim de mais facilmente ser mantida, entendida e instanciada.

Nas próximas seções, será discutido em detalhes cada um desses componentes, fornecendo informações de como eles podem ser utilizados e suas interfaces de comunicação.

4.1. FONTES DE DADOS

No contexto da arquitetura H-KaaS, entende-se como fonte de dados os componentes responsáveis por fornecer dados, informação ou conhecimento para o serviço provedor de conhecimento, componente central da arquitetura que, por sua vez, através de extratores de conhecimento, responderá a consultas feitas por aplicativos consumidores de conhecimento.

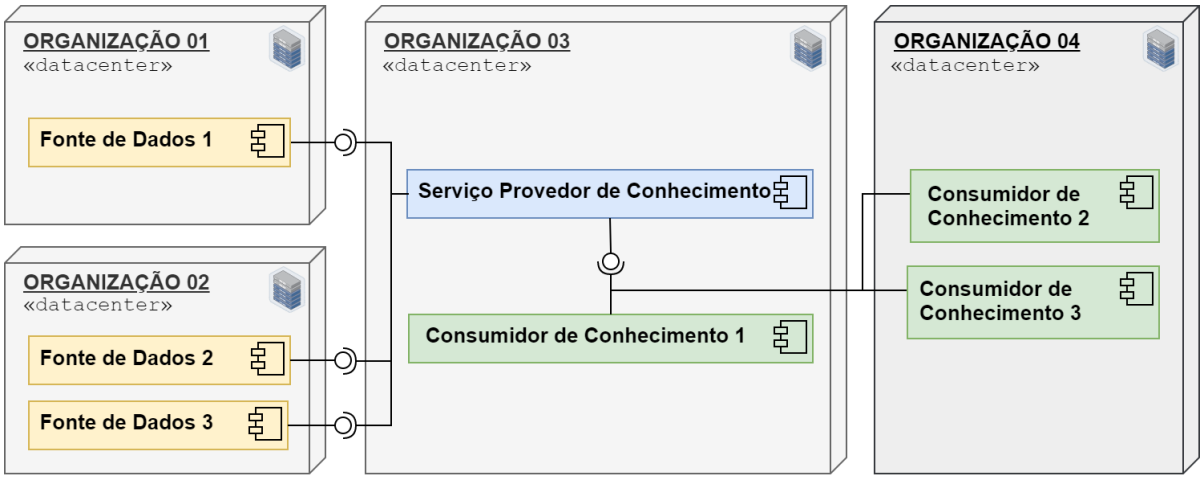
Uma mesma instância da arquitetura H-KaaS pode possuir uma ou mais fontes de dados de diferentes tipos, possivelmente com interfaces de comunicação distintas. Conforme detalhado por Xu e Zhang (2005), ao definirem os componentes do paradigma KaaS, embora os extratores de conhecimento possam manipular e filtrar os dados de cada fonte de dados, não é sua responsabilidade a anonimização desses dados. Dessa forma, as fontes de dados podem possuir algoritmos e filtros internos responsáveis por filtrar e manipular os dados entregues, de maneira a garantir a segurança das informações trocadas.

Na arquitetura de referência H-KaaS, como mencionado anteriormente, fontes de dados podem ser, por exemplo, o resultado de testes clínicos, ontologias de domínio, livros, periódicos, redes bayesianas para o suporte à decisão clínica, orientações para o tratamento de doenças, sistemas de suporte à decisão clínica ou outros sistemas produtores de conhecimento no domínio da saúde. Além desses exemplos, fontes de dados podem ser baseadas em modelos de conhecimento criados por especialistas ou por algoritmos de aprendizagem de máquina que, por sua vez, podem usar o *feedback* provido pelo sistema provedor de conhecimento para aprimorar seus modelos internos de conhecimento. Esse fluxo de dados não é previsto originalmente pelo paradigma KaaS porém, a definir a arquitetura H-KaaS, esse *feedback* se

mostrou necessário e foi representado na arquitetura pelas setas direcionadas dos extratores de conhecimento para as fontes de dados, apresentadas anteriormente na Figura 22.

A arquitetura H-KaaS, de maneira similar ao especificado no paradigma KaaS, prevê a possibilidade do uso de múltiplas fontes de dados independentes, permitindo uma maior flexibilidade na obtenção das informações e/ou conhecimento provido por serviços externos. Essa flexibilidade permite que as fontes de dados possam ser tratadas como sistemas isolados, que podem ou não ser mantidos pela mesma organização responsável pela implantação da H-KaaS. A Figura 23 apresenta um diagrama de implantação de uma arquitetura H-KaaS no formato UML (do inglês *Unified Modeling Language*) (JACOBSON; BOOCH; RUMBAUGH, 1999), uma linguagem de modelagem bastante utilizada para definições de arquitetura e processos de software. Nela, é demonstrado como fontes de dados e consumidores de conhecimentos podem ser mantidos por diferentes organizações em *datacenters* distintos. Essa possibilidade mostra a importância da definição e padronização da interface de comunicação, visto que esta é compartilhada entre esses sistemas de software que, conforme discutido, podem ser mantidos por times de desenvolvimento diferentes.

Figura 23 - Diagrama de implantação que mostra a possibilidade de as fontes de dados e consumidores de conhecimento serem mantidos por organizações diferentes



Fonte: Elaborado pelo autor.

Nesse contexto, devido ao fato de as fontes serem, em geral, modulares e independentes, é importante notar que estas podem ser, inclusive, outros sistemas provedores de conhecimento baseados ou não na arquitetura H-KaaS. Dessa maneira, é possível projetar sistemas compostos, encadeando múltiplas instâncias da H-KaaS, permitindo que uma instância faça o papel de fonte de dados e outra faça o papel de consumidor de conhecimento. Essa possibilidade de

composição de sistemas segue o princípio de modularidade de sistemas SOA, aumentando a flexibilidade das aplicações baseadas nessas arquiteturas.

Como o acesso a algumas fontes de dados pode ser limitado ou, de alguma forma, custoso, o arquiteto pode optar por adicionar, dentro da interface de comunicação, meios para o agendamento de requisições ou mecanismos de cache, uma forma de armazenamento temporário que pode ajudar na redução do tempo de resposta em caso de consultas similares ou repetidas.

Nas seções seguintes serão descritas as principais classificações das fontes de dados quanto à sua forma de processamento, comunicação e outras características. Além disso, serão detalhadas algumas particularidades em relação às possíveis formas de comunicação com as fontes de dados.

4.1.1. CLASSIFICAÇÃO DAS FONTES DE DADOS

As fontes de dados podem ser classificadas sob diferentes perspectivas, fornecendo informações importantes ao engenheiro de software responsável por implementar uma arquitetura baseada na H-KaaS.

Essas classificações podem ser utilizadas tanto para melhor entender sistemas de software já existentes quanto para auxiliar na tomada de decisão durante a criação de novas instâncias da H-KaaS. Além disso, com a classificação das fontes de dados de um sistema provedor de conhecimento, pode-se ter uma visão geral das possíveis tecnologias a serem utilizadas em diversos pontos da arquitetura, como, por exemplo, na interface de comunicação com as fontes de dados e nos extratores de conhecimento.

As subseções seguintes descrevem como as fontes de dados podem ser classificadas quanto à sua localização, modelo de dados ou conhecimento, forma de processamento e metodologia para entrega de dados.

4.1.1.1. LOCALIZAÇÃO

Do ponto de vista da organização, as fontes de dados podem ser gerenciadas de maneiras diferentes. Desta forma, uma fonte de dados pode ser mantida pela própria organização responsável por implementar o sistema baseado no H-KaaS, fonte de dados interna, ou pode ser mantida por uma organização distinta, fonte de dados externas.

Fontes de dados **internas** à organização são, em geral, mais flexíveis visto que as decisões de projeto relativas a elas podem ser feitas pelo mesmo time de engenheiros responsável pelo projeto do extrator de conhecimento. Por outro lado, fontes **externas** à organização geralmente possuem uma interface de acesso predefinida, geralmente imutável do ponto de vista da organização que está implementando a H-KaaS. Dessa forma, o número de tecnologias utilizadas para comunicação, autenticação e processamento de dados fica restrito àquelas impostas pela organização detentora dos dados.

No contexto do acesso aos dados, uma fonte de dados **local** está localizada no mesmo contexto de execução onde seu extrator de conhecimento é executado, permitindo uma comunicação rápida e estável. No entanto, uma fonte de dados **remota** pode estar em uma rede de computadores distinta que, ocasionalmente, pode ficar indisponível. Essas fontes são, em geral, menos estáveis e mais lentas do que fontes locais.

4.1.1.2. MODELO DE DADOS OU CONHECIMENTO

As fontes de dados podem ser classificadas em relação ao tipo de dado, informação ou conhecimento provido. Fontes de dados baseadas em **dados brutos** proveem acesso a conjuntos de dados pouco processados, com pouca organização. Como exemplo, podem ser citados: dados clínicos de pacientes, estatísticas, arquivos de texto semiestruturados, diretrizes clínicas escritas em linguagem natural, periódicos, imagens, entre outros.

Em contrapartida, fontes de dados baseadas em **modelos de conhecimento** possuem um modelo interno, capaz de permitir consultas ao conhecimento modelado. Essas, podem ser categorizadas sob duas perspectivas: quanto ao tipo de conhecimento representado e quanto ao paradigma da representação do conhecimento utilizado.

Do ponto de vista do tipo de conhecimento, esses modelos, baseados nas definições de Sun, Merrill e Peterson (2001), podem ser classificados em modelos de conhecimento **implícitos** ou modelos de conhecimento **explícitos**. Os modelos de conhecimento explícitos podem ser expressados de maneira formal e são geralmente baseados em regras e símbolos, como por exemplo, ontologias de domínio, que são projetadas a partir de conceitos e regras de certa área do conhecimento e, através de algoritmos raciocinadores, são capazes de responder a consultas específicas. Por outro lado, modelos implícitos são gerados de maneira procedural sem que necessariamente sejam produzidas regras explícitas, como por exemplo as redes

bayesianas e as redes neurais artificiais, que são geradas através de algoritmos e procedimentos, e cujo conhecimento modelado é mais difícil de ser entendido por humanos.

Da perspectiva do paradigma sendo utilizado, considerando as definições apresentadas por Russell e Norvig (2013), os modelos de conhecimento no contexto da inteligência artificial podem ser classificados em diferentes grupos:

- **Simbólico:** Utiliza coleções de símbolos para representar objetos, relações, padrões e processos que, por sua vez, podem ser interpretados por computador, permitindo a criação de novos padrões e símbolos não representados anteriormente (NEWELL; SIMON, 2007);
- **Estatístico-probabilístico:** Utiliza conceitos de probabilidade e estatística para a criação de modelos que permitem a representação eficiente do conhecimento incerto, além de propiciar a aprendizagem a partir de exemplos (RUSSELL; NORVIG, 2013);
- **Conexionista:** Baseia-se na ideia de como o cérebro humano funciona, criando conexões entre diferentes nós em um grafo, que se assemelham a neurônios humanos e suas ligações. Esse paradigma surgiu a partir da proposta de Mcculloch e Pitts (1943), que apresentou o primeiro modelo matemático de um neurônio, e, a partir disso, outros pesquisados vêm tentando criar sistemas cada vez mais complexos que visam simular a inteligência humana.

Por fim, com o intuito de exemplificar a utilização das classificações das fontes de dados quanto aos modelos de dados ou conhecimento, a Tabela 5 mostra as classificações de algumas fontes de dados comuns a sistemas computacionais aplicados à saúde.

Tabela 5 - Classificação de fontes de dados comuns a sistemas da área da saúde

Fonte de dados	Classificação	Tipo	Paradigma
Ontologias	Modelo de conhecimento	Explícito	Simbólico
Redes bayesianas	Modelo de conhecimento	Implícito	Estatístico-Probabilístico
Redes neurais artificiais	Modelo de conhecimento	Implícito	Conexionista
Diretrizes clínicas escritas em linguagem natural	Dados brutos	-	-
Diretrizes clínicas modeladas utilizando lógica	Modelo de conhecimento	Explícito	Simbólico
Livros e periódicos	Dados brutos	-	-

Dados de prontuários eletrônicos	Dados brutos	-	-

Fonte: Elaborado pelo autor.

4.1.1.3. PROCESSAMENTO DE CONSULTAS

A forma de processamento das consultas executadas nas fontes de dados pode ser síncrona ou assíncrona. As fontes **síncronas** processam as consultas de maneira imediata e tentam enviar uma resposta o mais rápido possível. Geralmente, estas proveem os dados como uma resposta direta para cada consulta executada de forma unidirecional, onde os dados fluem a partir da fonte de dados para o extrator de conhecimento. Além disso, outra característica importante das fontes síncronas é que estas, geralmente, são mais fáceis de serem implementadas, visto que não precisam de mecanismos complexos para enfileiramento de consultas ou agendamento de tarefas.

Por outro lado, uma fonte de dados **assíncrona** provê respostas às consultas depois de certo tempo de processamento, ou pode publicar dados para o extrator de conhecimento sem uma consulta ter sido executada. Dessa forma, fontes de dados assíncronas podem ser utilizadas para agregar dados periodicamente e podem implementar padrões assíncronos de entrega de dados como *publisher-subscriber* (SCHMIDT et al., 2013), estratégia de entrega de dados muito utilizada em domínios como internet das coisas, em que mensagens em tempo real são trocadas através de uma fila de mensagens entre duas ou mais aplicações.

Outra característica importante de fontes de dados assíncronas é a utilização de métodos de *call-backs* assíncronos, que são chamadas de função que executam depois de algum tempo de processamento para indicar que um evento específico foi concluído, como, por exemplo, a concussão do processamento de um bloco de dados.

É importante notar que uma mesma fonte de dados pode ser **híbrida**, possuindo interfaces síncronas e assíncronas, dependendo da requisição realizada pelo extrator de conhecimento.

4.1.1.4. COMUNICAÇÃO

A forma de comunicação entre os extratores de conhecimento e as fontes de dados pode ser classificada em dois tipos: unidirecional e bidirecional.

As fontes de dados com comunicação **unidirecional** se limitam a enviar seus dados ao extrator de conhecimento, sem a necessidade de receber resposta, estatísticas ou feedbacks em relação aos dados enviados. No entanto, as fontes **bidirecionais** podem receber dados vindos do serviço provedor de conhecimento para serem utilizados com diferentes fins, como, por exemplo, melhorar ou atualizar os seus modelos de conhecimento internos, coletar estatísticas quanto à utilização dos dados, atualizar regras e permissões de acesso aos dados, entre outros. Os dados recebidos podem estar incluídos diretamente nas consultas, em caso de comunicação síncrona, ou ser enviados pelo extrator de conhecimento de maneira assíncrona.

Na seção seguinte serão apresentados detalhes sobre as interfaces de comunicação com as fontes de dados e como estas podem ser implementadas em uma arquitetura de software baseada na H-KaaS.

4.1.2. INTERFACES DE COMUNICAÇÃO

Devido à grande quantidade de fontes de dados existentes no domínio da saúde, durante o desenvolvimento de uma nova arquitetura de software baseada na H-KaaS, o engenheiro de software deverá escolher as tecnologias e interfaces baseadas no problema a ser resolvido e nas fontes de dados a serem implementadas.

O acesso aos dados pode variar de acordo a interface de comunicação com a fonte e, por isso, requer-se, do extrator de conhecimento, a implementação dos meios necessários para a leitura e extração das informações relevantes.

Dessa forma, as tecnologias escolhidas devem estar relacionadas ao tipo de fonte de dados a ser utilizado e suas classificações. Por exemplo, para obter dados de uma ontologia no formato OWL, um modelo de conhecimento explícito do paradigma simbólico, deve-se procurar tecnologias capazes de raciocinar sob essa forma de representação do conhecimento, como, por exemplo, a Jena API e SPARQL.

De maneira similar, fontes de dados brutos em formato textual, como textos de artigos científicos na área da saúde e diretrizes clínicas, talvez precisem ser manipuladas por algoritmos de processamento de linguagem natural. Por outro lado, fontes de dados remotas, podem estar envelopadas em APIs baseadas em um protocolo de comunicação como o HTTP, com padrões REST e SOAP, tecnologias bastante utilizadas nessa situação, conforme conclusões apresentadas no capítulo 3.

É importante notar que a interface de comunicação com as fontes de dados deve estar bem definida para que ocorra, na medida do possível, o desacoplamento entre a fonte e os serviços provedores de conhecimento. Dessa forma, em alguns casos, se houver necessidade, uma fonte de conhecimento que fique indisponível pode ser substituída por outra similar. Além disso, a especificação da interface de comunicação pode contribuir para o reuso de componentes como, por exemplo, a reutilização de um mesmo extrator de conhecimento para lidar com fontes similares, como as ontologias OWL de um mesmo domínio.

Para algumas fontes de dados remotas, por exemplo, poderá ser feita a obtenção em massa das informações, com o propósito de que seja executada a mineração de dados local através de algoritmos de aprendizagem de máquina. Dessa forma, mesmo que a fonte de dados fique indisponível depois da obtenção dos dados, os servidores de conhecimento ainda serão capazes de executar a extração a partir dos dados previamente obtidos.

De forma geral, ao se projetar uma arquitetura de software baseada na H-KaaS, deve-se seguir os seguintes passos a fim de escolher as tecnologias e interfaces de comunicação entre os extratores de conhecimento e a fontes de dados:

1. Definir quais fontes de dados estão disponíveis para ser usadas pela arquitetura;
2. Classificar cada fonte de dados quanto sua localização, forma de comunicação, modelo de dado ou conhecimento e forma de processamento utilizada;
3. Para cada fonte de dados, avaliar quais tecnologias são viáveis de se utilizar, de acordo com sua classificação. Por exemplo, fontes externas à organização podem ter que utilizar uma interface de comunicação e sistema de autenticação pré-definida pela organização responsável pela mesma. Por outro lado, uma fonte de dados local, como uma ontologia de domínio em formato OWL, pode ser acessada utilizando diferentes mecanismos de manipulação de ontologias e algoritmos raciocinadores;
4. Baseadas nas tecnologias escolhidas, definir e padronizar interfaces de comunicação que serão utilizadas e quais extratores de conhecimento precisam ser implementados;
5. Por fim, deverá ser feita a implementação dos extratores de conhecimento de acordo com as interfaces de comunicação especificadas a fim de se expor, para os servidores de conhecimento, as informações por eles requisitadas.

Em conclusão, a especificação das fontes de dados em uma arquitetura provedora de conhecimento deve ser uma das primeiras ações tomadas pelo projetista a fim de facilitar o desenvolvimento de outros componentes do sistema. Além disso, as classificações das fontes

de dados poderão ser utilizadas para comparar sistemas de software existentes, facilitando seu entendimento e contribuindo para uma melhor compreensão de como as fontes de dados estão sendo utilizadas em arquiteturas cujo objetivo é o compartilhamento de conhecimento no domínio da saúde.

4.2. SERVIÇO PROVEDOR DE CONHECIMENTO

Segundo Xu e Zhang (2005), de acordo com as definições presentes no paradigma de conhecimento como serviço, o serviço provedor de conhecimento tem como objetivo acessar e processar os dados de múltiplas fontes de dados, gerir modelos de conhecimento e permitir consultas feitas pelos aplicativos consumidores de conhecimento.

Dessa forma, um dos principais objetivos da arquitetura de referência H-Kaas é a centralização do acesso ao conhecimento por meio de um serviço provedor de conhecimento, capaz de extrair dados e conhecimento de diferentes fontes e prover, de maneira padronizada, respostas às consultas feitas por aplicativos consumidores de conhecimento através de uma interface de comunicação bem definida.

No contexto da H-KaaS, o serviço provedor de conhecimento é constituído de três subcomponentes: (1) os extratores de conhecimento, (2) os servidores de conhecimento e (3) o servidor da API de comunicação.

Os extratores de conhecimento são componentes responsáveis pela comunicação e obtenção de dados de uma ou mais fontes de dados. A extração de conhecimento pode ser feita à medida que consultas são realizadas pelos aplicativos consumidores, através de inferências a ontologias, consultas a documentos e modelos de conhecimento ou por meio do acesso a outras fontes de dados. Alternativamente, o conhecimento pode ser extraído de maneira assíncrona, a partir de bancos de dados utilizando algoritmos de aprendizagem de máquina e técnicas de KDD.

Paralelamente, os servidores de conhecimento são responsáveis por prover respostas às consultas realizadas através da API de comunicação com a ajuda dos dados fornecidos pelos extratores de conhecimento e algoritmos internos. Um mesmo serviço provedor de conhecimento pode possuir múltiplos servidores de conhecimento, cada um responsável por responder a certo subconjunto de consultas, porém, todos compartilham os mesmos componentes extratores de conhecimento e interfaces de comunicação com os aplicativos consumidores.

Adicionalmente, com o objetivo de responder às consultas dos aplicativos consumidores, o serviço provedor de conhecimento possui um componente chamado **servidor da API de comunicação**, cujo objetivo é fornecer respostas padronizadas que sejam facilmente entendidas pelos aplicativos consumidores. Além disso, esse componente é responsável pela autenticação e monitoramento das consultas realizadas, sendo um intermediário entre o aplicativo consumidor e os diversos servidores de conhecimento.

4.2.1. SUBCOMPONENTES

Nas subseções seguintes serão fornecidos mais detalhes sobre os subcomponentes do serviço provedor de conhecimento e suas interfaces de comunicação.

4.2.1.1. EXTRATORES DE CONHECIMENTO

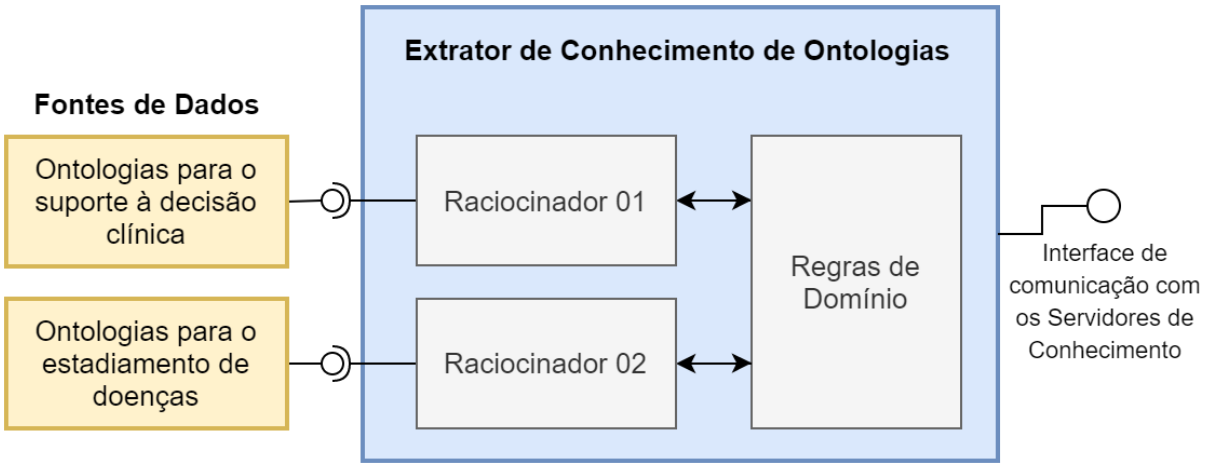
Os extratores de conhecimento são responsáveis pela comunicação e extração de conhecimento a partir de diversas fontes, sendo utilizados como intermediários entre os servidores de conhecimento e as fontes de dados.

No domínio da saúde, um extrator de conhecimento, por exemplo, pode ser responsável por lidar com fontes de dados baseadas em ontologias de domínio (GRUBER, 1993), diretrizes clínicas, textos acadêmicos, modelos de conhecimento baseados em aprendizagem de máquina, entre outras.

A Figura 24 mostra, a partir de um diagrama de componentes modelado em UML, a definição de um extrator de conhecimento conceitual, baseado em ontologias de domínio, que pode ser implementado e organizado para ser utilizado no domínio da saúde. O extrator em questão se utiliza de algoritmos raciocinadores (KHAMPARIA; PANDEY, 2017) e de regras de domínio de suas fontes de dados para inferir sobre uma ou mais ontologias, a fim de responder a consultas recebidas em sua interface de comunicação disponibilizada aos servidores de conhecimento.

Cada fonte de dados possui regras de extração específicas implementadas por seus respectivos extratores. Sendo assim, para cada nova fonte de dados adicionada a uma arquitetura baseada na H-KaaS, regras de extração e acesso precisam ser escritas para que seja possível sua integração com o sistema.

Figura 24 - Arquitetura conceitual de um possível extrator de conhecimento de ontologias de domínio



Fonte: Elaborado pelo autor.

Além disso, estão definidos no contexto do extrator de conhecimento algoritmos auxiliares baseados nas regras de domínio da fonte de dados, responsáveis por diversas funções comuns para a leitura e manipulação das informações das fontes de dados integradas ao sistema. Alguns exemplos desses algoritmos são: funções para leitura de arquivos de texto, cálculos necessários para execução de consultas em uma certa fonte, leitura e conversão de imagens, filtros para garantir o anonimato dos dados de pacientes, entre outros.

Também, um extrator de conhecimento pode utilizar outros mecanismos de persistência presentes no serviço provedor de conhecimento, para complementar os dados fornecidos pela fonte, ou melhorar o conhecimento já modelado. Por exemplo, um extrator de conhecimento baseado em mineração de dados pode se utilizar de dados de pacientes, armazenados no componente banco de dados de consultas, para automaticamente, e periodicamente, extrair conhecimento útil, que poderá ser disponibilizado em consultas futuras ou utilizado como *feedback* enviado à fonte de dados.

Os extratores de conhecimento podem ser classificados em **específicos** ou **genéricos**. Um extrator de conhecimento genérico pode ser reutilizado para a extração e manipulação de diversas fontes similares. Por exemplo, um extrator de conhecimento baseado em ontologias de domínio pode ser genérico o suficiente para suportar múltiplas ontologias de diferentes domínios, o que permite seu reuso. De maneira similar, extratores baseados em mineração de dados podem ser reutilizados para lidar com fontes de dados semelhantes. Por outro lado, os extratores de conhecimento específicos são desenvolvidos com o objetivo de suportar apenas

uma fonte de dados particular. Dessa forma, ele estará tão acoplado à sua fonte de dados que não poderá ser reutilizado ou instanciado para utilização com outras fontes.

Cada extrator de conhecimento pode oferecer uma interface de comunicação distinta para os diversos servidores de conhecimento implementados e, embora os extratores de conhecimento possam manipular e filtrar os dados de cada fonte de dados, não é sua responsabilidade garantir a anonimização desses dados.

Em um contexto de encadeamento de arquiteturas H-KaaS, uma das arquiteturas, por meio de um extrator de conhecimento especificamente desenvolvido, pode exercer o papel de um aplicativo consumidor de conhecimento. Nesse caso, um dos extratores de conhecimento implementaria os protocolos de comunicação especificados na API de comunicação e seria responsável por obter conhecimento disponibilizado pelo outro sistema.

Outra característica importante dos extratores de conhecimento é que eles podem agregar dados de múltiplas fontes de dados, gerenciando as dependências entre elas e possibilitando a execução de consultas complexas. Por exemplo, para que uma consulta seja realizada na fonte A, talvez seja necessário um dado fornecido pela fonte B. Dessa forma, o extrator poderá gerenciar essas regras de dependências, fornecendo aos servidores de conhecimento uma interface de alto nível, abstraindo essas dependências entre fontes de dados distintas.

Por fim, cada extrator de conhecimento está associado ao tipo de fonte de dados e forma de representação do conhecimento e raciocínio utilizados por suas fontes. Por exemplo, extratores de conhecimento baseado em ontologias de domínio são capazes de extrair conhecimento a partir dessas ontologias, uma forma de representação do conhecimento baseada no paradigma simbólico. De forma similar, extratores baseado em modelos de aprendizagem de máquina, devem ser capazes de consultar modelos de conhecimento implícitos, gerados por tais algoritmos.

4.2.1.2. SERVIDOR DE CONHECIMENTO

Um servidor de conhecimento, no contexto da arquitetura de referência proposta, é o componente responsável por responder a consultas feitas pelos aplicativos consumidores, de acordo com o conhecimento extraído por um ou mais extratores de conhecimento, através do uso das interfaces de comunicação providas pelo servidor da API de comunicação.

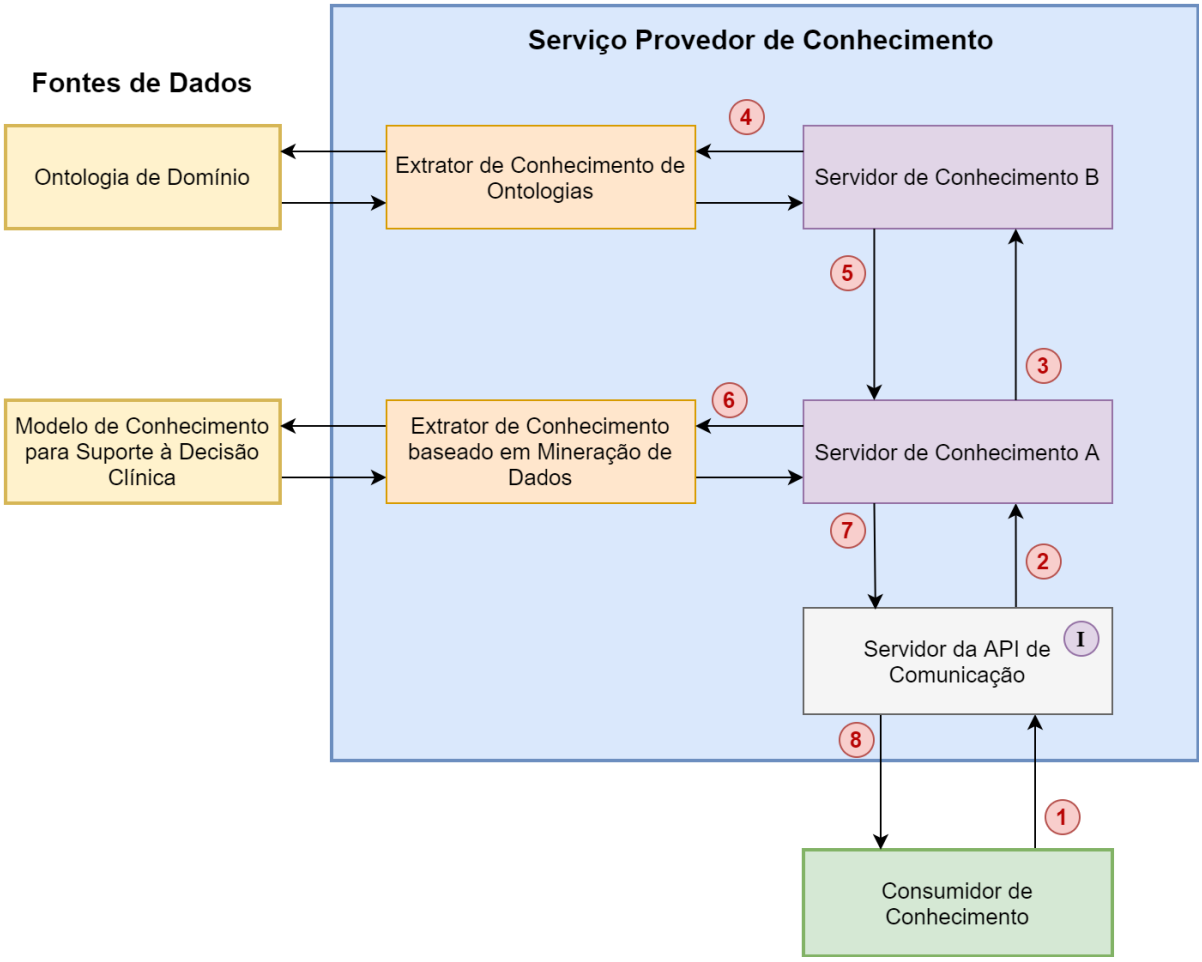
De maneira similar, Xu e Zhang (2005), ao descreverem os principais componentes do paradigma de conhecimento como serviço, definem o servidor de conhecimento como o componente encarregado de fornecer o conhecimento extraído através do uso de extratores de conhecimento apropriados, de maneira a responder às consultas executadas pelos aplicativos consumidores.

É importante notar que um serviço provedor de conhecimento pode possuir um ou mais servidores de conhecimento distintos que compartilham dos mesmos extratores de conhecimento e do servidor da API de comunicação. Dessa forma, além de permitir o reuso de componentes, essa abordagem possibilita que cada servidor de conhecimento seja especializado em responder a grupos de consultas de um subdomínio. Por exemplo, no domínio da saúde, um mesmo serviço provedor de conhecimento pode disponibilizar um servidor de conhecimento específico para diagnóstico de doenças relacionadas à oncologia e outro para estadiamento de doenças como a doença renal crônica.

Servidores de conhecimento em um mesmo serviço provedor de conhecimento podem também apresentar uma relação de dependência entre si, permitindo o encadeamento de consultas ou composição de serviços internos mais abrangentes, nesse caso, sem a necessidade de executar o encadeamento de instâncias de arquiteturas H-KaaS. Por exemplo, um servidor de conhecimento A pode executar uma consulta no servidor de conhecimento B, a fim de complementar as respostas de suas próprias consultas. A Figura 25 mostra, através de uma visão simplificada da arquitetura, como múltiplos servidores de conhecimento podem ser encadeados para responder a consultas complexas, viabilizando a criação de regras de dependências entre eles, compartilhando uma mesma instância do servidor da API de comunicação (I).

Desta forma, com o objetivo de exemplificar essa dependência, a figura mostra uma consulta (1) sendo enviada pelo consumidor de conhecimento ao servidor da API de comunicação que, por sua vez, a encaminha (2) para o servidor de conhecimento A. Em seguida, uma nova consulta (3) é executada no servidor de conhecimento B a fim de complementar os dados solicitados, através de consultas (4) executadas no extrator de conhecimento de ontologias. O resultado dessa consulta (5), em conjunto com a extração de conhecimento feito pelo extrator de conhecimento baseado em mineração de dados (6), possibilita o servidor de conhecimento A responder (7) a consulta original (2), enviando seus resultados (8) ao consumidor de conhecimento através do servidor da API de comunicação.

Figura 25 - Composição de um serviço provedor conhecimento baseado em múltiplos servidores de conhecimento encadeados



Fonte: Elaborado pelo autor.

Também, como essa interface de comunicação é compartilhada pelos diferentes consumidores de conhecimento, a partir do servidor da API de comunicação, o acesso ao conhecimento provido é facilitado, pois um mesmo consumidor de conhecimento poderá acessar múltiplos serviços distintos com a mesma implementação do cliente da API de comunicação. Além disso, a composição de servidores de conhecimento, internos ao serviço provedor de conhecimento, pode ser abstraída dos consumidores.

Diferentemente das fontes de dados e dos consumidores de conhecimento, os servidores de conhecimento, por estarem acoplados ao serviço provedor de conhecimento, parte central da arquitetura H-KaaS, devem ser mantidos e especificados pela mesma organização, visto que centralizarão as consultas e o acesso a fontes de conhecimento e seus respectivos extratores e,

para isso, precisarão implementar interfaces de comunicação com os extratores, com outros servidores de conhecimento e com o servidor da API de comunicação.

Um mesmo servidor de conhecimento pode se comunicar com um ou mais extratores de conhecimento dependendo da complexidade das consultas a serem respondidas. Além disso, eles podem possuir modelos de conhecimento internos, funções de mapeamento, bancos de dados de consultas e pacientes próprios, ou qualquer outro subcomponente que se faça necessário para a sua execução. Desta forma, o servidor de conhecimento se torna um dos componentes mais flexíveis da H-KaaS pois, pode representar diferentes subsistemas, principalmente quando usado na comparação de sistemas existentes.

O **banco de dados de consultas** é um componente opcional que pode ser utilizado para armazenar o histórico de consultas, estatísticas ou dados de pacientes relativos àquele servidor de conhecimento específico. Esses dados podem ser utilizados internamente ou expostos aos extratores de conhecimento com diferentes objetivos, como, por exemplo, melhoria de modelos de conhecimento, validação de consultas, criação de relatórios, entre outros. Desta forma, fica claro que os extratores podem, além de extrair conhecimento das fontes, utilizar o banco de dados de consultas e outros subcomponentes dos servidores de conhecimento em sua execução.

Além do banco de dados de consultas, os servidores de conhecimento podem ser compostos por **outros subcomponentes**, que representam genericamente componentes específicos ao problema a ser resolvido. Esses componentes podem ter funções diversas como, por exemplo: cálculos matemáticos e regras para transformações de dados; scripts de execução periódica; componentes adicionais de monitoramento; sistemas de cache; entre outros.

4.2.1.3. SERVIDOR DA API DE COMUNICAÇÃO

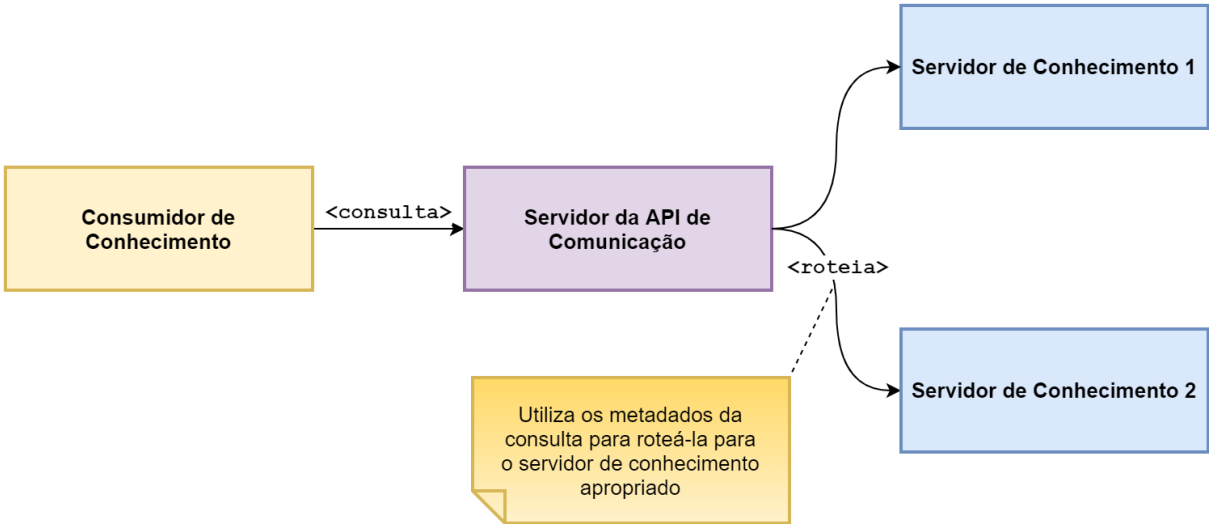
O servidor da API de comunicação tem como principal função expor uma interface de comunicação compartilhada aos aplicativos consumidores, sendo utilizado como componente intermediador entre os servidores e os consumidores de conhecimento.

Além disso, outras funções do servidor da API de comunicação são o monitoramento, o roteamento e a autenticação das consultas executadas através de sua interface.

A implementação de um servidor da API de comunicação é obrigatória para todas as instâncias de um sistema baseado na arquitetura H-Kaas, visto que é a partir deste que a arquitetura será capaz de fornecer conhecimento para os aplicativos consumidores.

É importante notar que existe apenas uma instância do servidor da API de comunicação sendo executada, sendo essa compartilhada por todos os servidores de conhecimento instanciados, permitindo a centralização do acesso e a padronização da entrega do conhecimento extraído ao consumidores. Sendo assim, faz-se necessário o desenvolvimento de um mecanismo de roteamento de requisições e registro de novos servidores, para que estas sejam direcionadas para o servidor de conhecimento responsável por respondê-las. A Figura 26 mostra um diagrama detalhando a execução de uma consulta e como o servidor da API de comunicação roteia os dados para o servidor de conhecimento apropriado.

Figura 26 - Detalhes do roteamento de consultas feitas pelo servidor da API de comunicação em um sistema com múltiplos servidores de conhecimento



Fonte: Elaborado pelo autor.

Devido à grande diversidade de aplicativos consumidores e de serviços de conhecimento no domínio da saúde, existem várias formas de implementar o servidor da API de comunicação e, por essa razão, é importante notar que as escolhas feitas durante sua fase de projeto impactam diretamente nas tecnologias utilizadas nos subcomponentes dos consumidores de conhecimento, especialmente no cliente da API de comunicação.

A especificação da interface de comunicação e provimento de dados é uma atribuição do servidor da API de comunicação, porém, como os dados transmitidos no domínio da saúde são, em sua maior parte, sensíveis, geralmente serão necessárias a utilização de tecnologias para autenticação, como, por exemplo, a utilização de chaves compartilhadas, e criptografia, com o objetivo de proteger os dados transmitidos. Por outro lado, em algumas situações, essas proteções não serão necessárias – por exemplo, no caso de o aplicativo consumidor e o servidor

estarem sendo executados em uma mesma rede virtual privada isolada ou em situações em que os dados ou conhecimento transmitidos forem públicos.

O **módulo de controle de acesso** é o subcomponente responsável por validar o acesso de aplicativos consumidores de conhecimento e, por meio de um banco de dados com informações de regras de autenticação, chamado de banco de dados de autenticação, tem a responsabilidade de permitir ou bloquear requisições individuais enviadas pelos aplicativos consumidores.

O **banco de dados de autenticação** é responsável por armazenar as informações de autenticação necessárias para que o módulo de controle de acesso possa decidir sobre a autenticidade de uma requisição. Sendo assim, nele podem ser armazenadas chaves compartilhadas, regras de acesso, listas de IPs, ou quaisquer outros artefatos que possam ser utilizados durante a autenticação do acesso.

Uma regra de autenticação pode, por exemplo, bloquear o acesso a um método ou a um tipo de consulta baseada nas permissões de um aplicativo consumidor de conhecimento. Por exemplo, um profissional da atenção básica, ao utilizar um aplicativo consumidor de conhecimento, pode ter acesso a consultas relacionadas à tomada de decisão clínica, enquanto um paciente, ao utilizar um mesmo aplicativo consumidor de conhecimento, pode ter acesso à funções limitadas do conhecimento provido.

As tecnologias utilizadas no banco de dados de autenticação ficam a cargo do projetista, engenheiro de software responsável por sua implementação, variando de acordo com a sua aplicação, podendo ser, por exemplo, arquivos de textos com regras semiestruturadas ou banco de dados relacionais.

O **módulo de monitoramento** é o subcomponente responsável por coletar estatísticas de utilização e auxiliar o módulo de controle de acesso na tomada de decisão. Os dados de utilização dos serviços podem ser armazenados em um banco de dados, chamado pela H-Kaas de **banco de dados de utilização**, que, por sua vez, armazena metadados sobre as consultas realizadas por cada aplicativo consumidor de conhecimento. O histórico das consultas pode ser utilizado para diferentes propósitos, como geração de relatórios, monitoramento e correção de erros, além da possibilidade de serem empregados para detecção de anomalias no acesso ao conhecimento provido.

Sendo assim, em algumas situações, com o objetivo de impedir o acesso indevido ao conhecimento, caso sejam detectadas anomalias nas consultas a partir dos dados armazenados, o módulo de monitoramento pode ser capaz de criar ou alterar regras de acesso em tempo real,

diretamente no banco de dados de autenticação, de forma a mitigar ataques como o roubo de conhecimento.

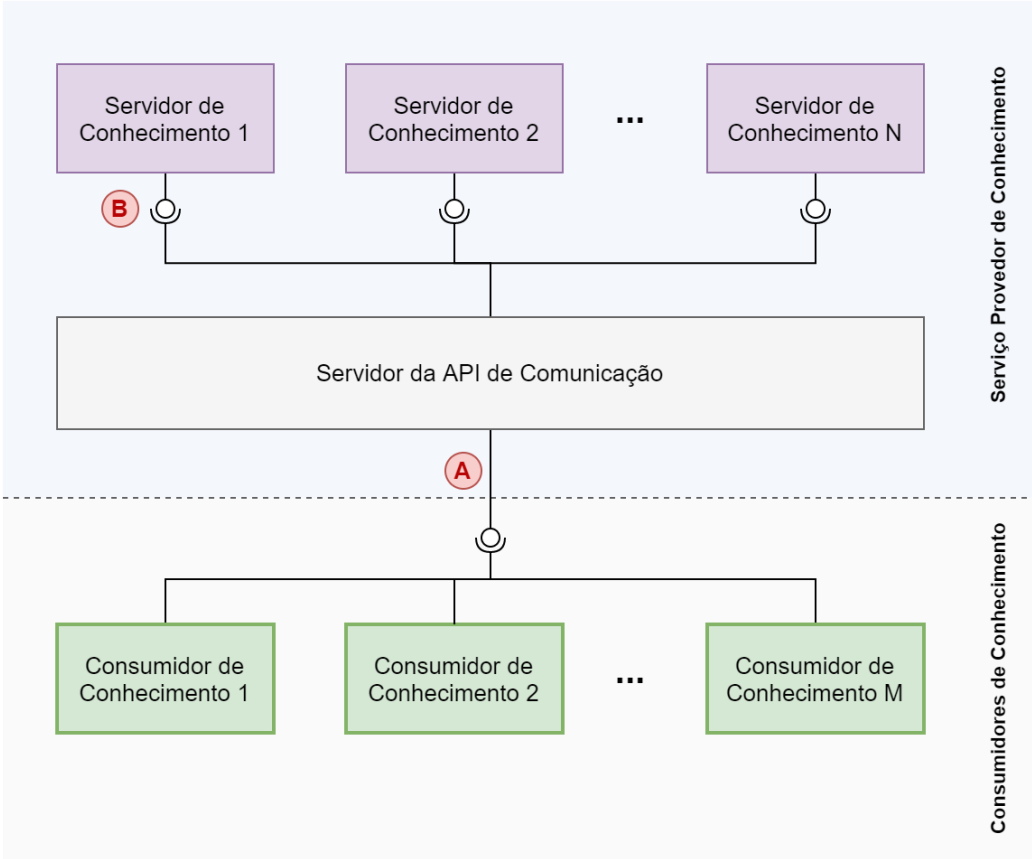
Cada projetista deverá decidir quais metadados das consultas serão armazenados, porém, alguns exemplos são: tempo de respostas, identificações do aplicativo consumidor, tempo de processamento, identificação do servidor de conhecimento, velocidade de transmissão, método executado, etc.

A maioria das propostas estudadas na seção 3 não define como é feito o monitoramento das consultas e autenticação dos aplicativos consumidores de conhecimento. Além disso, já existem arquiteturas de referência e padrões dedicados a solucionar esse problema, como, por exemplo, o padrão *eXtensible Access Control Markup Language* (XACML), que visa permitir a análise e controle de requisições de acordo com regras definidas em políticas de acesso bem definidas (OASIS STANDARD, 2013). Sendo assim, podemos concluir que essas decisões de projeto devem ser tomadas pelo arquiteto ao se adaptar a arquitetura de referência H-KaaS para um sistema específico, sendo possível tanto a utilização dos módulos definidos pela H-KaaS, quanto a adaptação de padrões já existentes para o controle de acesso e monitoramento.

4.2.2. INTERFACES DE COMUNICAÇÃO

A comunicação entre o serviço provedor de conhecimento e os aplicativos consumidores se dá através da API de comunicação provida pelo servidor da API de comunicação, conforme pode ser visto na Figura 27. Nela, através de um diagrama de componentes modelado utilizando a linguagem UML, é apresentada a interface de comunicação (A) provida pelo servidor da API de comunicação aos aplicativos consumidores, sendo esse, o único meio de comunicação previsto pela H-KaaS entre o serviço provedor de conhecimento e seus consumidores de conhecimento. Adicionalmente, podem ser vistas as interfaces (B), disponibilizadas pelos servidores de conhecimento, utilizadas pelo servidor da API de comunicação ao encaminhar requisições dos aplicativos consumidores.

Figura 27 - Interfaces de comunicação entre o serviço provedor de conhecimento e os consumidores



Fonte: Elaborado pelo autor.

Segundo Masse (2011), uma API é um serviço web baseado em interfaces de programação bem definidas, que permite a comunicação entre aplicativos. Em outras palavras, uma API pode ser considerada um meio pelo qual se permite a comunicação entre dois programas de computador.

Para a especificação da API de comunicação com os consumidores de conhecimento, recomenda-se o uso dos princípios mencionados por Bloch (2006), que auxiliam a criação de uma boa API de comunicação, sendo os principais deles:

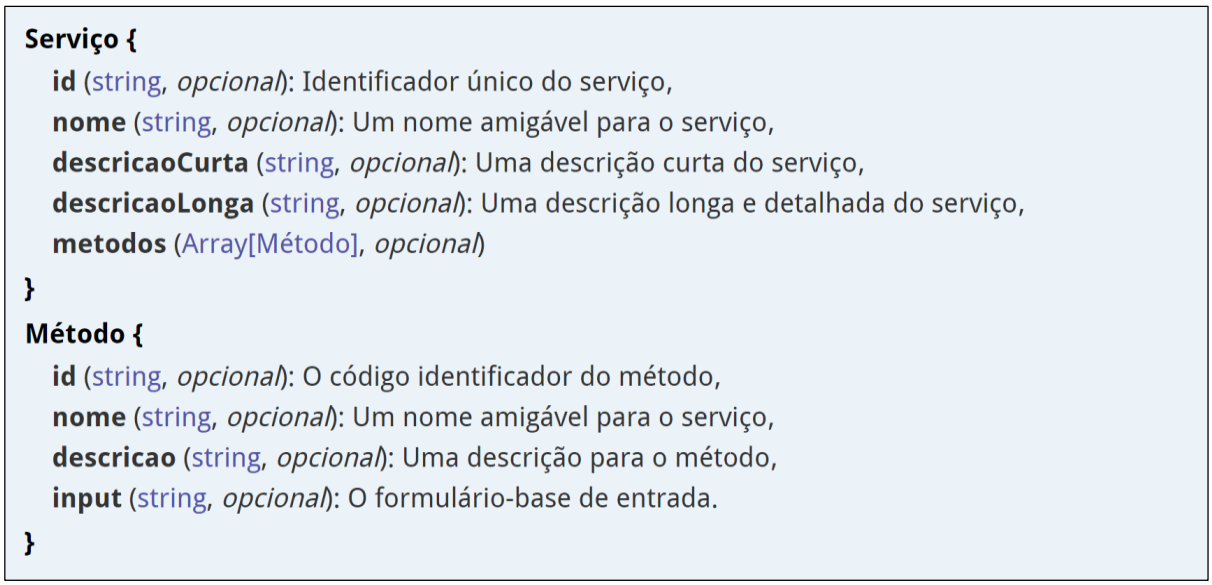
- Uma API deve ser pequena e simples;
- Quando ocorrerem, os erros devem ser reportados imediatamente e de forma consistente;
- Deve-se manter um número pequeno de parâmetros;
- As funções e objetos devem estar bem documentados.

As tecnologias utilizadas para a implementação e especificação da API de comunicação devem ser definidas pelo engenheiro de software responsável por adaptar a arquitetura de referência H-Kaas ao sistema a ser desenvolvido. Porém, com base nas conclusões geradas a partir do estudo de sistemas para o compartilhamento de conhecimento no domínio da saúde, apresentadas na seção 3, e nas características propostas pela H-KaaS, observou-se a necessidade da especificação de pelo menos dois tipos de entidades:

- **Serviços:** Responsável por detalhar e listar os serviços oferecidos pelos servidores de conhecimento. Cada serviço deve possuir suas descrições, identificação e, principalmente, uma lista com os métodos disponíveis. Essas informações serão importantes para que os aplicativos consumidores de conhecimento sejam capazes de enviar suas requisições e que estas sejam roteadas para o servidor de conhecimento correto.
- **Métodos:** Os métodos são funções que podem ser executadas pelos aplicativos consumidores, possibilitando o acesso ao conhecimento disponibilizado pelos servidores de conhecimento disponíveis.

Com objetivo de exemplificar como os objetos e requisições podem ser especificados, a Figura 28 mostra uma sugestão de especificação de entidades que podem ser utilizadas para definir os protocolos de comunicação da API.

Figura 28 - Exemplo de como podem ser definidas as entidades serviço e método em uma implementação da API de comunicação



Fonte: Elaborado pelo autor.

Em conclusão, a arquitetura H-KaaS não define quais tecnologias devem ser utilizadas para comunicação entre o servidor da API de comunicação e os consumidores de conhecimento, porém, o estudo de propostas similares no domínio da saúde, apresentadas no capítulo 3, em conjunto com os exemplos disponibilizados nesta seção, poderão servir de inspiração para as decisões a serem tomadas ao se adaptar a H-KaaS para uma arquitetura de software específica.

4.3. CONSUMIDOR DE CONHECIMENTO

A arquitetura H-KaaS, de maneira similar ao paradigma KaaS, centraliza o acesso ao conhecimento e prevê a possibilidade de ser acessada por diferentes aplicativos consumidores de conhecimento. Esses aplicativos se utilizam de uma interface de comunicação, provida pelo componente servidor da API de comunicação, para realizar consultas a um ou mais serviços de conhecimento e obter respostas referentes a essas consultas.

Segundo Xu e Zhang (2005), ao especificar os componentes do paradigma KaaS, os consumidores de conhecimento são aplicações que usam o conhecimento do serviço provedor em seu processo de tomada de decisão. Dessa forma, no contexto da arquitetura H-KaaS, um aplicativo consumidor de conhecimento é um software capaz de se comunicar com o serviço provedor de conhecimento, a fim de obter dados disponibilizados por servidores de

conhecimento que, por sua vez, são responsáveis por agregar, processar e executar consultas a diferentes fontes de dados.

Nesse contexto, os aplicativos consumidores de conhecimento podem variar de acordo com sua plataforma de acesso, como websites, aplicativos para dispositivos móveis, sistemas embarcados para monitoramento de pacientes, etc., e quanto à sua natureza e objetivos, como por exemplo, prontuários eletrônicos, sistemas de suporte à decisão clínica, sistemas educacionais, aplicações para gerenciamento de recursos hospitalares, sistemas especialistas, entre outros.

A implementação de cada aplicativo consumidor pode variar conforme seu objetivo e tecnologias, desde que a comunicação com o serviço provedor de conhecimento ocorra conforme a especificação da interface de comunicação disponibilizada. Nesse contexto, cada aplicativo é responsável por cumprir os requisitos de comunicação e segurança especificados pelo projetista da arquitetura sendo desenvolvida, a fim de autenticar, limitar e identificar as consultas realizadas. A falha no cumprimento dessas especificações impossibilitará o acesso ao serviço provedor de conhecimento, sendo este responsável pela validação e monitoramento desse acesso.

De maneira similar ao paradigma KaaS, uma característica importante dos aplicativos consumidores de conhecimento é a possibilidade de serem desenvolvidos e mantidos por diferentes organizações, permitindo o acesso centralizado ao conhecimento que, de outra forma, seria impossível de ser obtido. Dessa forma, a utilização de uma arquitetura H-KaaS pode contribuir para o compartilhamento de conhecimento entre diferentes organizações, simplificando seu acesso. Além disso, uma mesma implementação de um aplicativo consumidor de conhecimento pode ser instanciada de formas distintas e independentes entre si, a fim de atender a requisitos de diferentes organizações, contribuindo para a redução de custos de implantação e manutenção desses aplicativos.

4.3.1. SUBCOMPONENTES

Um aplicativo consumidor de conhecimento pode ser composto por vários subcomponentes responsáveis pela comunicação, armazenamento de dados locais, interação com os usuários, entre outros. Nas subseções seguintes, serão detalhados os seus subcomponentes previstos pela H-KaaS e como estes podem ser utilizados.

4.3.1.1. CLIENTE DA API DE COMUNICAÇÃO

O cliente da API de comunicação é responsável pela implementação da interface cliente do servidor da API de comunicação. Dessa forma, a fim de acessar o conhecimento provido por qualquer um dos servidores de conhecimento disponíveis, esse componente se torna obrigatório em qualquer aplicativo consumidor de conhecimento, tornando possível a execução dos métodos providos pela interface de comunicação.

A implementação desse componente pode ser bastante variada e dependerá das tecnologias escolhidas para desenvolvimento do aplicativo consumidor de conhecimento. Além disso, o cliente da API de comunicação poderá trocar informações de maneira bidirecional com a interface gráfica do usuário e com o banco de dados local, internos ao aplicativo consumidor de conhecimento, de maneira a oferecer respostas às consultas de seus usuários.

4.3.1.2. BANCO DE DADOS LOCAL

A arquitetura H-KaaS especifica o componente chamado de banco de dados local como forma de abstrair os métodos de armazenamento utilizados por aplicativos consumidores de conhecimento no domínio da saúde. Dessa forma, o banco de dados local do aplicativo consumidor de conhecimento é um componente opcional, que poderá ser utilizado para armazenar informações necessárias para sua execução e de seus subcomponentes. Como exemplos de uso desse banco de dados, podem ser citados:

- O armazenamento de informações sobre autenticação na interface de comunicação ou de usuários locais do aplicativo consumidor;
- A implementação de uma camada de cache com o objetivo de acelerar a execução de consultas;
- A organização de dados temporários a fim de agregar e otimizar as consultas executadas aos servidores de conhecimento;
- O armazenamento de informações sobre tarefas a serem executadas e agendamento de consultas à API.

4.3.1.3. INTERFACE GRÁFICA DO USUÁRIO

A interface gráfica do usuário do aplicativo consumidor de conhecimento é um componente opcional, responsável por permitir a interação de usuários com o software. Esses

usuários, no domínio da saúde, podem ser, por exemplo, médicos, profissionais da atenção básica, professores, administradores, pacientes, entre outros.

Além disso, a interface gráfica do usuário pode ser implementada utilizando diferentes tecnologias e, dependendo dos requisitos do sistema, pode permitir o acesso às funcionalidades do aplicativo consumidor de conhecimento em diferentes dispositivos.

Nem todos os aplicativos consumidores de conhecimento precisam implementar uma interface gráfica do usuário. Por outro lado, em algumas aplicações, pode ser necessária a implementação de múltiplas interfaces gráficas por diferentes motivos, como:

- Dividir usuários com base em suas permissões de acesso, como, por exemplo, prover uma interface gráfica dedicada para administradores do sistema e outra para profissionais de saúde;
- Permitir o acesso ao aplicativo consumidor de conhecimento a partir de diferentes dispositivos;
- Implementar interfaces gráficas dedicadas para certos tipos de profissionais de saúde, como, por exemplo, uma interface para o auxílio na tomada de decisão na saúde e outra interface gráfica voltada para facilitação da execução das decisões tomadas.

4.3.1.4. OUTROS SUBCOMPONENTES

Devido à grande quantidade de aplicações capazes de serem classificadas como consumidores de conhecimento no domínio da saúde, a arquitetura de referência H-KaaS abstrai alguns de seus subsistemas por meio do componente chamado outros subcomponentes. Este, por sua vez, pode representar quaisquer subsistemas presentes no aplicativo consumidor de conhecimento não descritos pela H-KaaS.

Esses componentes são específicos da aplicação consumidora de conhecimento e devem ser definidos durante o projeto da arquitetura do software a ser desenvolvido. Além disso, ao se analisar sistemas existentes no domínio da saúde, a especificação de outros componentes, dentro do contexto do aplicativo consumidor de conhecimento, dá à arquitetura de referência H-KaaS uma maior flexibilidade e permite ao engenheiro de software um melhor entendimento dos sistemas analisados.

4.3.2. INTERFACES DE COMUNICAÇÃO

Conforme discutido no capítulo 3, ao se analisar sistemas e arquiteturas projetadas para o compartilhamento de conhecimento no domínio da saúde, conclui-se que existe uma grande heterogeneidade em relação a tecnologias para comunicação e implementação de interfaces entre os serviços de conhecimento e os aplicativos consumidores.

É responsabilidade do aplicativo consumidor de conhecimento, através de seu subcomponente cliente da API de comunicação, implementar e seguir os protocolos de comunicação especificados pelo servidor da API de comunicação. Porém, as definições das tecnologias utilizadas para este fim ficam a cargo da organização responsável pelo aplicativo consumidor de conhecimento e seus engenheiros.

É importante notar que a arquitetura de referência H-KaaS não prevê a comunicação direta entre aplicativos consumidores de conhecimento distintos, tornando-os independentes entre si. Dessa maneira, a comunicação entre o serviço provedor de conhecimento e os aplicativos consumidores é feita de forma a centralizar e padronizar o acesso ao conhecimento.

4.4. CONSIDERAÇÕES FINAIS

Este capítulo detalhou uma proposta de arquitetura de referência baseada no paradigma de conhecimento como serviço para o domínio da saúde, chamada H-KaaS. O desenvolvimento dessa arquitetura realizou-se através da adaptação do paradigma KaaS para o domínio da saúde e da análise sistemática de propostas de arquiteturas e sistemas de software reais cujo objetivo é o compartilhamento de conhecimento nesse domínio.

Desse modo, foi especificado detalhadamente cada um de seus componentes, interfaces de comunicação, possíveis consumidores de conhecimento e fontes de dados, além de particularidades de sua implementação.

Finalmente, a arquitetura de referência H-KaaS mostra-se alinhada com sistemas de compartilhamento de conhecimento existentes no domínio e espera-se que, em conjunto com o paradigma Kaas, contribua para o desenvolvimento de novos sistemas centralizados de compartilhamento de conhecimento na área da saúde, além de facilitar o estudo de arquiteturas e sistemas existentes no domínio.

No próximo capítulo, serão apresentados os estudos de caso efetuados, cujo objetivo foi validar e exemplificar o uso da arquitetura H-KaaS, descrevendo como ela pode ser instanciada

em diferentes subdomínios da saúde, e detalhar o fluxo de conhecimento entre seus componentes.

5. ESTUDOS DE CASO

Com o objetivo de validar e exemplificar a utilização da arquitetura proposta, foram executados dois estudos de caso, por meio da instanciação de serviços distintos, baseados no paradigma de conhecimento como serviço utilizando a arquitetura H-KaaS. O primeiro destes foi a adaptação de uma plataforma de suporte à decisão clínica existente, que não foi elaborada originalmente como sistema KaaS. Já o segundo estudo de caso tratou da criação de um novo serviço para suporte à decisão clínica baseado na arquitetura de referência proposta. Em resumo, os dois estudos de caso executados foram:

- Adequação de uma plataforma de suporte à decisão clínica já existente, baseada em ontologias de domínio como principal sistema de representação do conhecimento e raciocínio;
- Desenvolvimento de um novo serviço baseado em mineração de dados médicos e aprendizagem de máquina, no domínio da oncologia, para o apoio à decisão na saúde de acordo com as especificações da H-KaaS.

A adaptação do sistema de suporte à decisão clínica baseado em ontologias existente permitiu a avaliação da arquitetura proposta, sendo possível efetuar a comparação dos resultados obtidos com a sua implementação original.

Por outro lado, a criação do serviço baseado em mineração de dados permitiu analisar o processo de concepção de novos serviços de conhecimento, já pensados segundo a arquitetura de referência proposta, evidenciando dificuldades ou facilidades resultantes de sua adoção nos estágios iniciais do desenvolvimento.

Os estudos de caso foram escolhidos de maneira a utilizar diferentes formas de representação de conhecimento e raciocínio em subdomínios distintos da saúde, de modo que cada serviço instanciasse subcomponentes diversos da arquitetura de referência H-KaaS, permitindo seu melhor entendimento e validação.

Nas seções seguintes, será apresentado em mais detalhes como foi executado cada estudo de caso, apresentando uma contextualização do problema, a metodologia utilizada para o desenvolvimento do serviço e os resultados obtidos.

5.1. ESTUDO DE CASO 1: SUPORTE À DECISÃO CLÍNICA BASEADO EM ONTOLOGIA NO DOMÍNIO DA NEFROLOGIA

5.1.1. CONTEXTUALIZAÇÃO

No domínio da saúde, muitas doenças podem ter seus índices de mortalidade significativamente reduzidos caso seja feito um diagnóstico precoce, com encaminhamento adequado do paciente ao especialista (ALENCAR; CIOSAK, 2015; RODRIGUES; CAMARGO, 2003). Dentre estas, destaca-se a doença renal crônica (DRC), considerada um dos principais problemas de saúde pública no mundo, tendo um alto índice de mortalidade tanto em países desenvolvidos quanto nos em desenvolvimento (MINISTÉRIO DA SAÚDE, 2014).

A doença renal crônica é definida como lesão do parênquima renal (com função renal normal) e/ou como a diminuição funcional renal por um período igual ou superior a três meses. 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).

Dessa forma, em sua fase mais avançada, devido à perda das funções renais de maneira progressiva e irreversível, os rins não conseguem mais manter a normalidade do meio interno do paciente (ROMÃO JUNIOR, 2004). Por consequência, a perda progressiva da função renal leva muitos pacientes a necessitar de algum tipo de terapia renal substitutiva e ao acompanhamento de um nefrologista, médico especialista no domínio da nefrologia, área da medicina cujo objetivo é o diagnóstico e tratamento clínico das doenças do sistema urinário (SOCIEDADE BRASILEIRA DE NEFROLOGIA, 2019).

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 (ROMÃO 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 (SOCIEDADE BRASILEIRA DE NEFROLOGIA, 2013). Dessa forma, o diagnóstico precoce e a prevenção da doença têm se tornado cada vez mais importantes para que sejam tomadas medidas preventivas a fim de retardar ou interromper o avanço da DRC (BASTOS; KIRSZTAJN, 2011).

Em vista disso, dada a problematização e a importância do encaminhamento correto e rápido de pacientes com suspeita de DRC, o domínio da nefrologia foi escolhido para servir como primeiro estudo de caso desta pesquisa.

Portanto, pretende-se com este estudo de caso adaptar um protótipo já existente na área da nefrologia à arquitetura H-KaaS, com o objetivo de fornecer o suporte à decisão clínica e compartilhar o conhecimento nele modelado, instanciando os componentes da arquitetura necessários, auxiliando profissionais da saúde no diagnóstico, estadiamento e encaminhamento de pacientes com DRC.

5.1.2. METODOLOGIA

Como o objetivo deste estudo de caso é a validação da arquitetura proposta por meio da adaptação de um protótipo já existente no domínio da nefrologia, foi escolhido um sistema de suporte à decisão clínica, baseado em uma ontologia de domínio, que não foi desenvolvido com foco no compartilhamento do conhecimento. Sendo assim, o serviço de suporte à decisão clínica escolhido para o desenvolvimento deste estudo de caso foi o OntoDecideDRC, proposto por Tavares (2016).

A ontologia OntoDecideDRC, modelo de conhecimento do serviço, tem como objetivo prover suporte à decisão clínica aos médicos especialistas da área da nefrologia e aos profissionais que trabalham na atenção primária da saúde. Portanto, o serviço da OntoDecideDRC baseia-se em uma ontologia no domínio da nefrologia, capaz de auxiliar médicos da atenção primária e nefrologistas quanto aos cuidados para o paciente com DRC. A ontologia foi desenvolvida utilizando protocolos de atendimento da nefrologia, sendo capaz de indicar a presença da DRC, o seu grau de risco e se o procedimento de encaminhamento do paciente ao especialista é necessário (TAVARES, 2016).

Após a escolha do modelo de conhecimento a ser usado como fonte de dados, com o objetivo de facilitar seu entendimento e de definir as tecnologias utilizadas durante a implementação do extrator de conhecimento e do servidor de conhecimento, a ontologia OntoDecideDRC, fonte de dado utilizada neste estudo de caso, foi classificada seguindo as sugestões da arquitetura H-KaaS. A Tabela 6 mostra as classificações da fonte de dados OntoDecideDRC baseadas nas categorias propostas.

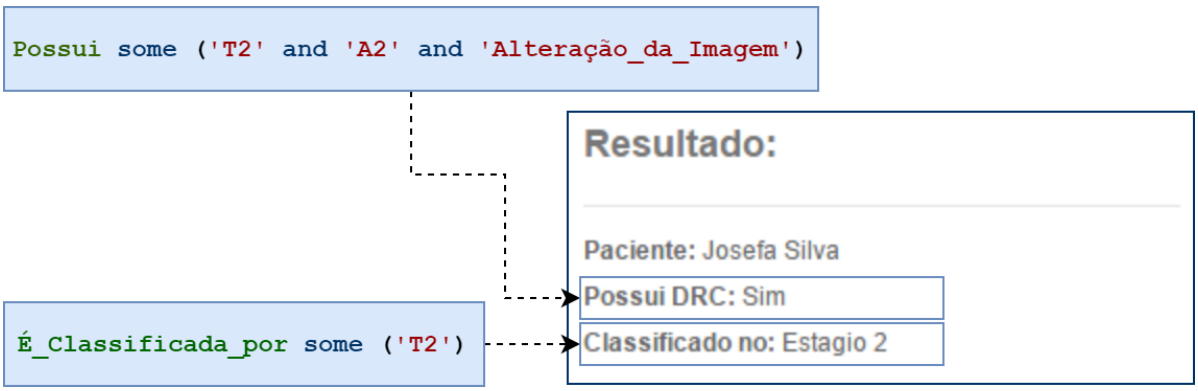
Tabela 6 - Classificações da OntoDecideDRC baseadas nas sugestões da H-KaaS

Categoria analisada	Classificação da fonte de dado
Localização	Interna; Local
Modelo de dado ou conhecimento	Modelo de conhecimento explícito do paradigma simbólico
Processamento de consultas	Síncrono
Comunicação	Unidirecional

Fonte: Elaborado pelo autor.

Dessa forma, para a utilização dessa fonte de dados pelo componente extrator de conhecimento, fez-se necessária a escrita de um aplicativo de linha de comando, capaz de executar o raciocinador HermiT 1.3.8.1 (SHEARER; MOTIK; HORROCKS, 2008) e inferir em uma ontologia do tipo OWL, utilizando consultas baseadas na lógica descritiva. Esse aplicativo foi implementado na linguagem de programação Java 1.8 que, utilizando a OWL API (HORRIDGE; BECHHOFER, 2011), tornou-se um dos principais componentes do extrator de conhecimento. A Figura 29 mostra dois exemplos de consultas realizadas e seus respectivos resultados.

Figura 29 - Consultas realizadas pelo extrator de conhecimento durante a execução do método de estadiamento da DRC



Fonte: Elaborado pelo autor.

Além disso, foram criadas, dentro do extrator de conhecimento, devido à forma como a ontologia foi projetada, algumas funções auxiliares baseadas em regras de domínio para a execução das consultas à ontologia. Uma dessas funções é a implementação do cálculo da taxa de filtração glomerular (TFG). Esse cálculo usa a fórmula *modification of diet in renal disease*

(MDRD) simplificada, especificada por Levey et al. (2000). A partir dele, e através das informações do paciente, chega-se a um valor numérico utilizado durante a realização das consultas à fonte de dados.

A implementação do cálculo da TFG, presente no componente extrator de conhecimento, pode ser vista na Figura 30. O método, escrito na linguagem PHP, recebe atributos do paciente como creatinina, sexo, cor da pele e idade e retorna o valor numérico aproximado da TFG. É importante notar que a metodologia de cálculo da TFG, classes e consultas OWL utilizadas para extração do conhecimento já haviam sido definidas por Tavares (2016) em seu protótipo original, sendo estas adaptadas para o funcionamento no novo serviço provedor de conhecimento desenvolvido.

Figura 30 - Função auxiliar para o cálculo da taxa de filtração glomerular presente no extrator de conhecimento

```php
/**
 * 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;
}
```

Fonte: Elaborado pelo autor.

O servidor de conhecimento foi implementado na linguagem de programação PHP 5.6 e, além de conter as funções necessárias para executar consultas à fonte de dados por meio do extrator de conhecimento, implementa métodos para conversão e manipulação de dados e conhecimento que serão utilizados pelo servidor da API de comunicação, seguindo sua especificação.

As tecnologias utilizadas para a implementação do servidor da API de comunicação foram baseadas nas sugestões da arquitetura de referência H-KaaS em conjunto com a análise dos dados coletados de trabalhos similares no domínio da saúde, apresentados no capítulo 3.

Dessa forma, o servidor da API de comunicação foi desenvolvido na linguagem de programação PHP 5.6, provendo uma API baseada na arquitetura REST, utilizando formato de dados serializados JavaScript Object Notation (JSON), que é um formato leve, baseado em texto, independente de linguagem, para a transferência de informações (CROCKFORD, 2006). Esse padrão define algumas regras de formatação que possibilitam a representação de dados de forma estruturada, facilmente entendidas pelo aplicativo consumidor de conhecimento.

Em relação ao desenvolvimento da documentação técnica da API de comunicação, foi utilizada a ferramenta Swagger Editor. Esta 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 semiautomática de páginas capazes de descrever de forma simplificada os aspectos de uma API. Além disso, é possível a execução de métodos, diretamente da página de documentação, criando uma maneira fácil para execução de testes durante o desenvolvimento de novos aplicativos (SWAGGER, 2018). As tecnologias utilizadas para a implementação dos componentes da arquitetura H-KaaS concreta utilizada neste estudo de caso podem ser vista na Tabela 7.

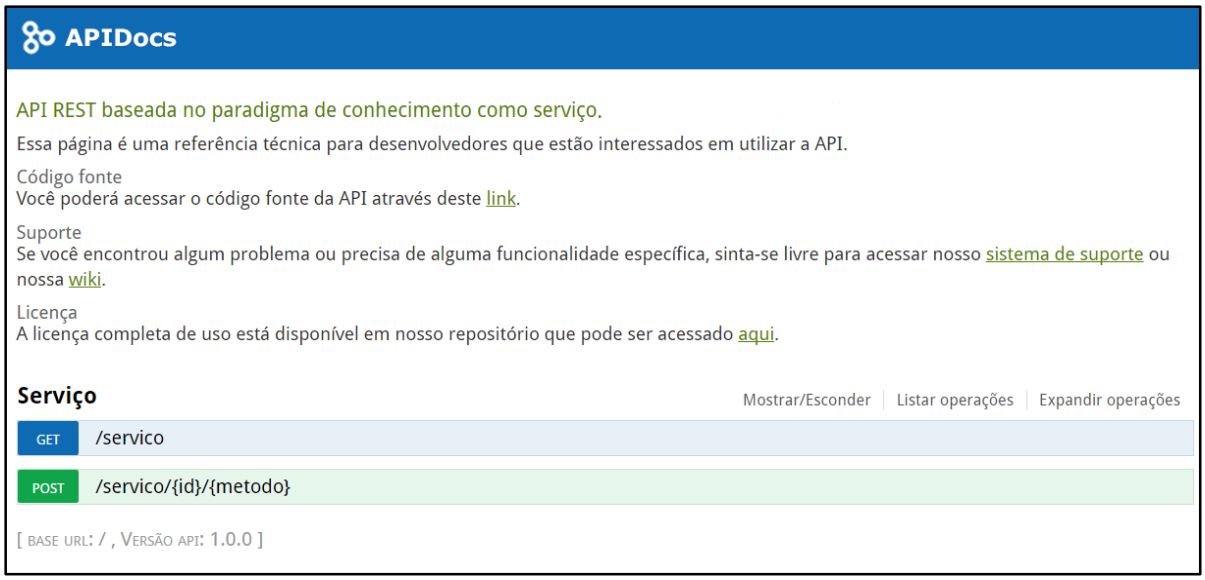
Tabela 7 - Tecnologias utilizadas na implementação dos componentes da arquitetura H-KaaS concreta

Componente	Tecnologia utilizada na implementação
Fonte de dados	Ontologia OWL; OntoDecideDRC
Extrator de conhecimento	Java 1.8
Regras de domínio	PHP 5.6
Raciocinador	HermiT 1.3.8.1; Java 1.8
Servidor de Conhecimento	PHP 5.6
Servidor da API de comunicação	PHP 5.6
API de comunicação	Rest; JSON; Swagger
Consumidor de conhecimento	PHP 5.6; Wordpress 4.9.15
Banco de dados do servidor de conhecimento	MySQL 5.7

Fonte: Elaborado pelo autor.

Como resultado da especificação da API de comunicação, foi possível o desenvolvimento de uma página dedicada para a descrição detalhada de suas funções, objetos e respostas, como pode ser visto na Figura 31.

Figura 31 - Página de documentação da API de comunicação com os consumidores de conhecimento



Fonte: Elaborado pelo autor.

De forma geral, durante a execução de uma consulta, o servidor da API de comunicação recebe requisições HTTP no formato POST e, por isso, para executá-la, o consumidor de conhecimento precisará fornecer informações serializadas baseadas em um formulário de entrada. A resposta desse comando é um objeto serializado em JSON, contendo informações sobre a consulta realizada e o resultado de sua execução.

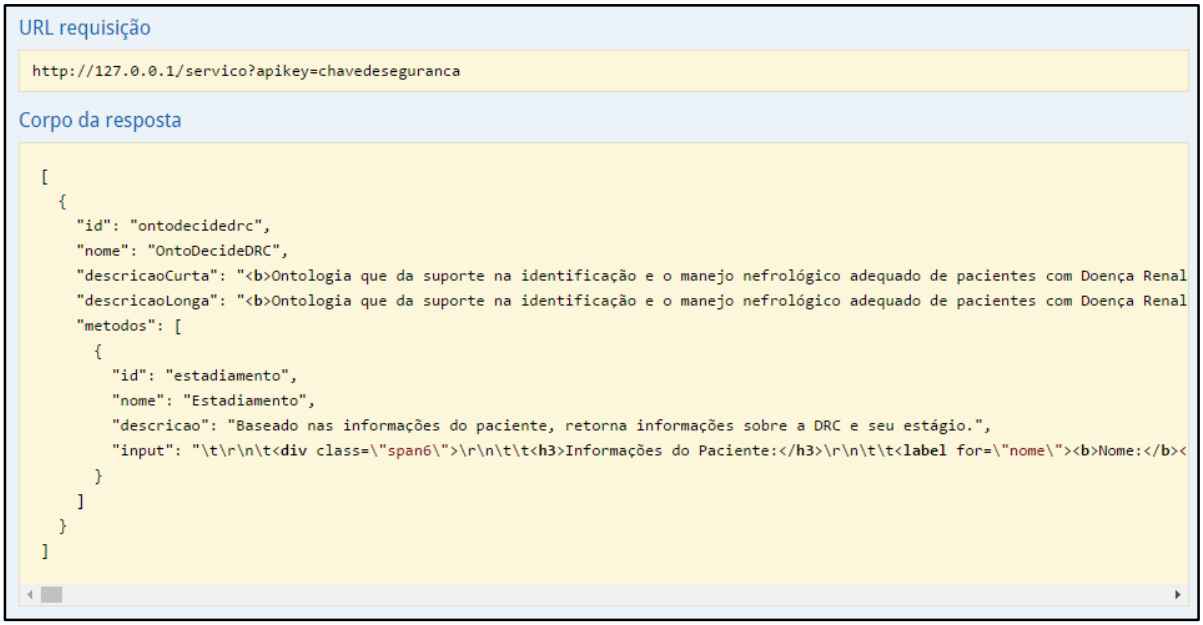
Os metadados referentes às consultas realizadas, como identificação do aplicativo consumidor, servidor de conhecimento escolhido, método chamado e tempo de execução, foram armazenados no banco de dados de estatísticas, utilizando um servidor MySQL 5.7, dentro do componente de monitoramento do servidor da API de comunicação.

Em relação ao tratamento e formato das mensagens de erro, a implementação do servidor da API de comunicação possui um objeto especial da classe erro, que é instanciado e retornado caso um problema fatal aconteça durante a execução da consulta. Alguns erros previstos pela API foram: 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 consulta, com o objetivo de identificar o aplicativo consumidor de conhecimento e garantir que o acesso ao conhecimento estaria limitado apenas a aplicações autorizadas, foi utilizado um mecanismo de chave compartilhada entre o aplicativo consumidor de conhecimento e o servidor da API de comunicação. O processo de autenticação e verificação das chaves foi feito com a ajuda do

banco de dados de autenticação, implementado utilizando MySQL 5.7, dentro do módulo de controle de acesso que, por sua vez, foi responsável por verificar a autenticidade de cada chave e consulta realizada. O resultado da execução de um dos comandos pode ser visto na Figura 32. O comando para listar serviços, por exemplo, retorna uma lista de serviços que, por sua vez, possui uma lista de métodos que podem ser executados.

Figura 32 - Exemplo de execução da listagem de serviços na API de comunicação



Fonte: Elaborado pelo autor.

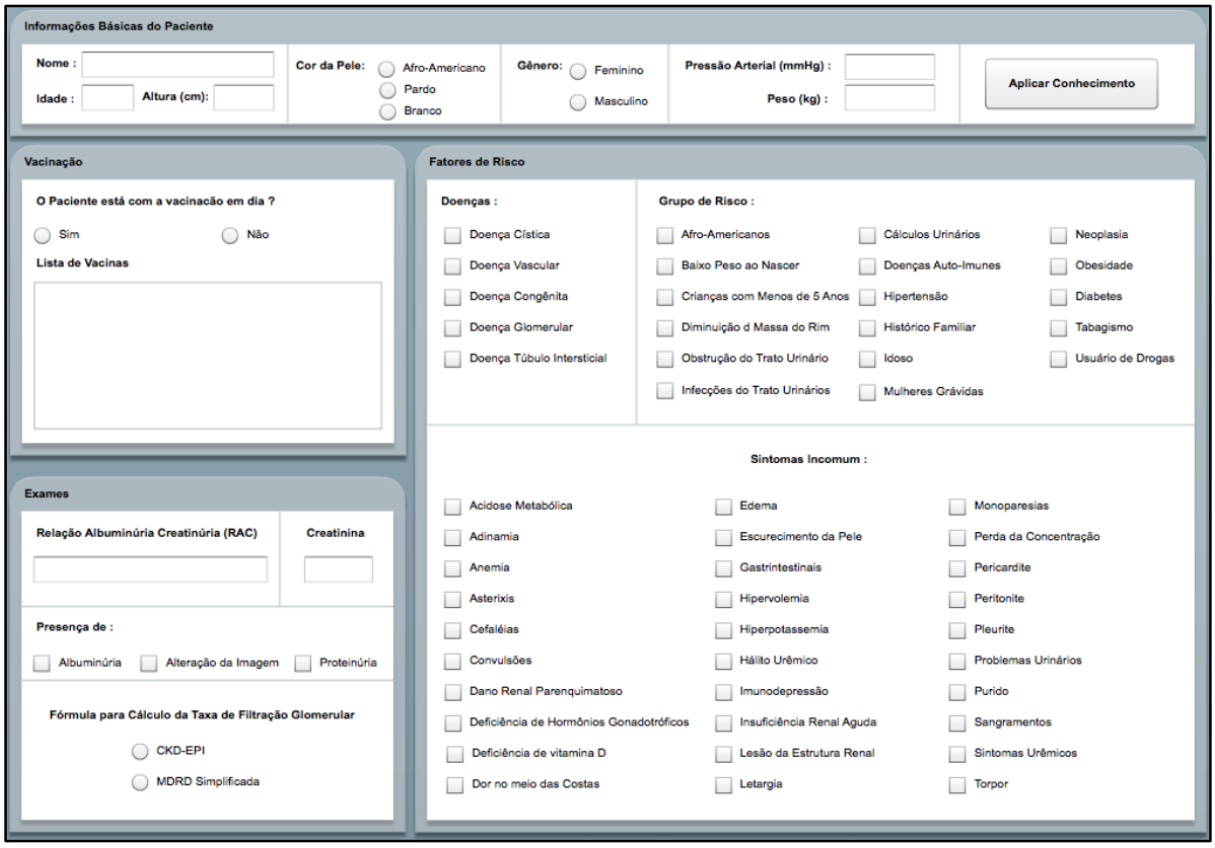
Por fim, o aplicativo consumidor de conhecimento teve como objetivo comprovar o funcionamento do serviço provedor de conhecimento e implementar a interface gráfica do protótipo escolhido.

A linguagem de programação utilizada para a implementação do website consumidor de conhecimento foi PHP 5.6 e, em conjunto com o *framework* Wordpress 4.9.15 e banco de dados MySQL 5.7, possibilitou a implementação de páginas como: login, registro, recuperar senha, listar serviços, executar método, contato, entre outras.

Ao adaptar o serviço OntoDecideDRC, percebeu-se que o código-fonte de sua interface gráfica não estava disponível e, por isso, foi necessária a criação de um novo formulário para a entrada de dados baseado na especificação original do protótipo. A Figura 33 mostra a tela original do protótipo para a entrada de dados. Esta serviu como base para o método de estadiamento do serviço OntoDecideDRC, capaz de diagnosticar o paciente quanto a presença da DRC, informando o grau de risco da doença e em qual estágio o paciente em questão se

encontra. Os campos do formulário foram reescritos e organizados de maneira similar à sua versão original.

Figura 33 - Tela original para entrada de dados no serviço OntoDecideDRC



Fonte: Tavares (2016).

Os resultados da implementação do formulário de estadiamento do serviço da OntoDecideDRC podem ser vistos na Figura 34. O formulário é gerado dinamicamente, a partir dos dados recebidos do servidor da API de comunicação. Quando submetido, uma requisição é enviada à API e, a partir dela, é criado o código HTML com a resposta que será mostrada ao usuário. Caso, durante o processamento ou envio da requisição, seja detectado um erro, uma mensagem amigável de erro e uma possível solução serão apresentadas ao usuário.

Figura 34 - Formulário para o estadiamento da DRC adaptado à H-KaaS

Fonte: Elaborado pelo autor.

Na seção seguinte, será apresentado como uma consulta ao serviço provedor de conhecimento, feita através do website consumidor de conhecimento, é respondida através do uso do servidor de conhecimento implementado.

5.1.3. EXECUÇÃO DAS CONSULTAS OWL E SUPORTE À DECISÃO CLÍNICA

Com o objetivo de demonstrar como as consultas à base de conhecimento, representadas por uma ontologia de domínio no formato OWL, são executadas dentro do serviço provedor de conhecimento, o fluxo de execução foi detalhado utilizando o caso clínico exemplificado a seguir:

> Um médico da atenção primária está atendendo uma mulher de 35 anos e cor de pele branca, apresentando uma dosagem de creatinina sanguínea de 0,9 mg/dL e alteração no exame de imagem. Foi identificada também, para esse paciente, uma relação entre albumina e creatinina de 35 mg/g (BARROS; GONÇALVES, 2009).

Inicialmente, o profissional da atenção primária preenche os dados na interface gráfica provida pelo protótipo, o que nesse caso é feito via uma página no navegador.

O consumidor de conhecimento, fazendo uso da API de comunicação, serializa e envia os dados do paciente para o serviço provedor de conhecimento, parte central da arquitetura proposta, que é responsável por processar e responder às consultas realizadas.

Depois que a requisição é validada e aceita pelo módulo de controle de acesso, os dados da consulta são encaminhados para o componente responsável pelo processamento da solicitação, o servidor de conhecimento. Nele, a partir dos dados do paciente enviados, são formuladas consultas que serão encaminhadas para o extrator de conhecimento apropriado, que, com a ajuda de um algoritmo raciocinador, executa consultas e inferências à ontologia por meio de consultas OWL.

Inicialmente, os dados como idade, cor da pele, sexo e creatinina são organizados e seus valores são, respectivamente: 35 anos, branca, mulher, 0.9 mg/dL. Estes dados são utilizados pelo extrator de conhecimento para calcular e discretizar a TFG usando a fórmula MDRD simplificada.

De acordo com o valor numérico calculado para a TFG, resultado da execução da função mdrd_simplificada detalhada na seção anterior, é selecionada a classe “T2”, que representa o intervalo de valores entre 60 e 89. Um processo similar ocorre no tocante à relação albuminúria/creatinúria do paciente, em que o valor de 35 mg/g é representado na ontologia pela classe “A2”, seguindo as instruções do protótipo original.

De acordo com o exemplo, o extrator de conhecimento seleciona as classes da ontologia que correspondem aos dados inseridos: “Alteração_da_Imagem”, “T2” e “A2”. Em seguida, o extrator de conhecimento produz a consulta OWL usando a propriedade “Possui”, que será executada no raciocinador HermiT com o objetivo de obter as informações necessárias para o auxílio ao diagnóstico do paciente. A consulta, escrita no formato DL, pode ser vista na Figura 35.

Figura 35 - Consulta realizada para determinar a presença de DRC

```
Possui some 'T2' and 'A2' and 'Alteração_da_Imagem'
```

Fonte: Elaborado pelo autor.

O método de inferência retorna a classe DRC, que indica que o paciente em questão possui doença renal crônica.

Em seguida, para obter o grau de risco do paciente, utilizando a classe DRC que foi retornada e os dados da consulta anterior, outra consulta é gerada e executada, como pode ser

visto na Figura 36. Para o paciente em questão, a consulta retorna a classe “Risco Moderadamente Aumentado”, que indica o grau de risco do paciente.

Figura 36 - Consulta realizada para determinar o grau de risco do paciente

```
(('DRC') and (Possui some 'T2' and 'A2' and
'Alteração_da_Imagem'))
```

Fonte: Elaborado pelo autor.

Em relação ao estadiamento da DRC, sabe-se que este é classificado de acordo com o valor da TFG, calculado previamente. Sendo assim, o extrator de conhecimento seleciona a classe “T2” e desenvolve a consulta, que pode ser vista na Figura 37, utilizando a propriedade “É_Classificada_por”, obtendo o resultado “Estágio_2”.

Figura 37 - Consulta realizada para determinar o estadiamento da DRC

```
É_Classificada_por some ('T2')
```

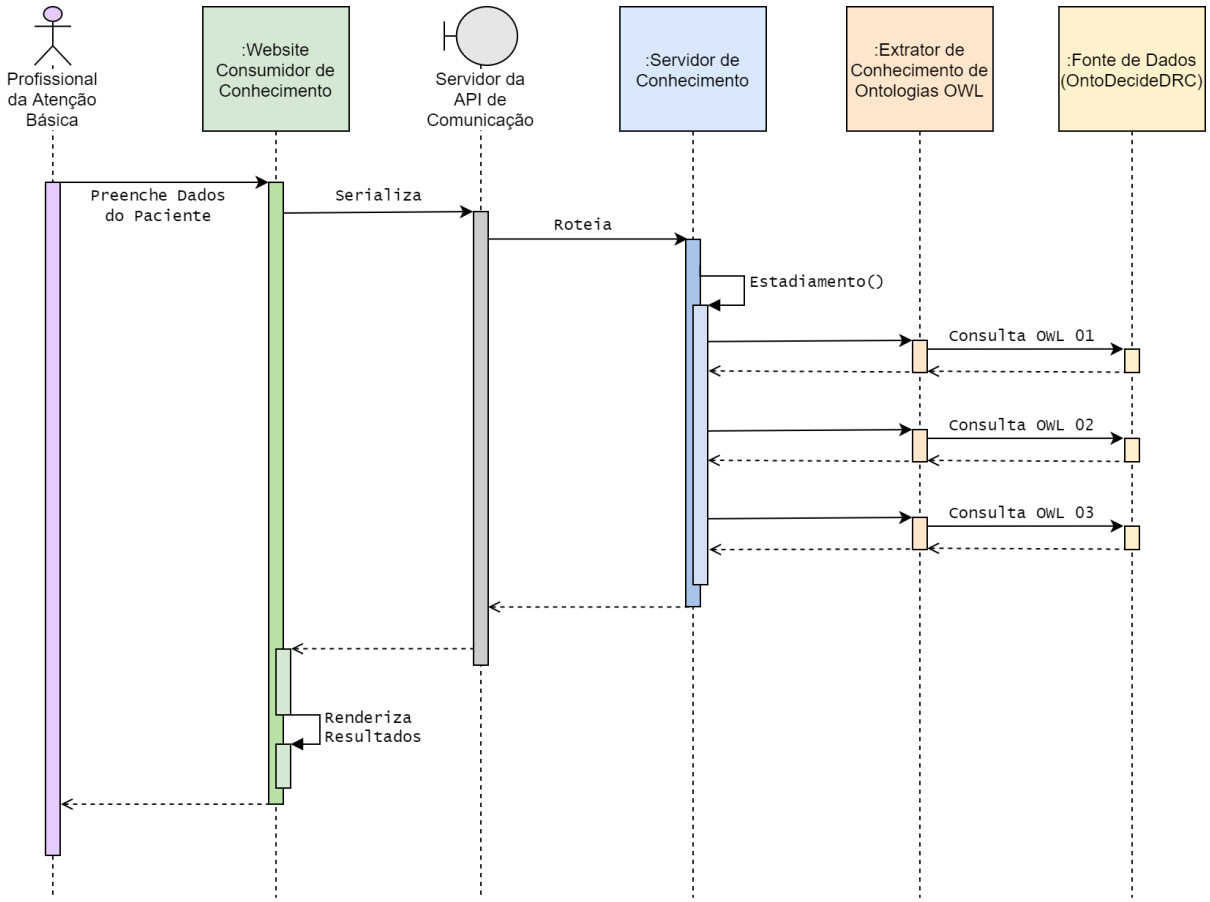
Fonte: Elaborado pelo autor.

Em resumo, para o caso clínico em questão, o extrator de conhecimento de ontologias, com base nos dados informados e nos resultados das consultas realizadas, obteve a confirmação de que o paciente possui DRC com risco moderadamente aumentado, estando no estágio 2 da doença. Além do exemplo citado, a ontologia OntoDecideDRC também permite consultas relativas ao encaminhamento de pacientes ao especialista e informações sobre o manejo clínico do paciente.

Para finalizar, os resultados das consultas executadas são processados pelo servidor de conhecimento e então serializados, através do servidor da API de comunicação, ao aplicativo consumidor de conhecimento, isto é, o website responsável por renderizar e apresentar ao profissional da atenção primária os resultados do processamento.

A Figura 38, por meio de um diagrama de sequência especificado no padrão UML, mostra um resumo do processo de extração e distribuição do conhecimento no sistema desenvolvido. Nele, é apresentado o fluxo de requisições enviadas através dos diferentes componentes da H-KaaS, a fim de exemplificar sua utilização.

Figura 38 - Fluxo de conhecimento no serviço de conhecimento desenvolvido utilizando a arquitetura H-KaaS



Fonte: Elaborado pelo autor.

5.1.4. RESULTADOS

O principal resultado da execução deste estudo de caso foi a validação da arquitetura de referência H-KaaS, através da implementação de uma instância por meio da adaptação de um serviço de suporte à decisão clínica no domínio da nefrologia baseado em ontologias OWL.

A implementação do consumidor de conhecimento possibilitou ao usuário da plataforma realizar consultas por meio de formulários similares aos providos originalmente pelo serviço OntoDecideDRC, protótipo escolhido para ser adaptado à H-KaaS neste estudo de caso.

Como fonte de dados, adaptou-se uma ontologia no domínio da nefrologia, que, com a ajuda de um extrator de conhecimento e um algoritmo raciocinador, permitiu a inferência no conhecimento através de um servidor de conhecimento e a criação do mecanismo de suporte à decisão clínica, similar ao protótipo original.

O extrator de conhecimento desenvolvido, embora tenha sido pensado em função da OntoDecideDRC, foi criado de maneira genérica, possibilitando a adaptação futura a outras ontologias de domínio disponibilizadas em formato OWL e compatíveis com o raciocinador HermiT.

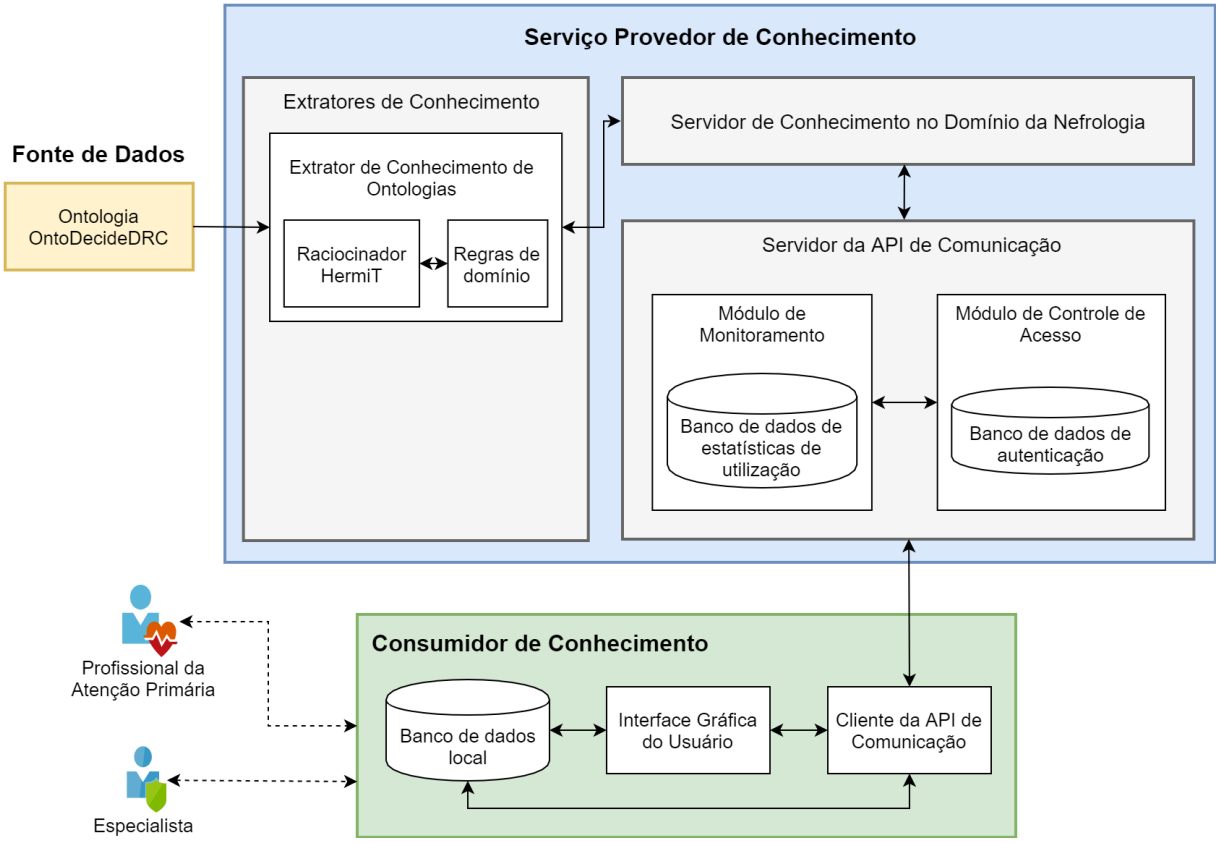
Além disso, durante a realização deste estudo de caso, foi especificado um protocolo de comunicação, usado entre o aplicativo consumidor e o serviço provedor de conhecimento, que permitiu que fossem enviadas consultas através do protocolo HTTP e por meio da serialização de entidades no formato JSON. Esse mesmo protocolo pode ser reutilizado para implementação de outros serviços de conhecimento, como será demonstrado no segundo estudo de caso desta pesquisa.

Por fim, foi feita uma análise de como o fluxo de conhecimento e consultas ocorrem dentro do serviço provedor de conhecimento, com o objetivo de esclarecer, validar e reforçar a descrição da arquitetura H-KaaS.

É importante notar que, devido ao fato de a H-KaaS ser uma arquitetura de referência, um modelo generalista de sistemas similares de certo domínio, nem todos os componentes precisaram ser instanciados, porém, como resultado deste estudo de caso, o sistema de conhecimento adaptado foi capaz de oferecer o suporte à decisão clínica através de seu serviço provedor de conhecimento, mostrando a efetividade da arquitetura. A Figura 39 mostra os componentes da H-KaaS instanciados por este estudo de caso.

Com relação às dificuldades para o desenvolvimento deste serviço de conhecimento, 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 a execução das consultas, especificadas pelo serviço original, fazendo uso da OWL API e da linguagem de programação Java. Além disso, a interface gráfica do protótipo não estava mais disponível e, portanto, foi observada a necessidade do desenvolvimento de novos formulários e páginas para a visualização dos resultados no aplicativo consumidor de conhecimento.

Figura 39 - Componentes instanciados da arquitetura de referência H-KaaS durante a implementação do primeiro estudo de caso



Fonte: Elaborado pelo autor.

O desenvolvimento deste estudo de caso foi supervisionado por um especialista no domínio da nefrologia que, através de consultas feitas utilizando o aplicativo consumidor de conhecimento, pôde validar os resultados obtidos, comprovando que estavam de acordo com a especificação do modelo de conhecimento original.

Em conclusão, este estudo de caso mostra a viabilidade da arquitetura H-KaaS para o compartilhamento de conhecimento na área da saúde, mais especificamente na nefrologia. Além disso, o estudo de caso demonstra a capacidade de a arquitetura lidar com fontes de dados com base em ontologias de domínio, que são bastante utilizadas em sistemas de suporte à decisão clínica.

5.2. ESTUDO DE CASO 2: SISTEMA DE SUPORTE À DECISÃO PARA PREDIÇÃO DE RESULTADOS DE EXAMES RELACIONADOS AO CÂNCER CERVICAL

5.2.1. CONTEXTUALIZAÇÃO

No domínio da oncologia, o câncer cervical, também conhecido como câncer de colo de útero, é a terceira maior causa de câncer em mulheres no mundo. Estima-se que ocorreram 530.000 casos em 2008, sendo registradas 275.000 mortes pela doença, perdendo apenas para o câncer de mama e o colorretal (ARBYN et al., 2011).

Alguns fatores estão fortemente ligados ao acometimento do câncer cervical, como: início precoce da atividade sexual, tabagismo, uso de contraceptivos orais, carência de vitaminas, múltiplos parceiros sexuais e, principalmente, infecções persistentes pelo vírus HPV (INSTITUTO NACIONAL DE CÂNCER, 2016). No Brasil, estima-se que em 2018 foram registrados 16.370 novos casos de câncer cervical, com um risco estimado de 15,43 casos a cada 100 mil mulheres (INSTITUTO NACIONAL DE CÂNCER, 2017).

A taxa de mortalidade por câncer cervical pode ser reduzida caso seja feito o diagnóstico precoce da doença (WORLD HEALTH ORGANIZATION, 2007). A identificação precoce do câncer cervical pode ser feita por meio do rastreamento de lesões precursoras, que podem ser detectadas e tratadas, impedindo o progresso da doença. Sendo assim, o diagnóstico rápido e o encaminhamento preciso de pacientes é uma das principais formas de garantir um tratamento efetivo contra o câncer cervical (INSTITUTO NACIONAL DE CÂNCER, 2018).

A inteligência artificial, em conjunto com o uso de sistemas de suporte à decisão clínica, tem ajudado a oncologia no desenvolvimento de algoritmos e técnicas para o diagnóstico e estadiamento de doenças (GYAWALI, 2018; LOBO, 2017).

Nesse contexto, destaca-se o uso de redes neurais artificiais, modelos computacionais inspirados pela biologia que podem ser utilizados em uma grande variedade de problemas de aprendizado de máquina (SCHMIDHUBER, 2015). Estas, segundo Huang, Zhu e Siew (2006), possuem a capacidade de aproximar mapeamentos não lineares complexos através de um conjunto de dados de entrada e, com isso, são capazes de encontrar soluções que normalmente são difíceis de modelar utilizando abordagens clássicas.

Pretende-se, assim, desenvolver um sistema de suporte à decisão clínica baseado em aprendizagem de máquina através do uso de redes neurais artificiais, integrado à arquitetura de software H-KaaS concreta apresentada no estudo de caso anterior, que poderá ser usada na área

oncológica visando auxiliar o diagnóstico apropriado de pacientes com suspeita de câncer cervical.

Além disso, o presente estudo de caso visa a demonstrar a utilização de múltiplos servidores de conhecimento dentro de um mesmo serviço provedor de conhecimento, evidenciando a possibilidade de reuso de componentes e comprovando a flexibilidade da arquitetura H-KaaS, mesmo quando instanciada em diferentes subdomínios da saúde.

5.2.2. METODOLOGIA

Durante o presente estudo de caso, será detalhado o desenvolvimento de um novo sistema de suporte à decisão clínica no domínio da oncologia, baseado na arquitetura de referência H-KaaS, que, através do uso de um extrator de conhecimento baseado em aprendizagem de máquina e mineração de dados, foi capaz de responder a consultas relacionadas à predição de resultados de biópsias de pacientes com suspeita de câncer cervical.

Como primeiro passo da elaboração do serviço, foi desenvolvida uma base de conhecimento que serviu como fonte de dados para o sistema de suporte à decisão clínica projetado. Em seguida, adaptou-se esta fonte de dados à arquitetura de referência H-KaaS, disponibilizando-a através de um novo serviço de conhecimento, que reutilizou diversos componentes já desenvolvidos, como o servidor da API de comunicação e aplicativo consumidor de conhecimento.

Com a finalidade de extrair, de forma inicial, o conhecimento de um conjunto de dados, foi feito o treinamento de uma rede neural artificial que, de tempos em tempos, poderá ser refinada pelo engenheiro de conhecimento a partir de dados coletados de novos pacientes inseridos pelos usuários da plataforma. Sendo assim, na seção 5.2.2.1 é detalhada a metodologia empregada na construção do modelo de conhecimento utilizado como fonte de dados.

Em seguida, a partir dos resultados obtidos durante a criação da fonte de dados, após o treinamento da rede neural artificial, a seção 5.2.2.2 apresenta a construção do extrator de conhecimento, capaz de executar inferências nessa fonte de dados, e um servidor de conhecimento, responsável por prover o suporte à decisão clínica e implementar meios para armazenar dados de consultas e resultados de exames que, por sua vez, poderão ser utilizados para refinamento do modelo de conhecimento compartilhado.

5.2.2.1. DESENVOLVIMENTO DO MODELO DE CONHECIMENTO

Com a finalidade de oferecer suporte à decisão clínica, o treinamento da rede neural artificial foi executado utilizando aprendizagem de máquina supervisionada (RUSSELL; NORVIG, 2013), a partir dos dados de fatores de risco do câncer cervical coletados por Fernandes, Cardoso e Fernandes (2017) e distribuídos pela Universidade da Califórnia. O banco de dados empregado consiste nas informações demográficas, hábitos e histórico médico anonimizados de 858 pacientes, juntamente com suas respostas às perguntas de um questionário sobre fatores de risco, podendo algumas destas não ter sido respondidas pelas pacientes por razões de privacidade, que tiveram de ser tratados antes da execução das tarefas de mineração de dados.

Para projetar e treinar a RNA que será utilizada para previsão de resultados de exames em pacientes com suspeita de câncer cervical, foi utilizado o *framework* Keras (CHOLLET, 2015), uma biblioteca de código aberto para aprendizagem de máquina escrita na linguagem de programação Python (ROSSUM; DRAKE, 2010), facilitando a criação e instância de redes neurais artificiais para tarefas de classificação e regressão.

Dentre os fatores de risco presentes no conjunto de dados, podemos destacar: número de parceiros sexuais, número de gravidezes, uso de contraceptivos hormonais, uso de dispositivo intrauterino, histórico de doenças sexualmente transmissíveis (DST) e diagnósticos prévios de câncer cervical.

A Tabela 8 lista os atributos presentes no conjunto de dados, valores máximos e mínimos, suas médias e o número de instâncias válidas para aquele atributo antes de qualquer pré-processamento destes dados.

Tabela 8 - Análise dos dados dos atributos do conjunto de dados utilizado

Atributo	Válidos	Méd.	Min.	Máx.	Atributo	Válidos	Méd.	Min.	Máx.
Age	858	26,8	13	84	STDs: pelvic inflammatory disease	753	0	0	1
Number of sexual partners	832	2,5	1	28	STDs: genital herpes	753	0	0	1
First sexual intercourse	851	17	10	32	STDs: molluscum contagiosum	753	0	0	1
Number of pregnancies	802	2,3	0	11	STDs: AIDS	753	0	0	0
Smokes	845	0,1	0	1	STDs: HIV	753	0	0	1
Smokes (years)	845	1,2	0	37	STDs: Hepatitis B	753	0	0	1
Smokes (packs/year)	845	0,5	0	37	STDs: HPV	753	0	0	1
Hormonal contraceptives	750	0,6	0	1	STDs: Number of diagnoses	858	0,1	0	3
Hormonal contraceptives (years)	750	2,3	0	30	STDs: Time since first diagnosis	71	6,1	1	22
IUD	741	0,1	0	1	STDs: Time since last diagnosis	71	5,8	1	22
IUD (years)	741	0,5	0	19	Dx: Cancer	858	0	0	1
STDs	753	0,1	0	1	Dx: CIN	858	0	0	1
STDs (number)	753	0,2	0	4	Dx: HPV	858	0	0	1
STDs: condylomatosis	753	0,1	0	1	Dx	858	0	0	1
STDs: cervical condylomatosis	753	0	0	0	Hinselmann	858	0	0	1
STDs: vaginal condylomatosis	753	0	0	1	Schiller	858	0,1	0	1
STDs: vulvo-perineal condylomatosis	753	0,1	0	1	Cytology	858	0,1	0	1
STDs: syphilis	753	0	0	1	Biopsy	858	0,1	0	1

Fonte: Elaborado pelo autor.

Para lidar com os atributos com muitas respostas faltantes, foi utilizada uma técnica tradicional, proposta por García-Laencina, Sancho-Gómez e Figueiras-Vidal (2010), que consiste na remoção de colunas sem quantidades significativas de respostas, partindo do pressuposto de que possuem baixa influência na classificação, ou, caso contrário, evitando que esses atributos influenciem negativamente na classificação dos pacientes que não responderam a tal pergunta. Dessa forma, optou-se por descartar os atributos "STDs: Time since first diagnosis" e "STDs: Time since last diagnosis".

Feito isso, com o objetivo de lidar com os dados faltantes em instâncias específicas, foram excluídas as linhas que possuíam pelo menos um valor de atributo desconhecido. O conjunto de dados final utilizado ficou com 668 instâncias e 34 atributos, sendo o campo "Biopsy" utilizado como supervisão.

O *framework* Keras, em sua versão 2.3.1, requer uma biblioteca auxiliar na comunicação e organização das estruturas de dados no computador. Para isso, neste estudo de caso, foi utilizada a biblioteca Tensorflow 1.13.1 (ABADI et al., 2016), criada pela equipe do Google Brain em 2015, com o objetivo de ajudar o desenvolvimento de sistemas baseado em aprendizagem de máquina supervisionada. Além disso, foram empregadas ferramentas como o Tensorboard, utilizado neste trabalho em sua versão 1.13.1, para a geração e acompanhamento dos gráficos em tempo real referentes aos erros de treinamento e teste da rede neural artificial durante seu desenvolvimento (GOLDSBOROUGH, 2016).

Com a finalidade de automatizar o treinamento e teste de diversas arquiteturas de redes, usou-se a biblioteca Scikit-learn (PEDREGOSA et al., 2012) em sua versão 0.21.3, permitindo que um número maior de arquiteturas de redes fosse treinado e testado paralelamente, o que reduziu significativamente o tempo de execução de seu treinamento.

Para a execução e testes das redes propostas, foi utilizado um servidor com as seguintes configurações: processador Intel Core i7-6700, com 64 GB DDR4 de memória RAM, GPU GeForce GTX 1080 e dois discos rígidos SSD de 500 GB SATA 6 Gb/s. As redes neurais foram instanciadas e treinadas com ajuda da placa gráfica.

A normalização e a escala dos dados foram feitas subtraindo o valor mediano dos dados entre o primeiro e o terceiro quantil para cada atributo do conjunto de dados, forçando os seus valores a ficarem próximos a zero (JAYALAKSHMI; SANTHAKUMARAN, 2011). Além disso, o cálculo da mediana no intervalo interquantil reduz o impacto negativo que valores atípicos teriam sobre os dados (BUITINCK; LOUPPE; BLONDEL; PEDREGOSA; MUELLER; GRISEL; NICULAE; PRETTENHOFER; GRAMFORT; GROBLER; LAYTON;

VANDERPLAS; JOLY; HOLT, 2013). Os valores medianos, encontrados durante a etapa de pré-processamento, foram serializados para um arquivo de texto, a fim de utilizá-los como parâmetros durante a execução de consultas à RNA em tempo de execução.

Redes neurais artificiais com muitos parâmetros têm tendência a possuir relações correlacionais confusas, produzindo um alto nível de ruído que pode influenciar em sua capacidade posterior de generalização. Tal problema é conhecido como superadaptação (do inglês, *overfitting*), que faz com que uma rede neural artificial já treinada não tenha uma boa capacidade em classificar corretamente novas instâncias que não estão contidas no conjunto de dados de treinamento (SRIVASTAVA et al., 2014).

Para evitar a superadaptação foi empregada a técnica de *dropout*, detalhada por Srivastava et al. (2014), que consiste na eliminação aleatória de alguns dos neurônios e suas conexões durante as épocas de treinamento da rede. O valor de *dropout* varia de 0 a 1 e consiste na taxa de neurônios eliminados aleatoriamente a cada rodada de aprendizado. Dessa forma, foram escolhidos dois valores possíveis de *dropout* a serem avaliados: 0 e 0,4.

Como funções de ativação das camadas ocultas das arquiteturas, foram escolhidas duas opções bastante utilizadas, sendo elas: ELU (do inglês, *exponential linear unit*) e sigmoide. A função ELU, uma melhoria em relação à função RELU (do inglês, *rectified linear unit*), é indicada para evitar uma eventual impossibilidade no treinamento das redes causada por valores nulos dos pesos devido a ajustes negativos constantes (DENG; WANG; WANG, 2016). A função sigmoide foi escolhida principalmente como função não linear complementar alternativa, devido a seu amplo uso no treinamento de redes neurais artificiais (KARLIK; OLGAC, 2011).

A métrica utilizada para a escolha da melhor arquitetura e seus hiperparâmetros foi a acurácia, pois ela fornece uma visão geral da performance do modelo, levando em consideração quantas pacientes foram classificadas corretamente dentre todas as instâncias.

Com a finalidade de encontrar os melhores hiperparâmetros de configuração da RNA, foram treinadas 12 configurações de arquiteturas densas diferentes, variando-se o número de neurônios por camada, o valor de *dropout* e a função de ativação. Tais arquiteturas foram geradas por meio de uma técnica conhecida como *grid-search*, que facilita a criação, treinamento e teste de diversas arquiteturas de redes por meio de intervalos de parâmetros predefinidos (BUITINCK; LOUPPE; BLONDEL; PEDREGOSA; MUELLER; GRISEL; NICULAE; PRETTENHOFER; GRAMFORT; GROBLER; LAYTON; VANDERPLAS; JOLY; HOLT; et al., 2013). A Tabela 9 mostra os parâmetros utilizados pela técnica de *grid-search* para a criação das arquiteturas.

Tabela 9 - Hiperparâmetros utilizados para a criação das arquiteturas das RNAs

Parâmetro	Possíveis Valores
Camadas	1; 128-1; 128-128-1
Função de ativação	ELU; Sigmoide
Dropout	0; 0.4
Métricas adicionais	Acurácia
Função Loss	Entropia binária cruzada
Otimizador	Adam

Fonte: Elaborado pelo autor.

O treinamento das RNAs foi realizado em 300 épocas, ou seja, apresentando o conjunto de treinamento para cada uma delas de maneira completa 300 vezes. Já o tamanho dos *minibatches* empregados (isto é, quantos exemplos de treinamento eram exibidos por vez à rede antes do ajuste dos pesos) foi de 32.

Para validar o modelo proposto, utilizou-se a técnica de validação cruzada estratificada com múltiplas dobras. A validação cruzada é uma técnica para avaliar a capacidade de generalização de um modelo a partir de um conjunto de dados. Essa técnica é amplamente empregada em problemas em que o objetivo da modelagem é a predição. Busca-se então estimar o quão preciso é este modelo na prática, ou seja, o seu desempenho para um novo conjunto de dados nunca antes visto pela rede (BAKER; ISOTANI; CARVALHO, 2011).

Durante a execução da técnica de validação cruzada foram empregadas 10 dobras, seguindo as recomendações propostas por Kuhn e Johnson (2013), que aconselham a utilização deste valor com o objetivo de reduzir possíveis vieses nos resultados obtidos. Dessa forma, cada uma das arquiteturas, instanciadas pelo algoritmo de *grid-search*, foi treinada e testada 10 vezes, permitindo o cálculo da acurácia média e seu desvio padrão no conjunto de dados de treinamento e teste. O conjunto de dados foi dividido em conjunto de treinamento e conjunto de testes de maneira aleatória, durante a execução da validação cruzada, por meio da utilização das classes StratifiedKFold e GridSearchCV da biblioteca Scikit-learn 0.21.3, utilizando seus valores padrão.

A Tabela 10 mostra um comparativo dos resultados obtidos a partir do treinamento de cada arquitetura de RNA proposta utilizando o conjunto de dados de treinamento e teste, além

do desvio padrão obtido pela técnica de validação cruzada. O tempo total de treinamento e teste para o conjunto de RNAs, em todas as dobras, foi de 2 horas, 22 minutos e 6 segundos.

Tabela 10 - Arquitetura e acurácia média apresentada por cada RNA ao fim do treinamento

Arquitetura	Função de ativação	*Dropout*	Acurácia	
			Treinamento	Teste
1	ELU	0	94,84% ± 3,02%	93,71% ± 4,11%
128-1	ELU	0	99,09% ± 0,45%	94,01% ± 5,20%
128-128-1	ELU	0	99,88% ± 0,21%	93,11% ± 5,57%
1	ELU	0,4	94,89% ± 1,46%	94,16% ± 4,44%
128-1	ELU	0,4	98,57% ± 0,45%	95,06% ± 4,02%
128-128-1	ELU	0,4	99,00% ± 0,51%	93,86% ± 4,31%
1	Sigmoide	0	95,81% ± 0,58%	94,46% ± 3,70%
128-1	Sigmoide	0	98,25% ± 0,70%	94,76% ± 4,45%
128-128-1	Sigmoide	0	98,65% ± 0,60%	94,61% ± 3,86%
1	Sigmoide	0,4	95,77% ± 0,46%	94,31% ± 3,39%
128-1	Sigmoide	0,4	97,54% ± 0,79%	95,06% ± 4,23%
128-128-1	Sigmoide	0,4	97,89% ± 0,80%	94,31% ± 5,13%

Fonte: Elaborado pelo autor.

Desse modo, em vista dos resultados obtidos durante a fase de treinamento das possíveis arquiteturas de redes neurais a serem utilizadas, a arquitetura de rede neural artificial que obteve uma melhor acurácia no banco de dados de testes foi escolhida para ser o modelo de conhecimento para servir de fonte de dados neste estudo de caso. Esta, por sua vez, possui uma camada oculta, com 128 neurônios, e uma camada de saída com um único neurônio. Além disso, utiliza o valor de *dropout* de 0.4 entre as camadas e função de ativação ELU nas camadas ocultas.

A RNA escolhida apresentou 95,06% de acurácia média no conjunto de dados de teste, indicando uma boa capacidade de generalização, e 98,57% nos dados de treinamento. Portanto, de forma a permitir a execução de consultas pelo extrator de conhecimento, sua arquitetura e pesos foram serializados em um arquivo de texto no formato JSON, que descreve a arquitetura da rede, e um arquivo com os pesos das ligações entre os neurônios no formato HDF5 (FOLK et al., 2011), um formato de arquivo otimizado para armazenamento de grande quantidade de dados numéricos.

Na seção seguinte será apresentada a metodologia aplicada para a adaptação da rede neural artificial desenvolvida à arquitetura H-KaaS, descrevendo como o extrator de conhecimento foi projetado, além dos detalhes de implementação do serviço de conhecimento e do banco de dados de consultas.

5.2.2.2. ADAPTAÇÃO À ARQUITETURA H-KAAS

Após o desenvolvimento do modelo de conhecimento que será utilizado como fonte de dados para implementação do novo serviço de conhecimento, com o propósito de facilitar o seu entendimento e de definir as tecnologias empregadas durante a implementação do extrator e servidor de conhecimento, a fonte de dado neste estudo de caso foi classificada seguindo as sugestões da arquitetura H-KaaS. A Tabela 11 mostra as classificações da fonte de dados baseadas nas categorias propostas pela H-KaaS.

Tabela 11 - Classificações da fonte de dados utilizada neste estudo de caso

Categoria analisada	Classificação da fonte de dado
Localização	Interna; Local
Modelo de dado ou conhecimento	Modelo de conhecimento implícito do paradigma estatístico-probabilístico
Processamento de consultas	Síncrono
Comunicação	Unidirecional

Fonte: Elaborado pelo autor.

Dessa forma, o extrator de conhecimento, desenvolvido na linguagem de programação Python 3.7.4, foi implementado de maneira a receber dados de requisições serializadas no formato JSON, via linha de comando, e a instanciar, através da biblioteca Keras 2.3.1 com o *backend* Tensorflow 1.13.1, a fonte de dados em questão.

Portanto, os dados do paciente são recebidos pelo extrator de conhecimento de maneira que cada chave do objeto corresponde a um atributo da instância a ser utilizada. Esses dados são então pré-processados e empregados nas consultas à rede neural artificial, o modelo de conhecimento previamente instanciado.

Os atributos da instância a ser executada (como idade, número de parceiros sexuais, entre outros) são normalizados e colocados numa mesma escala, por meio do mesmo algoritmo de pré-processamento empregado durante o treinamento do modelo de conhecimento. Em situações em que o paciente se recusou a responder a certa questão do formulário, os dados

faltantes foram substituídos pela mediana do atributo em questão, previamente armazenada durante o treinamento do modelo de conhecimento.

Após a execução da consulta feita ao extrator, o conhecimento extraído, ou seja, o resultado da previsão e seu grau de confiança, é encaminhado ao servidor de conhecimento para ser armazenado e, posteriormente, enviado como resposta à requisição do consumidor de conhecimento.

Em seguida, após o desenvolvimento do extrator de conhecimento, o servidor de conhecimento foi implementado de maneira que fosse capaz de se comunicar com o extrator e, ao mesmo tempo, possuir um banco de dados interno para armazenamento das consultas e dados de pacientes. Dessa forma, o servidor de conhecimento, chamado OncoService, foi implementado na linguagem de programação PHP 5.6, valendo-se de uma instância do banco de dados MySQL 5.7 como banco local.

Nesse contexto, o servidor de conhecimento permitiu, através do uso do servidor da API de comunicação implementado no estudo de caso anterior, a disponibilização de dois métodos para os aplicativos consumidores de conhecimento:

- **Predizer resultados:** Utiliza um modelo de conhecimento baseado em mineração de dados e aprendizagem de máquina para predizer os resultados de uma biópsia a ser solicitada a uma paciente com suspeita de câncer cervical. Além disso, esse método armazena os dados da paciente em um banco de dados local, a fim de serem usados posteriormente para o refinamento do modelo de conhecimento presente na fonte de dados;
- **Informar biópsia:** Armazena informações sobre um exame de biópsia já executado em uma paciente, associando este resultado a predições feitas anteriormente.

A Figura 40 mostra, no contexto do aplicativo consumidor de conhecimento, os métodos disponibilizados pelo OncoService através de seu serviço de conhecimento.

Figura 40 - Métodos disponibilizados pelo OncoService aos usuários do aplicativo consumidor de conhecimento

Fonte: Elaborado pelo autor.

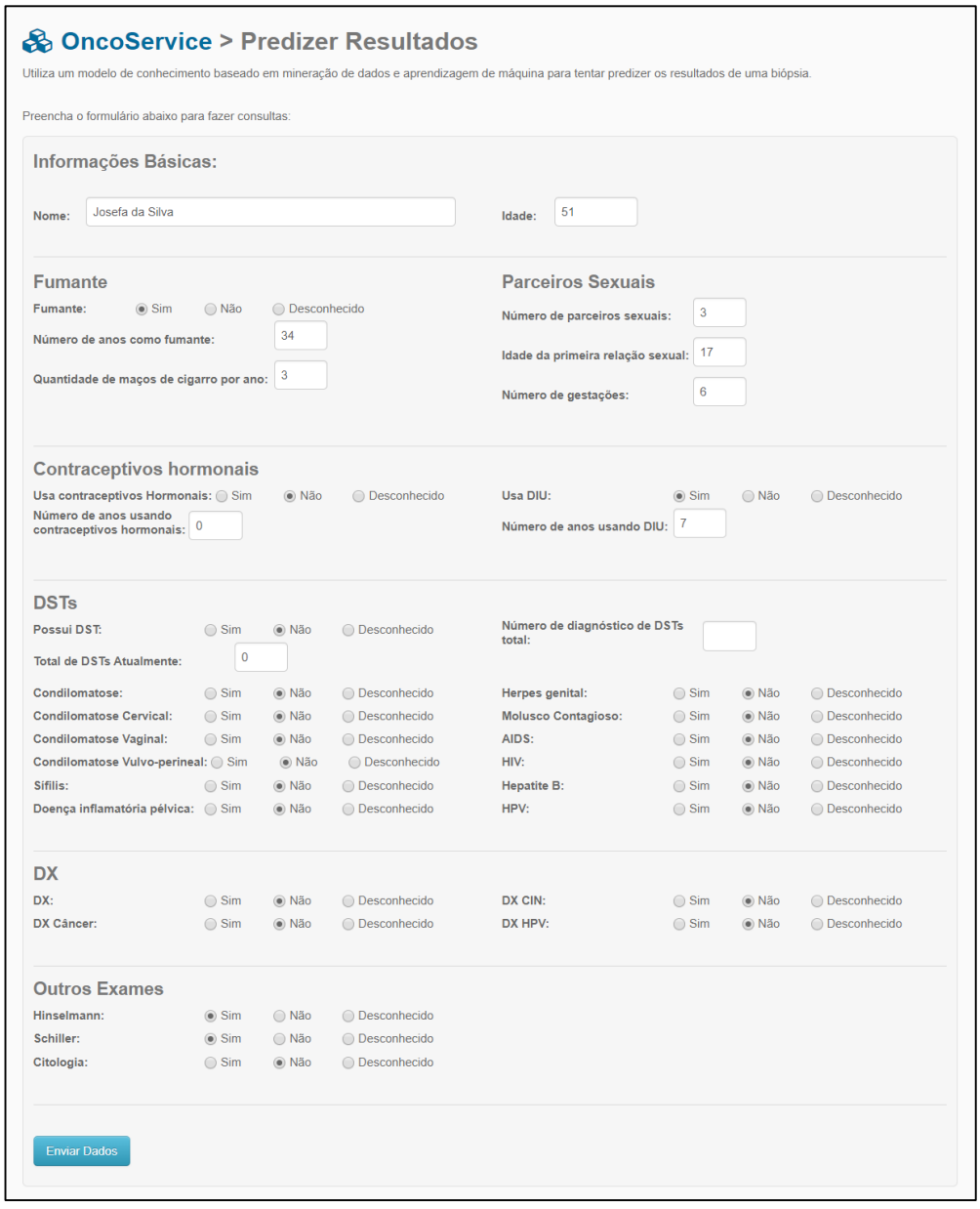
O armazenamento dos dados das consultas dos pacientes foi feito através do banco de dados de consultas, subcomponente do servidor de conhecimento. Esse banco de dados foi responsável por organizar as informações referentes a cada paciente e consulta realizada, armazenando informações tanto sobre resultados de previsões quanto sobre resultados obtidos de biópsias realizadas. O banco de dados em questão poderá ser utilizado pelo engenheiro de conhecimento de forma a melhorar o modelo de conhecimento em etapas futuras de utilização do serviço.

O conjunto de parâmetros de entrada para cada método foi especificado pelo serviço de conhecimento e disponibilizado aos consumidores de conhecimento através do servidor da API de comunicação, sendo seus campos definidos com base nos atributos requeridos pela fonte de dados e banco de dados de consultas.

Portanto, os formulários de entrada e seus metadados foram providos pelo servidor da API de comunicação aos aplicativos consumidores de conhecimento que, por sua vez, são responsáveis por renderizá-los dinamicamente, possibilitando a inserção de dados pelos usuários do sistema.

Por fim, o aplicativo consumidor de conhecimento não necessitou de modificações, visto que, devido à sua flexibilidade e capacidade de receber as especificações dos formulários de entrada e telas de saída via API de comunicação, permitiu a disponibilização do OncoService baseado apenas nas especificações dos serviços e métodos providos pelo serviço provedor de conhecimento. A Figura 41 apresenta o formulário de entrada, gerado dinamicamente, a partir das especificações do método para predizer resultados, em que os atributos de entrada foram organizados por tópicos, de acordo com os atributos necessários para a execução do método.

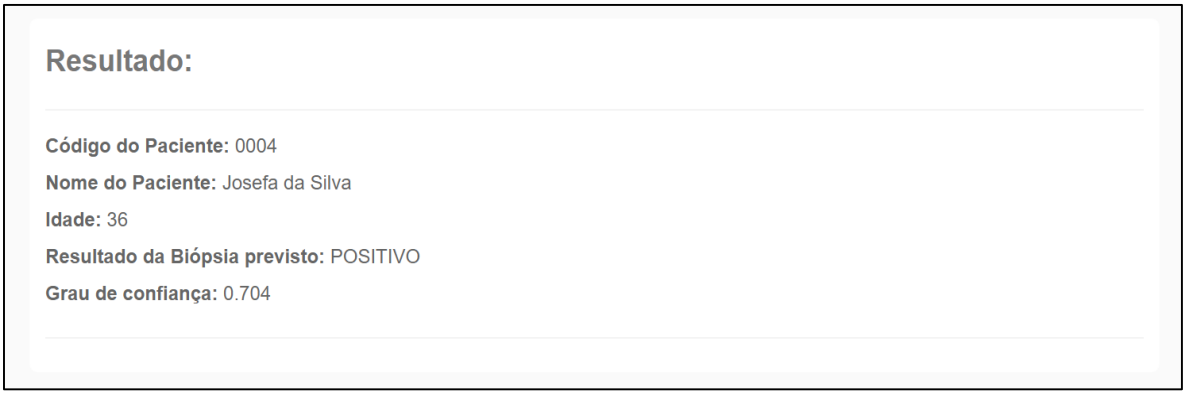
Figura 41 - Formulário de entrada para o método de predição de resultados



Fonte: Elaborado pelo autor.

A partir dos dados inseridos pelos usuários da plataforma, após o processamento pelo servidor de conhecimento e, consequentemente, pelo extrator de conhecimento, os resultados da consulta são apresentados ao usuário, de maneira a fornecer o resultado previsto para a biópsia e o grau de confiança associado à predição (Figura 42). Adicionalmente, é fornecido um identificador único numérico que permite o uso futuro dessa predição pelo método “informar biópsia”.

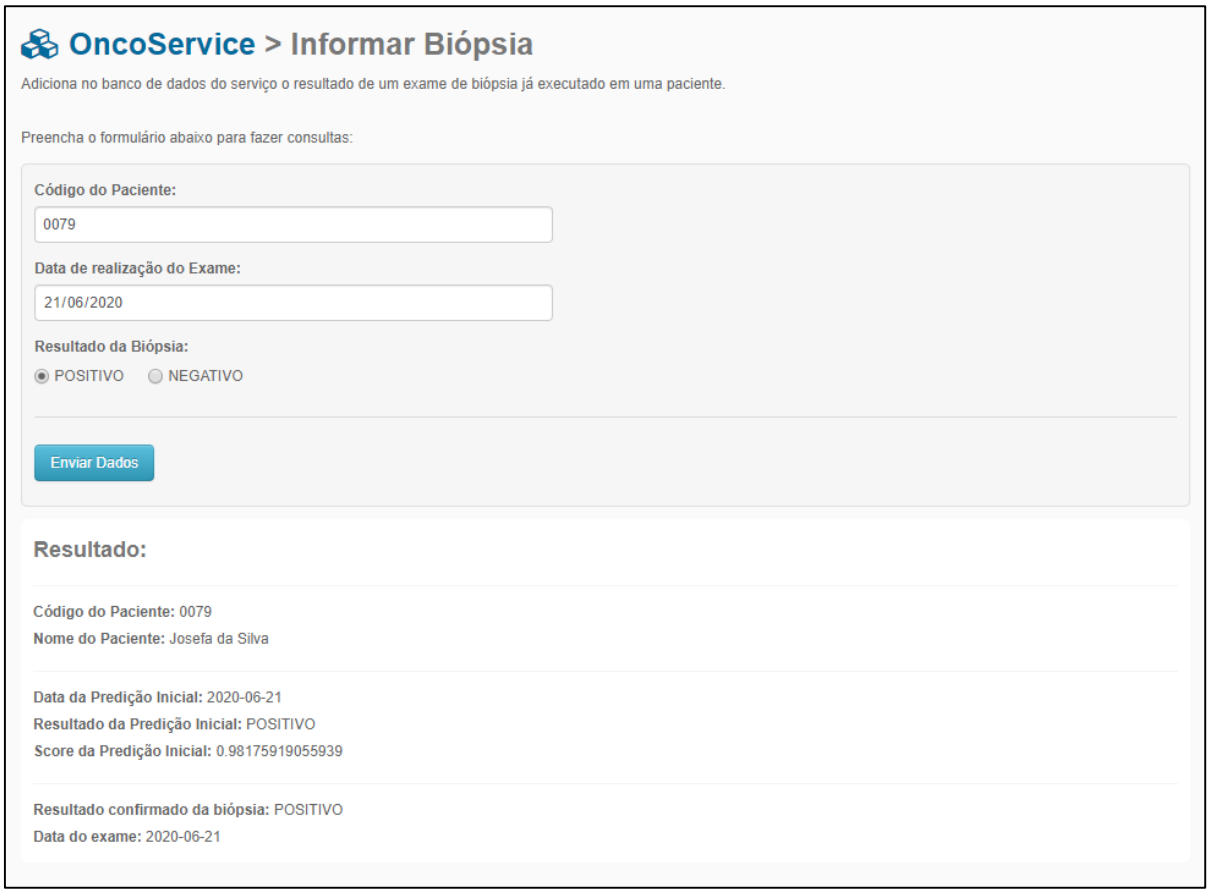
Figura 42 - Resultado de uma consulta executada ao método de predizer resultados



Fonte: Elaborado pelo autor.

Por fim, a Figura 43 apresenta o formulário de entrada para o método “informar biópsia” e um exemplo de resultado obtido, onde os dados de uma predição, anteriormente executada, são comparados com o valor recém-cadastrado do exame de biópsia da paciente.

Figura 43 - Formulário de entrada e exemplo de execução do método “informar biópsia”



Fonte: Elaborado pelo autor.

5.2.3. RESULTADOS

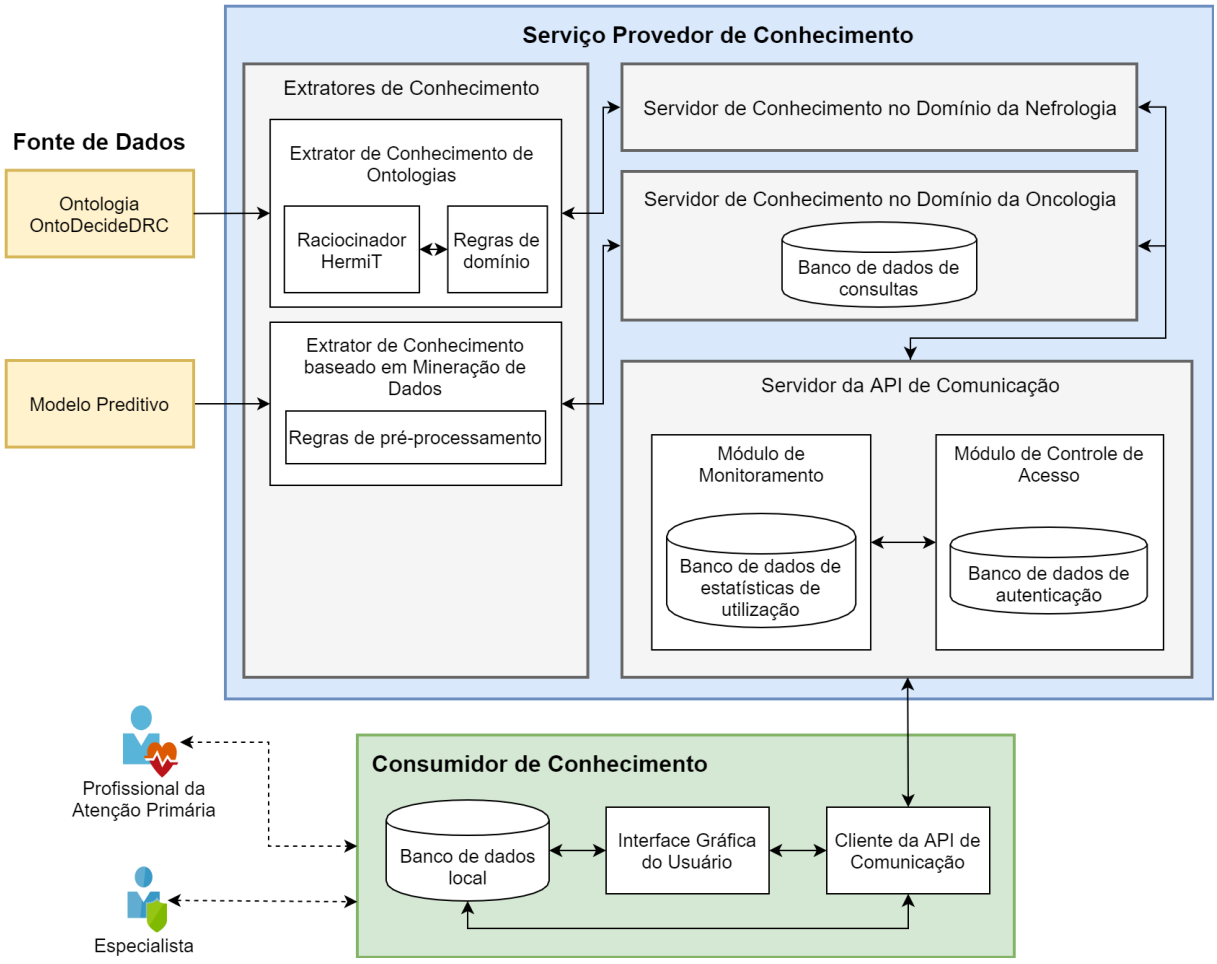
O sistema de suporte à decisão clínica desenvolvido através do uso da arquitetura H-KaaS permitiu o acesso ao conhecimento modelado por meio de técnicas de aprendizagem de máquina e mineração de dados.

A rede neural artificial projetada – fonte de dados empregada neste estudo –, treinada a partir de um conjunto de dados de pacientes com suspeita de câncer cervical, foi capaz de prever corretamente, em média, o resultado da biópsia para 95,06% dos pacientes presentes no conjunto de dados de teste.

Nesse contexto, o extrator de conhecimento desenvolvido possibilitou a execução de consultas à fonte de dados de maneira a permitir a predição de resultados de exames de biópsias através das informações fornecidas pelo paciente e de acordo com seu histórico médico.

Dessa forma, de maneira similar ao estudo de caso anterior, o servidor de conhecimento foi implementado de acordo com as especificações da H-KaaS, reutilizando componentes já implementados como o website consumidor de conhecimento e o servidor da API de comunicação. Além disso, diferentemente do servidor de conhecimento na nefrologia, houve a necessidade do desenvolvimento de um componente de banco de dados para armazenamento de consultas e dados de pacientes que, por sua vez, foi utilizado para prover dados históricos e para guardar informações sobre predições realizadas e resultados de exames confirmados. A Figura 44 mostra a instância da arquitetura H-KaaS, após a adição dos novos componentes desenvolvidos neste estudo de caso, destacando a execução em paralelo dos servidores de conhecimento, um no domínio da nefrologia, resultado do estudo de caso anterior, e outro no domínio da oncologia, implementado neste estudo de caso.

Figura 44 - Componentes instanciados da arquitetura de referência H-KaaS após a implementação do segundo estudo de caso



Fonte: Elaborado pelo autor.

Este estudo de caso diferencia-se do anterior porque, além de ter sido executado em um subdomínio diferente da saúde, na oncologia, valeu-se de mineração de dados para a extração e compartilhamento de conhecimento. Outro ponto que diferencia a abordagem foi o uso de um banco de dados de consultas presente no servidor de conhecimento que, além de armazenar o histórico de consultas e resultados das predições, pode servir como base para que um mecanismo de refinamento do modelo de conhecimento seja implementado.

A arquitetura H-KaaS, ao ser instanciada como uma arquitetura de software concreta, mostrou-se flexível e capaz de centralizar o acesso ao conhecimento provido por diferentes serviços de conhecimento, em diferentes subdomínios da saúde, e por múltiplas fontes de dados.

Em conclusão, este estudo de caso instanciou os componentes necessários para permitir o compartilhamento do modelo de conhecimento desenvolvido, oferecendo-o como um serviço, de maneira a reutilizar componentes resultantes da execução do primeiro estudo de caso. Além disso, ambos os serviços de suporte à decisão clínica foram executados em paralelo, através do mesmo serviço provedor de conhecimento, evidenciando a possibilidade de composição de sistemas complexos empregando múltiplos servidores de conhecimento e reutilizando instâncias de aplicativos consumidores.

6. RESULTADOS E DISCUSSÃO

Este capítulo expõe os resultados obtidos com a realização desta pesquisa e discute como a arquitetura de referência proposta pode ser utilizada, tanto na elaboração de novos sistemas de suporte à decisão quanto na comparação entre sistemas similares existentes.

Como principal resultado do desenvolvimento desta pesquisa, uma arquitetura de referência foi proposta e validada, baseada no paradigma de conhecimento como serviço, para o domínio da saúde. Inicialmente, foi executada uma revisão sistemática da literatura, através da qual foram obtidos dados sobre sistemas existentes para o compartilhamento de conhecimento no domínio da saúde, permitindo que estes servissem de inspiração e que, em conjunto com as diretrizes fornecidas pelo paradigma KaaS, viabilizassem o desenvolvimento da arquitetura em questão.

Os componentes da arquitetura foram descritos e suas possíveis aplicações foram exemplificadas. Além disso, foram apresentadas as interfaces de comunicação com as fontes de dados e consumidores de conhecimento, discutindo como elas poderão ser implementadas e especificadas ao se projetar uma arquitetura de software concreta baseada na H-KaaS.

A adaptação ao domínio da saúde possibilitou a especificação e validação de uma arquitetura de referência capaz de lidar com fontes de dados, baseadas em dados brutos, modelos de conhecimento implícitos ou explícitos, e aplicativos consumidores específicos do domínio, como por exemplo uma ontologia no domínio da nefrologia, além de permitir a discussão sobre a possibilidade da extração de conhecimento de fontes baseadas em diretrizes clínicas, resultados de exames e dados de prontuários eletrônicos.

Dessa forma, a H-KaaS foi descrita de maneira a facilitar o entendimento de seus componentes e a permitir sua validação através da execução de diferentes estudos de caso. Neles, foram instanciados dois sistemas de suporte à decisão clínica, utilizando diferentes fontes de dados e modelos de conhecimento – um no domínio da nefrologia, e outro no domínio da oncologia. Adicionalmente, o acesso ao conhecimento foi feito através de um aplicativo consumidor de conhecimento compartilhado, demonstrando algumas das características da H-KaaS, como sua flexibilidade em lidar com diversas fontes de dados e a possibilidade de execução de múltiplos servidores de conhecimento, bem como do reuso de diversos componentes, como aplicativos consumidores e extratores, e o servidor da API de comunicação.

Além disso, extratores de conhecimento foram desenvolvidos de forma a serem capazes de obter o conhecimento de suas respectivas fontes: um deles utilizando um algoritmo

raciocinador para executar inferências em ontologias de domínio no formato OWL, e outro capaz de acessar o conhecimento armazenado em modelos preditivos baseados em aprendizagem de máquina.

A arquitetura H-KaaS, por ser uma arquitetura de referência, propõe componentes comuns a sistemas de compartilhamento de conhecimento no domínio da saúde. Porém, para o desenvolvimento de uma arquitetura concreta de software baseada na H-KaaS pode ser necessária a criação de vários outros subcomponentes, específicos ao problema a ser resolvido. Portanto, espera-se que a H-KaaS forneça a base para o desenvolvimento dessas arquiteturas concretas de software, que, por sua vez, determinarão como esses componentes (e quais deles) serão instanciados.

Devido ao fato de a arquitetura H-KaaS ser proposta para o domínio da saúde, os estudos de caso foram escolhidos de maneira a implementar servidores de conhecimento em diferentes subdomínios. Como ambos foram instanciados de forma independente, em um mesmo serviço provedor de conhecimento, foi possível demonstrar o uso de arquiteturas baseadas na H-KaaS em duas situações distintas:

- A aplicação da H-KaaS para um subdomínio específico, com apenas um servidor de conhecimento, conforme resultados do primeiro estudo de caso;
- A integração de múltiplos servidores de conhecimento em uma mesma instância da H-KaaS, em subdomínios diferentes da saúde, de acordo com os resultados apresentados pelo segundo estudo de caso.

Dessa forma, fica evidente a possibilidade de criação de instâncias da arquitetura de referência H-KaaS dedicadas a um subdomínio específico da saúde ou de domínio geral, onde podem ser disponibilizados vários servidores de conhecimento focados em diferentes subdomínios da saúde.

No contexto do segundo estudo de caso realizado, um ponto importante a se notar é que várias outras abordagens poderiam ter sido escolhidas para a criação do modelo de conhecimento a ser compartilhado. Uma rede neural artificial foi escolhida com o objetivo de demonstrar como a H-KaaS poderia ser instanciada de forma a lidar com modelos de conhecimento implícitos, criados a partir de técnicas de mineração de dados e algoritmos de aprendizagem de máquina, diferenciando este estudo de caso do anterior.

Ademais, embora esta pesquisa tenha descrito a metodologia utilizada para o treinamento da RNA e consequente criação do modelo de conhecimento, este não era o principal foco do estudo de caso. Sendo assim, escolhas como o banco de dados de pacientes e

detalhes da rede neural utilizada, como o número de camadas, funções de ativação e pré-processamento, embora afetem diretamente na acurácia do modelo de conhecimento resultante, tiveram pouco impacto na adaptação das fontes de dados à instância da H-KaaS. Em conclusão, o modelo de conhecimento desenvolvido serviu para exemplificar e validar a utilização dos componentes previstos pela H-KaaS em outro subdomínio da saúde.

Um dos desafios correntes no domínio da informática em saúde está na interoperabilidade entre sistemas distintos e em como os dados de pacientes são armazenados e distribuídos de maneira segura, entre os diferentes sistemas que os utilizam. No Brasil, por exemplo, vários sistemas de informação em saúde são disponibilizados pelo Departamento de Informática do Sistema Único de Saúde (DataSUS), que tem a missão de promover a modernização por meio da tecnologia da informação para apoiar o Sistema Único de Saúde (MINISTÉRIO DA SAÚDE, 2020b). Como exemplos desses sistemas, podem ser citados:

- **Sistema de Informações Hospitalares do Sistema Único de Saúde (Sihsus):** Visa a registrar e gerar relatórios sobre os atendimentos e internações hospitalares financiados pelo Sistema Único de Saúde (MINISTÉRIO DA SAÚDE, 2010);
- **Sistema de Controle de Exames Laboratoriais de CD4+/CD8+ e Carga Viral do HIV (Siscel):** Um sistema de informação que tem como objetivo facilitar o controle de processos de cadastramento de pacientes e armazenamento do histórico de exames realizados (MINISTÉRIO DA SAÚDE, 2020a);
- **Sistema de Informações de Agravos de Notificação (Sinan):** Tem como objetivo apoiar a análise de informações de vigilância de doenças, armazenando, transmitindo e compartilhando dados gerados rotineiramente pelo sistema de vigilância epidemiológica do governo (MINISTÉRIO DA SAÚDE, 2007);

Nesse contexto, a arquitetura proposta tenta flexibilizar o acesso a esse tipo de dado e permite duas interpretações em relação a como esses sistemas seriam integrados à H-KaaS: (1) como uma fonte de dados ou (2) como um banco de dados de consultas, presentes no servidor de conhecimento.

Ambas as interpretações são possíveis, porém, a escolha de como o acesso a esses dados é feita deve levar em consideração as características de cada um desses subcomponentes. As fontes de dados são mais flexíveis, pois não estão necessariamente dentro do serviço provedor de conhecimento, e podem ser desenvolvidas e mantidas em organizações diferentes. Além disso, uma mesma fonte de dados pode ser compartilhada entre diferentes servidores de conhecimento, permitindo seu reuso e a extração do conhecimento com o mesmo componente

extrator. Nesse contexto, utilizando a interpretação mencionada, os sistemas de informação Sihsus, Siscel, Sinan e similares seriam tratados como fontes de dados externas à organização.

Por outro lado, caso esses sistemas de informação fossem implementados como um componente no banco de dados de consultas, seriam parte de um servidor de conhecimento específico que, conforme discutido durante a especificação desse componente, precisaria ser mantido pela mesma organização responsável pelo desenvolvimento da instância da H-KaaS, que, neste exemplo, seria o DataSUS.

Dessa forma, caso a H-KaaS seja instanciada por uma organização diferente do DataSUS (como um hospital, clínica, universidade, etc.), recomenda-se utilizar os dados providos por sistemas similares como uma das fontes de dados do sistema, desenvolvendo extratores de conhecimento capazes de implementar as interfaces de comunicação requeridas. Conforme mencionado, essa abordagem permite uma maior flexibilidade em relação a como os dados são obtidos, devido à separação entre o que é parte do serviço provedor de conhecimento e os sistemas providos pelo DataSUS, ou qualquer outro sistema similar externo à organização.

Notou-se certa limitação em algumas situações específicas da arquitetura H-KaaS, quando utilizada como meio de comparação entre sistemas, como, por exemplo, em certos domínios, como a internet das coisas aplicada à saúde, ou quando o compartilhamento de conhecimento é feito de forma distribuída, apesar de esta arquitetura ter sido criada com inspiração em sistemas reais de compartilhamento de conhecimento no domínio da saúde, conforme apresentado no capítulo 3.

O uso da H-KaaS para descrever arquiteturas distribuídas pode-se mostrar um desafio, visto que a H-KaaS, por ser uma arquitetura centralizada, parte do princípio de que haverá um componente central responsável por coordenar a comunicação entre os diversos aplicativos consumidores de conhecimento e fontes de dados. Dessa forma, o uso da arquitetura H-KaaS não é recomendável nessa situação, a não ser que cada nó da rede possa ser considerado uma instância distinta da H-KaaS.

No domínio da internet das coisas, quando aplicado à área da saúde, por exemplo, há muitas vezes a necessidade de gerenciamento de um grande número de sensores com o objetivo de capturar informações em tempo real (ZGHEI et al., 2017). Dessa forma, dependendo do problema a ser resolvido e do subdomínio a ser escolhido, a H-KaaS pode ou não facilitar o desenvolvimento e a comparação de novos sistemas e, em alguns casos, serão necessárias adaptações e criação de interfaces não previstas pela H-KaaS.

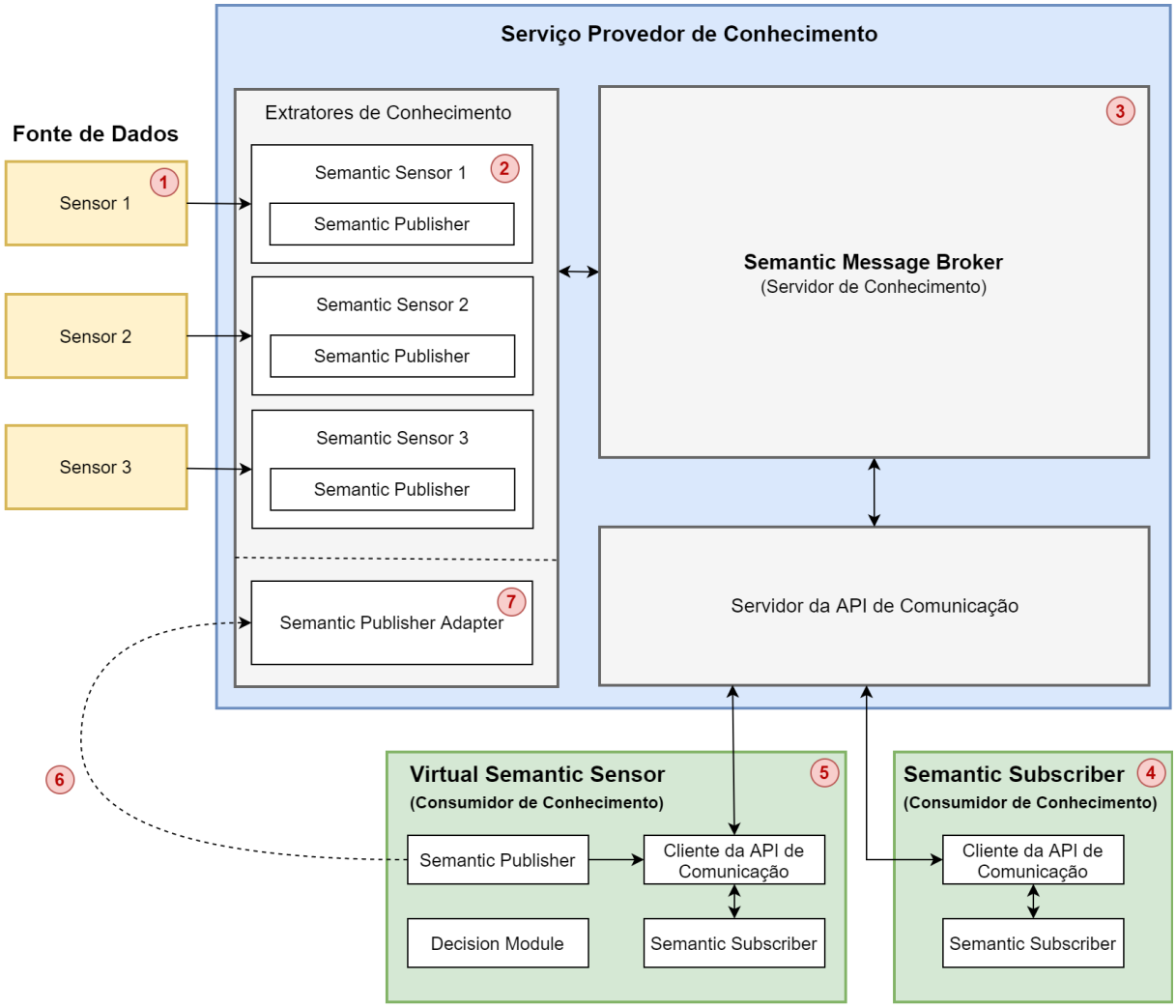
Um exemplo que pode ser citado, baseado nas arquiteturas estudadas que serviram de inspiração para a H-KaaS, encontra-se na arquitetura proposta por Zghei et al. (2017), apresentada anteriormente no capítulo 3, na Figura 12.

Embora os autores visem a propor uma arquitetura baseada no domínio da internet das coisas aplicada à saúde, com o objetivo de lidar com dados de múltiplos sensores provedores de dados em tempo real, a solução apresentada descreve uma arquitetura centralizada, capaz de lidar com múltiplas fontes de dados (chamados de *semantic publishers*) e múltiplos consumidores (chamados de *semantic subscribers*), através de um barramento central capaz de gerenciar a comunicação entre as diferentes partes do sistema (chamado de *semantic message broker*).

Dessa forma, sob a visão da H-KaaS essa arquitetura poderia ser adaptada conforme apresentado na Figura 45. A figura mostra como os componentes dessa arquitetura poderiam ser instanciados seguindo as diretrizes e especificações dos componentes da H-KaaS. Os sensores físicos (1) poderiam ser interpretados como fontes de dados, sendo suas versões semânticas associadas (2) representadas por extratores de conhecimento dedicados ao sensor. Além disso, o barramento central (3), chamado de *semantic message broker*, ficaria responsável por monitorar e repassar as mensagens entre as diversas partes do sistema, conforme sua especificação original, sendo implementado como um servidor de conhecimento. Além disso, os *semantic subscribers* (4) e o *virtual semantic sensor* (5) podem ser implementados como consumidores de conhecimento.

É importante notar que, de acordo com os autores, o *virtual semantic sensor* também deve ser capaz de publicar dados baseados em seu módulo de decisão semântica através de um *semantic publisher*. Sendo assim, uma possível abordagem seria permitir a ligação (6) entre o consumidor e um extrator de conhecimento dedicado, chamado de *semantic publisher adapter* (7), algo não previsto pela H-KaaS. Outra solução para o problema, utilizando apenas as interfaces previstas, seria fazer com que seu *semantic publisher* usasse apenas os comandos disponibilizados pelo servidor da API de comunicação (8), evitando, assim, a comunicação direta com o extrator de conhecimento.

Figura 45 - Adaptação de uma arquitetura do domínio da internet das coisas à H-KaaS



Fonte: Elaborado pelo autor, baseado na arquitetura proposta por Zghei et al. (2017),

Dessa forma, o exemplo apresentado mostra que, embora a H-KaaS tenha se mostrado flexível em relação à implementação de novos sistemas de suporte à decisão clínica, ao se adaptar sistemas reais a suas especificações, talvez ainda seja necessária a criação de interfaces e componentes adicionais, não especificados anteriormente.

No contexto do paradigma de conhecimento como serviço, as fontes de dados (empregando a nomenclatura proposta pela H-KaaS) são chamadas em inglês de "*data owners*" (XU; ZHANG, 2005). Essa nomenclatura tem como principal objetivo mostrar que esses componentes podem ser mantidos por organizações diferentes, uma das principais características do paradigma KaaS. Diante disso, outras nomenclaturas que, de certo modo, poderiam ter sido utilizadas para representar esses mesmos componentes na H-KaaS são: fontes de conhecimento, detentores de dados, modelos de conhecimento e proprietário de dados. Essa

falta de padronização ficou evidente durante a realização da revisão sistemática da literatura, realizada antes da proposta da H-KaaS, que utilizou o termo “fontes de conhecimento” para representar esse componente durante a análise.

Na grande maioria das situações, os dados partem das fontes em direção ao serviço provedor de conhecimento, e a palavra “dado”, na computação, pode abranger tanto “informação” quanto “conhecimento” (VALENTIM, 2002). Portanto, ao definir os componentes da arquitetura de referência H-KaaS, foi escolhido o termo “fonte de dados” para padronizar o nome desse componente.

Em conclusão, este capítulo discutiu os resultados obtidos durante este projeto de pesquisa a partir da especificação e validação da arquitetura H-KaaS, abordando temas como o seu uso em diferentes subdomínios da saúde, a interoperabilidade com sistemas existentes, formas de armazenamento de dados de pacientes e consultas, possíveis dificuldades em sua utilização e, por fim, a nomenclatura empregada em seus componentes.

7. CONCLUSÕES

Neste capítulo, são apresentadas as considerações finais do presente trabalho, principais contribuições, limitações, trabalhos futuros relacionados ao tema, e, por fim, publicações associadas a esta pesquisa.

No domínio da saúde, apesar de existirem várias propostas de padrões para semântica e interoperabilidade de dados, faltam meios para facilitar o compartilhamento de conhecimento e comparar sistemas existentes que tenham esse propósito.

Diante disso, esta pesquisa teve como objetivo geral projetar uma arquitetura de referência baseada no paradigma de conhecimento como serviço para o domínio da saúde. Dessa forma, a arquitetura de referência H-KaaS foi desenvolvida para ser usada na área da saúde, inspirada pelos componentes do paradigma de conhecimento como serviço e no resultado da análise de sistemas similares existentes no domínio.

O objetivo específico inicial era realizar uma revisão sistemática da literatura, com o intuito de coletar dados sobre arquiteturas, paradigmas e protótipos existentes capazes de compartilhar conhecimento no domínio da saúde. Esse objetivo foi atendido e seus resultados serviram de entrada para a metodologia utilizada para a especificação da H-KaaS.

Em seguida, conforme descrito pelo segundo e terceiro objetivos específicos, também atingidos, os componentes e interfaces da H-KaaS foram identificados e descritos de forma a exemplificar sua utilização e fornecer detalhes sobre seu funcionamento.

Dessa forma, foi possível realizar a especificação e o refinamento da arquitetura H-KaaS, visando a facilitar o compartilhamento de conhecimento e criação de meios para a análise de sistemas existentes, atingindo o quarto objetivo específico desta pesquisa.

Por fim, o quinto objetivo específico foi alcançado através da execução de dois estudos de caso com a finalidade de validar e exemplificar a arquitetura proposta. O primeiro, no domínio da nefrologia, adaptou um sistema de suporte à decisão clínica baseado em ontologias de domínio à H-KaaS. De maneira similar, o segundo estudo de caso, realizado no domínio da oncologia, além de demonstrar a implementação de novos componentes, mostrou como múltiplos servidores de conhecimento podem ser executados em um mesmo serviço provedor de conhecimento. Além disso, a execução dos estudos de caso demonstrou como componentes extratores de conhecimento podem ser implementados e instanciados dentro da H-KaaS, além de descrever o fluxo de conhecimento dentro dela.

Esta pesquisa partiu da hipótese de que o paradigma de conhecimento como serviço pode ser aplicado ao domínio da saúde, facilitando o compartilhamento de conhecimento extraído de várias fontes de dados e permitindo que aplicativos consumidores de conhecimento o acessem de forma centralizada. Durante a realização desta pesquisa, a partir da especificação da arquitetura H-KaaS e da realização dos estudos de caso, confirmou-se a veracidade da hipótese, pois foi possível instanciar a arquitetura proposta em múltiplos subdomínios da saúde, demonstrando como o acesso ao conhecimento pode ser padronizado e facilitado por meio de um serviço provedor de conhecimento, conforme descrito pelo paradigma KaaS, capaz de lidar com múltiplas fontes de dados com diferentes formas de representação do conhecimento e raciocínio.

Em conclusão, o desenvolvimento desta pesquisa abrirá caminho para que, na área da saúde, novas fontes de dados e conhecimento sejam desenvolvidas e melhor disponibilizadas, permitindo a criação de serviços sofisticados baseados em conhecimento, possibilitando um maior aproveitamento do conhecimento gerado através das experiências dos profissionais da área médica e facilitando a criação de novas aplicações capazes de utilizar esse conhecimento, consequentemente contribuindo para o avanço do estado da arte da informática em saúde e de aplicações da inteligência artificial e seus sistemas baseados em conhecimento.

7.1. PRINCIPAIS CONTRIBUIÇÕES

A arquitetura de referência H-KaaS mostrou-se promissora no que diz respeito à distribuição e acesso ao conhecimento. Dessa forma, como principal contribuição desta pesquisa, espera-se que a arquitetura de referência proposta seja capaz de facilitar o compartilhamento de modelos de conhecimento existentes no domínio da saúde, melhorando o atendimento e a tomada de decisão por profissionais da área, além de favorecer o desenvolvimento de novas aplicações que possam se beneficiar de tal conhecimento. Adicionalmente, a arquitetura proposta pode ser utilizada como um modelo para a comparação e análise entre sistemas existentes no domínio com finalidades semelhantes.

Portanto, esta pesquisa propõe uma arquitetura de referência baseada em conhecimento como serviço para a área da saúde, capaz de facilitar o compartilhamento de modelos de conhecimento e algoritmos extratores, possibilitando seu melhor uso em domínios em que, embora exista uma grande quantidade de dados sendo coletados diariamente, ainda não haja formas eficientes para que o conhecimento seja repassado de maneira satisfatória.

Além disso, foi executada uma revisão sistemática da literatura, com uma abrangência de cinco anos, onde foram identificadas arquiteturas, paradigmas e protótipos cujo objetivo é o compartilhamento de conhecimento e dados na área da saúde, respondendo a seis questões de pesquisa relevantes. Adicionalmente, foram coletados dados sobre os componentes e interfaces de comunicação dessas arquiteturas, podendo ser utilizados em pesquisas futuras.

Cada estudo de caso possibilitou uma contribuição única em seu respectivo domínio, visto que seus protótipos, através das informações fornecidas pelos profissionais da saúde, foram capazes de extrair conhecimento útil de suas respectivas fontes de dados, fornecendo suporte à decisão clínica.

No primeiro estudo de caso, realizado no domínio da nefrologia, foi adaptado um sistema já existente para a arquitetura de referência KaaS. Também, se realizou o detalhamento de como consultas ao conhecimento são executadas no contexto de uma arquitetura KaaS, mostrando como esta permite a execução de consultas à base de conhecimento, através do uso de extratores, possibilitando a criação de mecanismos de suporte à decisão.

Além disso, foi descrita uma API de comunicação entre o serviço provedor de conhecimento e os consumidores, facilitando a interoperabilidade e o desenvolvimento de novos aplicativos capazes de interagir com diferentes servidores de conhecimento.

Em relação ao segundo estudo de caso, realizado na área da oncologia, tendo em vista o detalhamento de como seu modelo de conhecimento foi desenvolvido e adaptado à H-KaaS, foi possível entender como novas aplicações, no domínio da saúde, podem ser beneficiadas pela escolha da arquitetura de referência proposta em estágios iniciais do projeto.

Outra contribuição importante desse estudo de caso foi o desenvolvimento de uma base de conhecimento, através de técnicas de mineração de dados e do uso de algoritmos de aprendizagem de máquina supervisionada, para predição de resultados de exames relacionados a câncer cervical, possibilitando a elaboração de um sistema de suporte à decisão clínica, permitindo a implementação e validação de uma instância da arquitetura H-KaaS baseada em conhecimento do tipo implícito, extraído por meio de algoritmos de aprendizagem de máquina, que podem vir a dar suporte a atividades de decisão clínica.

Por fim, foi realizado o projeto e implementação de dois componentes extratores de conhecimento, capazes de acessar e raciocinar sob conhecimento representado por ontologias de domínio e redes neurais artificiais.

Em conclusão, o desenvolvimento da arquitetura de referência H-KaaS abre novos caminhos para um melhor aproveitamento do conhecimento gerado através das experiências

dos profissionais e de serviços de saúde, além de facilitar a criação e análise de aplicações capazes de utilizar esse conhecimento.

7.2. LIMITAÇÕES DA PESQUISA

Diante da metodologia aplicada na realização desta pesquisa, que se valeu de dados obtidos em uma revisão sistemática da literatura, e através do estudo do paradigma KaaS, com a proposta da arquitetura H-KaaS e a validação por meio de dois estudos de caso, foram identificadas algumas limitações que podem afetar o uso da arquitetura de referência proposta, tanto quando usada como base para novos sistemas de compartilhamento de conhecimento na área da saúde quanto como quando usada como base de comparação entre sistemas existentes.

Por ser uma arquitetura de referência na área da saúde, e devido à grande gama de aplicações e serviços que podem ser oferecidos, algumas escolhas quanto ao uso de subcomponentes só podem ser feitas pelo engenheiro de conhecimento ou arquiteto em fase de implementação. Dessa forma, o desenvolvedor, ao utilizar a arquitetura H-KaaS para a criação de um novo sistema de software, deve levar em consideração a análise de outros sistemas semelhantes, as diretrizes apresentadas nas definições de componentes da H-KaaS e, é claro, sua própria experiência como base para a tomada de decisão em relação às tecnologias a serem empregadas.

Outra limitação importante desta pesquisa está relacionada aos estudos de caso que, embora abrangentes, não cobrem todas as subáreas da saúde nem todas as técnicas da IA de sistemas baseados em conhecimento e, portanto, não se pode garantir que a arquitetura será adequada em todas as situações. Portanto, mais estudos de caso são necessários a fim de validar a arquitetura em outros subdomínios ou outros serviços baseados em conhecimento. Além disso, não foram executados estudos de caso com fontes de dados que proveem dados em tempo real, como redes de sensores baseados em internet das coisas. Essas fontes possuem a característica de gerar uma enorme quantidade de dados em pouco tempo, gerando assim desafios quanto sua escalabilidade, sendo necessária a otimização quanto à latência das consultas e acesso aos dados.

Por fim, ontologias da área da saúde, em geral, são grandes e complexas. Por esta razão, arquiteturas baseadas em paradigmas que tentam centralizar conhecimento e raciocínio podem sofrer com a falta de recursos computacionais, dificultando sua operação. Não foi proposta uma solução para que o conhecimento seja acessado parcialmente ou que sistemas de raciocínio complexos sejam mapeados para regras mais simples e escaláveis, como, por exemplo, através

da criação de teoremas. A composição de servidores de conhecimento pode ser uma solução plausível para o problema, porém, devido a restrições de tempo e recursos desta pesquisa, a arquitetura H-KaaS não foi validade nesse contexto.

7.3. TRABALHOS FUTUROS

Durante o desenvolvimento desta pesquisa, foram identificados possíveis temas para a realização de trabalhos futuros relacionados à arquitetura proposta. Entre estes, está o refinamento da arquitetura através da realização da continuação da revisão de literatura, tornando-a mais abrangente, utilizando a nomenclatura e descrição de componentes propostos pela H-KaaS como meio de comparação entre os sistemas encontrados. Além disso, poderiam ser usados outros buscadores e palavras-chave, com o objetivo de encontrar diferentes propostas de arquiteturas e sistemas não identificados anteriormente.

Outro trabalho que poderia ser realizado diz respeito ao projeto e implementação de mais extratores e servidores de conhecimento ou sua automação independente de domínio, de maneira a serem capazes de lidar com fontes de dados que usem outros paradigmas da representação de conhecimento e raciocínio, como, por exemplo, modelos estatísticos baseados em redes bayesianas (PEARL, 1987). Estas, diferentemente de ontologias de domínio e redes neurais artificiais, possuem a capacidade de lidar com incertezas através do raciocínio probabilístico e, consequentemente, necessitariam de extratores de conhecimento dedicados, especificamente projetados para suportar esse tipo de raciocínio.

Adicionalmente, podem ser realizados estudos que visem a instanciar e validar a arquitetura proposta em outros subdomínios da saúde, com o objetivo de identificar problemas, propor soluções e exemplificar seu uso nesses domínios.

Uma outra questão levantada durante a realização desta pesquisa foi a de reutilização da arquitetura em domínios que não seja a saúde pois o problema do compartilhamento de conhecimento também existe em outras áreas de conhecimento. Sendo assim, poderiam ser estudadas aplicações da H-KaaS em diferentes áreas do conhecimento, a fim de adaptá-la e torná-la compatível com outros domínios.

Por fim, observou-se a necessidade de um estudo mais aprofundado relacionado à composição de sistemas utilizando a arquitetura H-KaaS, empregando múltiplos componentes servidores e extratores de conhecimento, produzindo relações de dependências entre eles, de

forma a permitir responder a consultas complexas utilizando fontes de dados que, sozinhas, não seriam capazes de respondê-las.

7.4. PUBLICAÇÕES

Como resultados adicionais desta pesquisa, foram publicados resumos, artigos completos e capítulos de livro em conferências e periódicos das áreas da computação e saúde.

Ainda em 2017, na Conferência Ibero Americana WWW/Internet, foi publicado o artigo completo "OntoDRC: prevenindo a doença renal crônica", focado no desenvolvimento de uma ontologia de domínio na área da nefrologia, similar à base de conhecimento utilizada em um dos estudos de caso desta pesquisa: Gomes et al. (2017).

Em 2018, o artigo completo *"H-KaaS: A knowledge-as-a-service architecture for e-health"* foi publicado no periódico *Brazilian Journal of Biological Sciences*, propondo uma versão simplificada da arquitetura H-KaaS: Barreto et al. (2018b).

Também em 2018, o artigo "Clinical decision support based on OWL queries in a knowledge-as-a-service architecture" foi apresentado na conferência internacional "RuleML+RR" e publicado em *Lecture Notes in Computer Science*, volume 11092. Sua principal contribuição foi o detalhamento do fluxo de conhecimento dentro dos módulos da arquitetura proposta: Barreto et al. (2018a).

Ainda em 2018, com o objetivo de entender o funcionamento das redes bayesianas usadas para o suporte à decisão clínica na área da nefrologia, o trabalho "Rede bayesiana e ontologia: uma abordagem no domínio da nefrologia" foi publicado no XVI Congresso Brasileiro de Informática em Saúde – CBIS 2018, tendo como objetivo a proposta de uma metodologia para criação de redes bayesianas a partir de uma ontologia de domínio: Souza et al. (2018).

Ao fim de 2018 e durante o primeiro semestre de 2019, a partir de pesquisas na área de aprendizagem de máquina e oncologia, foco de um dos estudos de caso, foram publicados resumos, artigos completos e um capítulo de livro em conferências e periódicos da área da saúde: Barreto et al. (2018), Marinho et al. (2018), Dantas et al. (2018), Dantas et al. (2019) e Barreto et al. (2019).

Em conclusão, as publicações mencionadas contribuíram para o refinamento da arquitetura de referência H-KaaS, por tornar possível o entendimento das dificuldades enfrentadas por pesquisadores ao desenvolverem novos modelos de conhecimento e ao

projetarem sistemas para o compartilhamento de conhecimento e suporte à decisão clínica no domínio da saúde.

REFERÊNCIAS

ABADI, M. et al. TensorFlow: A system for large-scale machine learning. **CoRR**, v. abs/1605.0, 2016.

ABATAL, A.; KHALLOUKI, H.; BAHAJ, M. A semantic smart interconnected healthcare system using ontology and cloud computing. abr. 2018, [S.l.]: IEEE, abr. 2018. p. 1–5.

AKATKIN, Y. M. et al. Application of semantic integration methods for cross-agency Information sharing in healthcare. **Proceedings of the XXVI International Symposium on Nuclear Electronics & Computing (NEC'2017)**, v. 2023, p. 324–329, 2017.

ALAMRI, A.; BERTOK, P.; FAHAD, A. Towards an architecture for managing semantic knowledge in semantic repositories. **International Journal of Parallel, Emergent and Distributed Systems**, v. 30, n. 5, p. 411–425, 3 set. 2015.

ALENCAR, R. A.; CIOSAK, S. I. Late diagnosis and vulnerabilities of the elderly living with HIV/AIDS. **Revista da Escola de Enfermagem da USP**, v. 49, n. 2, p. 0229–0235, abr. 2015.

ALMEIDA, M. B. Revisiting ontologies: A necessary clarification. **Journal of the American Society for Information Science and Technology**, v. 64, n. 8, p. 1682–1693, ago. 2013.

ALONSO, F. et al. Combining expert knowledge and data mining in a medical diagnosis domain. **Expert Systems with Applications**, v. 23, n. 4, p. 367–375, 2002.

ALONSO, G. et al. Web Services. **Web Services: Concepts, Architectures and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 123–149.

AMAZON. **Amazon S3: Armazenamento de objetos simples, resiliente e massivamente escalável**. Disponível em: <https://aws.amazon.com/pt/s3/>. Acesso em: 26 nov. 2018.

ANYA, O.; TAWFIK, H.; AL-JUMEILY, D. Context-Aware Clinical Knowledge Sharing in Cross-Boundary E-Health: A Conceptual Model. out. 2015, [S.l.]: IEEE, out. 2015. p. 589–595.

ARBYN, M. et al. Worldwide burden of cervical cancer in 2008. **Annals of Oncology**, v. 22, n. 12, p. 2675–2686, 2011.

ARCH-INT, N. et al. Graph-Based Semantic Web Service Composition for Healthcare Data Integration. **Journal of Healthcare Engineering**, v. 2017, p. 1–19, 2017.

ARMBRUST, M. et al. A View of Cloud Computing Clearing the clouds away from the true potential and obstacles posed by this computing capability. **Communications of the ACM**, v. 53, n. 4, p. 50–58, 2010.

ASHBURNER, M. et al. Gene Ontology: tool for the unification of biology. **Nature Genetics**, v. 25, n. 1, p. 25–29, maio 2000.

BAKER, R.; ISOTANI, S.; CARVALHO, A. Mineração de Dados Educacionais:

Oportunidades para o Brasil. **Revista Brasileira de Informática na Educação**, v. 19, n. 02, p. 03, 2011.

BARISEVIČIUS, G. et al. Supporting Digital Healthcare Services Using Semantic Web Technologies. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l: s.n.], 2018. v. 11137 LNCS. p. 291–306.

BARRETO, R. G.; AVERSARI, L. O. C.; et al. Clinical Decision Support Based on OWL Queries in a Knowledge-as-a-Service Architecture. **Rules and Reasoning. RuleML+RR 2018. Lecture Notes in Computer Science**. [S.l.]: Springer, Cham, 2018a. p. 226–238.

BARRETO, R. G.; AVERSARI, L.; et al. H-KaaS: A Knowledge-as-a-Service architecture for E-health. **Brazilian Journal of Biological Sciences**, v. 5, n. 9, p. 3–12, 30 abr. 2018b.

BARRETO, R. G. et al. Utilizando Redes Neurais Artificiais para o Diagnóstico de Câncer Cervical. **Revista Saúde & Ciência Online**, ISSN 2317-8469, v. 7, n. 2, p. 59–67, 2019.

BARRETO, R. G.; MARINHO, G. M. G. A.; et al. Utilizando Redes Neurais Artificiais Para O Diagnóstico de Câncer Cervical. 2018, João Pessoa, PB: Associação dos Portadores de Epilepsia do Estado da Paraíba, 2018. p. 246.

BARROS, E.; GONÇALVES, L. F. **Nefrologia: Série No Consultório**. [S.l.]: Artmed Editora, 2009.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. [S.l.]: Addison-Wesley Professional, 2003.

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. **Jornal Brasileiro de Nefrologia**, v. 33, n. 1, p. 93–108, mar. 2011.

BEEMAN, D. **Multi-layer perceptrons and back propagation**. Disponível em: <http://ecee.colorado.edu/~ecen4831/lectures/NNet3.html#backprop>. Acesso em: 29 jun. 2019a.

_____. **Neural Network Examples and Demonstrations**. Disponível em: <http://ecee.colorado.edu/~ecen4831/lectures/NNdemo.html>. Acesso em: 15 jun. 2019b.

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, 2011.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, v. 13, n. Feb, p. 281–305, 2012.

BERNER, E. S. **Clinical decision support systems**. [S.l.]: Springer, 2007. v. 233.

BLOCH, J. How to design a good API and why it matters. 2006, [S.l: s.n.], 2006. p. 506–507.

BRACHMAN, R. J.; LEVESQUE, H. J. **Knowledge Representation and Reasoning**. [S.l: s.n.], 2004. v. 1.

BRAGA, A. de P.; CARVALHO, A. P. de L. F. de; LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações**. 2ª Edição ed. [S.l.]: LTC - Livros Técnicos e Científicos Editora S.A., 2000.

BRASIL, L. M. Informática em Saúde. **Informática em Saúde**. [S.l: s.n.], 2008. .

BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; et al. API design for machine learning software: experiences from the scikit-learn project. **CoRR**, v. abs/1309.0, 2013.

BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B. **RobustScaler: Scikit-Learn Documentation**. Disponível em: <http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html>. Acesso em: 7 abr. 2018.

CAMPOS, S. P. R. **Sistema para raciocínio semântico no domínio dos hospitais de João Pessoa**. 2013. Universidade Federal da Paraíba, João Pessoa, PB, 2013.

CHAIM, R. M.; OLIVEIRA, E. C.; ARAUJO, A. P. F. Technical specifications of a service-oriented architecture for semantic interoperability of EHR — electronic health records. jun. 2017, [S.l.]: IEEE, jun. 2017. p. 1–6.

CHANG, Y. S. et al. Mobile cloud-based depression diagnosis using an ontology and a Bayesian network. **Future Generation Computer Systems**, v. 43–44, p. 87–98, fev. 2015.

CHEN, H. Architecture strategies and data models of Software as a Service: A review. 2016, [S.l: s.n.], 2016. p. 382–385.

CHEN, M. et al. Disease Prediction by Machine Learning over Big Data from Healthcare Communities. **IEEE Access**, v. 5, p. 8869–8879, 2017.

CHOLLET, F. **Keras: The Python Deep Learning library**. **Keras.Io**. [S.l.]: Keras.io. , 2015

COIERA, E. **Guide to Health Informatics, Third Edition**. London: CRC Press, 2015.

CONSELHO FEDERAL DE MEDICINA. **Demografia médica no Brasil (Vol. 2): Cenários e indicadores de distribuição**. . São Paulo: [s.n.], 2013.

CORCHO, O.; GÓMEZ-PÉREZ, A. A roadmap to ontology specification languages. 2000, [S.l: s.n.], 2000. p. 80–96.

COSTA, E.; COSTA, C.; SANTOS, M. Y. Efficient Big Data modelling and organization for Hadoop hive-based data warehouses. 2017, [S.l: s.n.], 2017. p. 3–16.

CROCKFORD, D. The application/json media type for javascript object notation (json).

2006.

DAGHER, G. G. et al. SecDM: privacy-preserving data outsourcing framework with differential privacy. **Knowledge and Information Systems**, p. 1–38, 2019.

DANTAS, B. L. et al. Sistemas de Apoio à Decisão Médica: Uma Inovação na Medicina Oncológica. 2018, João Pessoa, PB: Associação dos Portadores de Epilepsia do Estado da Paraíba, 2018. p. 221.

_____. Sistemas de Apoio à Decisão Médica: Uma Inovação na Medicina Oncológica. **Ciências da Saúde: Da Teoria à Prática 10**. [S.l.]: Atena Editora, 2019. p. 255–262.

DENG, Z.; WANG, Z.; WANG, S. Stochastic area pooling for generic convolutional neural network. **Frontiers in Artificial Intelligence and Applications**, v. 285, p. 1760–1761, 2016.

DOGMUS, Z.; ERDEM, E.; PATOGLU, V. RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud. **Robotics and Computer-Integrated Manufacturing**, v. 33, p. 100–109, 1 jun. 2015.

DROPBOX. **Dropbox: Seus documentos em qualquer lugar**. Disponível em: <https://www.dropbox.com/pt_BR/>. Acesso em: 26 set. 2018.

ELSEVIER, S. Scopus: Content coverage guide. **Amesterdam: Elsevier BV**, 2017.

ESFANDIARI, N. et al. Knowledge discovery in medicine: Current issue and future trend. **Expert Systems with Applications**, v. 41, n. 9, p. 4434–4463, 2014.

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.

FATTAH, S. M. M.; CHONG, I. Restful web services composition using semantic ontology for elderly living assistance services. **Journal of Information Processing Systems**, v. 14, n. 4, p. 1010–1032, 2018.

FERNANDES, K.; CARDOSO, J. S.; FERNANDES, J. Transfer learning with partial observability applied to cervical cancer screening. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.]: Springer, Cham, 2017. v. 10255 LNCS. p. 243–250.

FOLK, M. et al. An overview of the HDF5 technology suite and its applications. 2011, [S.l: s.n.], 2011. p. 36–47.

FOX, A.; PATTERSON, D. A.; JOSEPH, S. **Engineering software as a service: an agile approach using cloud computing**. [S.l.]: Strawberry Canyon LLC, 2014.

FOX, J. et al. OpenClinical.net: A platform for creating and sharing knowledge and promoting best practice in healthcare. **Computers in Industry**, v. 66, p. 63–72, 1 jan. 2015.

FRAKES, W. B.; BAEZA-YATES, R. **Information retrieval: Data structures and algorithms**. [S.l.]: Prentice-Hall, Inc., 1992.

FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. Knowledge Discovery in Databases: An Overview. **AI Magazine**, v. 13, n. 3, p. 57–70, 1992.

FREMANTLE, P.; WEERAWARANA, S.; KHALAF, R. Enterprise services. **Communications of the ACM**, v. 45, n. 10, p. 77–82, 1 out. 2002.

GAI, K. et al. Ontology-Based Knowledge Representation for Secure Self-Diagnosis in Patient-Centered Teleheath with Cloud Systems. nov. 2015, [S.l.]: IEEE, nov. 2015. p. 98–103.

GALLAGHER, B. P. **Using the architecture tradeoff analysis methodsm to evaluate a reference architecture: a case study**. . [S.l: s.n.], 2000.

GARCÍA-LAENCINA, P. J.; SANCHO-GÓMEZ, J.-L.; FIGUEIRAS-VIDAL, A. R. Pattern classification with missing data: a review. **Neural Computing and Applications**, v. 19, n. 2, p. 263–282, 2010.

GEURTS, P.; IRRTHUM, A.; WEHENKEL, L. **Supervised learning with decision tree-based methods in computational and systems biology**. **Molecular BioSystems**. [S.l.]: The Royal Society of Chemistry. , 12 nov. 2009

GLIMM, B. et al. HermiT: an OWL 2 reasoner. **Journal of Automated Reasoning**, v. 53, n. 3, p. 245–269, 2014.

GOLDSBOROUGH, P. A Tour of TensorFlow. **CoRR**, v. abs/1610.0, 2016.

GOMES, C. N. A. P. et al. OntoDRC: Prevenindo a Doença Renal Crônica. 2017, [S.l: s.n.], 2017. p. 56–62.

GOOGLE. **Google Drive: Todos os seus arquivos, sempre que você precisar**. Disponível em: <https://www.google.com/intl/pt-BR/drive/>. Acesso em: 26 out. 2018.

GREENES, R. A. **Clinical decision support: the road ahead**. [S.l.]: Elsevier, 2011.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge acquisition**, v. 5, n. 2, p. 199–220, 1993.

_____. **What is an ontology?** Disponível em: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. Acesso em: 23 nov. 2017.

GYAWALI, B. Does global oncology need artificial intelligence? **The Lancet Oncology**, v. 19, n. 5, p. 599–600, 2018.

HAMOUDA, I. Ben; TANTAN, O. C.; BOUGHZALA, I. Towards an Ontological Framework for Knowledge Sharing in Healthcare Systems. **Pacific Asia Conference on Information Systems (PACIS)**, p. 1–8, 2016.

HAN, J.; PEI, J.; KAMBER, M. **Data mining: concepts and techniques**. [S.l.]: Elsevier, 2011.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. [S.l.]: Prentice Hall, 1994.

HERSH, W. R.; HOYT, R. E. **Health Informatics: Practical Guide Seventh Edition**. [S.l.]: Lulu. com, 2018.

HORRIDGE, M.; BECHHOFER, S. The OWL API: A Java API for OWL ontologies. **Semantic Web**, v. 2, n. 1, p. 11–21, 2011.

HORRIDGE, M.; BECHHOFER, S.; NOPPENS, O. Igniting the OWL 1.1 touch paper: The OWL API. **CEUR Workshop Proceedings**, v. 258, 2007.

HOYT, R. E. **Medical informatics: Practical guide for the healthcare professional**. [S.l.]: University of West Florida, School of Allied Health and Life Sciences, 2009.

HUANG, G. Bin; ZHU, Q. Y.; SIEW, C. K. Extreme learning machine: Theory and applications. **Neurocomputing**, v. 70, n. 1–3, p. 489–501, 2006.

INSTITUTO NACIONAL DE CÂNCER. **Detecção precoce: Ações de Controle do Câncer do Colo do Útero**. Disponível em: <https://www.inca.gov.br/controle-do-cancer-do-colo-do-utero/acoes-de-controle/deteccao-precoce>. Acesso em: 2 maio 2019.

_____. **Diretrizes brasileiras para o rastreamento do câncer do colo do útero**. 2ª edição ed. Rio de Janeiro, RJ: Ministério da Saúde, 2016.

_____. **Estimativa 2018-Incidência de câncer no Brasil**. Rio de Janeiro, RJ: Ministério da Saúde, 2017.

ISO/IEC/IEEE 42010. **Systems and software engineering — Architecture description**. . [S.l.]: International Organization for Standardization. , 2011

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The unified software development process**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1999.

JAYALAKSHMI, T.; SANTHAKUMARAN, a. Statistical normalization and back propagation for classification. **International Journal of Computer …**, v. 3, n. 1, p. 1–5, 2011.

KARLIK, B.; OLGAC, V. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks. **International Journal of Artificial Intelligence And Expert Systems (IJAE)**, v. 1, n. 4, p. 111–122, 2011.

KAWAMOTO, K. Standards for Scalable Clinical Decision Support: Need, Current and Emerging Standards, Gaps, and Proposal for Progress. **The Open Medical Informatics Journal**, v. 4, n. 1, p. 235–244, 2012.

KHAMPARIA, A.; PANDEY, B. Comprehensive analysis of semantic web reasoners and tools: a survey. **Education and Information Technologies**, v. 22, n. 6, p. 3121–3145, 2017.

KITCHENHAM, B. et al. **Systematic literature reviews in software engineering - A systematic literature review**. **Information and Software Technology**. [S.l: s.n.]. , 2009

KOLYVAKIS, P.; YOO, M.-J.; KIRITSIS, D. Knowledge as a service in the IoT era. 2017, [S.l: s.n.], 2017. p. 1–6.

KRISHNASWAMY, S.; LOKE, S. W.; 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)**. [S.l: s.n.], 2001. v. 2176. p. 120–134.

KUHN, M.; JOHNSON, K. **Applied predictive modeling**. [S.l.]: Springer, 2013. v. 26.

LADEIRA, M. **Representação de conhecimento e redes de decisão**. 1997. 150 f. Universidade Federal Do Rio Grande Do Sul, Porto Alegre, 1997.

LEE, H. J.; SOHN, M. Health Service Knowledge Management to Support Medical Group Decision Making. jul. 2016, [S.l.]: IEEE, jul. 2016. p. 409–414.

LEVEY, A. S. et al. A simplified equation to predict glomerular filtration rate from serum creatinine. **Journal of The American Society of Nephrology**, v. 11, 2000.

LOBO, L. C. Inteligência Artificial e Medicina. **Revista Brasileira de Educação Médica**, v. 41, n. 2, p. 185–193, 2017.

MAIMON, O.; ROKACH, L. Introduction to Knowledge Discovery and Data Mining. **Data Mining and Knowledge Discovery Handbook**. Boston, MA: Springer US, 2010. p. 1–15.

MANASHTY, A.; LIGHT, J.; YADAV, U. Healthcare event aggregation lab (HEAL), a knowledge sharing platform for anomaly detection and prediction. out. 2015, [S.l.]: IEEE, out. 2015. p. 648–652.

MARINHO, G. M. G. A. et al. A Inteligência Artificial auxiliando no diagnóstico precoce de Câncer de Mama. 2018, João Pessoa, PB: Associação dos Portadores de Epilepsia do Estado da Paraíba, 2018. p. 17.

MARTÍNEZ-FERNÁNDEZ, S. et al. Benefits and drawbacks of reference architectures. 2013, [S.l: s.n.], 2013. p. 307–310.

MASSE, M. **REST API Design Rulebook**. [S.l.]: O'Reilly Media, Inc., 2011.

MASSOUD, M. M. Y.; IKRAM, R. Ontology and Knowledge Sharing in E-Health. out. 2015, [S.l.]: IEEE, out. 2015. p. 446–446.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, 1943.

MEGA. **MEGA: Comunicação e armazenamento em nuvem seguros**. Disponível em: <https://mega.nz/>. Acesso em: 2 jun. 2019.

MICHALSKI, R. S.; CHILAUSKY, R. L. Knowledge acquisition by encoding expert rules versus computer induction from examples: a case study involving soybean pathology. 1980.

MICROSOFT. **Microsoft Application Architecture Guide (Patterns & Practices)**. 2nd Editio ed. [S.l.]: Microsoft Press, 2009.

MINISTÉRIO DA SAÚDE. **Diretrizes Clínicas para o Cuidado ao paciente com Doença**

Renal Crônica – DRC no Sistema Único de Saúde/ Ministério da Saúde. Brasília: Ministério da Saúde, 2014.

_____. **Manual Técnico Operacional do Sistema De Informação Hospitalar**. Brasília: [s.n.], 2010.

_____. **Sistema de Controle de Exames Laboratoriais da Rede Nacional de Contagem de Linfócitos CD4+/CD8+ e Carga Viral do HIV (SISCEL)**. Disponível em: <http://www.aids.gov.br/pt-br/sistema-de-informacao/sistema-de-controle-de-exames-laboratoriais-da-rede-nacional-de-contagem-de>. Acesso em: 17 jun. 2020a.

_____. **Sistema de Informação de Agravos de Notificação – SINAN: Normas e Rotinas**. 2ª edição ed. Brasília: Secretaria de Vigilância em Saúde, Departamento de Vigilância Epidemiológica., 2007.

_____. **Sobre o DataSUS**. Disponível em: <https://datasus.saude.gov.br/sobre-o-datasus/>. Acesso em: 18 jun. 2020b.

MISHRA, R. B.; KUMAR, S. Semantic web reasoners and languages. **Artificial Intelligence Review**, v. 35, n. 4, p. 339–368, 2011.

MITRPANONT, J. et al. K4ThaiHealth: A Prototype for Thai Routine Medical Research Knowledge Extraction Sharing. jul. 2018, [S.l.]: IEEE, jul. 2018. p. 1–6.

MOHER, D. et al. Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. **PLoS Medicine**, v. 6, n. 7, p. e1000097, 21 jul. 2009.

MOREIRA, A.; ALVARENGA, L.; OLIVEIRA, A. de P. O nível do conhecimento e os instrumentos de representação: tesauros e ontologias. **DataGramaZero-Revista de Ciência da Informação**, v. 5, n. 6, p. 1–25, 2004.

NEWELL, A.; SIMON, H. A. Computer science as empirical inquiry: Symbols and search. **ACM Turing award lectures**. [S.l: s.n.], 2007. p. 1975.

OASIS STANDARD. **eXtensible Access Control Markup Language (XACML) Version 3.0**. . [S.l: s.n.]. , 2013

PAPAZOGLOU, M. P.; VAN DEN HEUVEL, W. J. Service oriented architectures: Approaches, technologies and research issues. **VLDB Journal**, v. 16, n. 3, p. 389–415, 2007.

PEARL, J. Evidential reasoning using stochastic simulation of causal models. **Artificial Intelligence**, v. 32, n. 2, p. 245–257, 1 maio 1987.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, n. Oct, p. 2825–2830, 2012.

PENG, C.; GOSWAMI, P.; BAI, G. Linking Health Web Services as Resource Graph by Semantic REST Resource Tagging. **Procedia Computer Science**, v. 141, p. 319–326, 2018.

PERAL, J. et al. An ontology-oriented architecture for dealing with heterogeneous data applied to telemedicine systems. **IEEE Access**, v. 6, p. 41118–41138, 2018.

RAFIQUE, A. et al. Towards scalable and dynamic data encryption for multi-tenant saas. 2017, [S.l: s.n.], 2017. p. 411–416.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. Barueri, SP: Editora Manole Ltda, 2003.

RODRIGUES, K. E.; CAMARGO, B. de. Diagnóstico Precoce do Câncer Infantil: Responsabilidade de Todos. **Rev Assoc Med Bras**, v. 49, n. 1, p. 29–34, 2003.

ROMÃO JUNIOR, J. E. Doença Renal Crônica: Definição, Epidemiologia e Classificação. **Jornal Brasileiro de Nefrologia**, v. 26, n. 1, p. 1–3, 2004.

ROSSUM, G. Van; DRAKE, F. L. **Python Tutorial**. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 2010. v. 42.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. 3rd Editio ed. [S.l.]: Malaysia; Pearson Education Limited, 2013.

SACHDEVA, S.; BHALLA, S. Semantic interoperability in standardized electronic health record databases. **Journal of Data and Information Quality**, v. 3, n. 1, p. 1–37, 2012.

SATTLER, U. et al. Description Logics Handbook. **The Description Logics Handbook: Theory, Implementation and Applications**, v. 18, p. 142, 2003.

SCHMIDHUBER, J. Deep Learning in neural networks: An overview. **Neural Networks**, v. 61, p. 85–117, 2015.

SCHMIDT, D. C. et al. **Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects**. [S.l.]: John Wiley & Sons, 2013. v. 2.

SENEVIRATNE, O. et al. Enabling Trust in Clinical Decision Support Recommendations through Semantics. 2019, Auckland, New Zealand: [s.n.], 2019.

SETHURAMAN, R.; SNEHA, G.; SWETHA BHARGAVI, D. A semantic web services for medical analysis in health care domain. 2017, [S.l.]: IEEE, 2017. p. 1–5.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014.

SHANG, Y. et al. Development of a Service-Oriented Sharable Clinical Decision Support System Based on Ontology for Chronic Disease. **Studies in health technology and informatics**, v. 245, p. 1153–1157, 2017.

SHEARER, R.; MOTIK, B.; HORROCKS, I. HermiT: A Highly-Efficient OWL Reasoner. 2008, [S.l: s.n.], 2008. p. 91.

SILVA, E. G. da. **Plataforma para orquestração de serviços para cuidados continuados de saúde**. 2019. Universidade Federal de Goiás, 2019.

SILVA, R. **Modelo de apoio ao diagnóstico no domínio médico: aplicando raciocínio baseado em casos**. 2005. 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**, v. 8, n. 6, p. 527–534, 1 nov. 2001.

SIRIN, E. et al. Pellet: A practical owl-dl reasoner. **Journal of Web Semantics**, v. 5, n. 2, p. 51–53, 2007.

SIRIN, E.; PARSIA, B. Pellet: An OWL DL reasoner. **CEUR Workshop Proceedings**, v. 104, p. 1–2, 2004.

SIU, P. K. Y. et al. An Intelligent Clinical Decision Support System for Assessing the Needs of a Long-Term Care Plan. **Advances in Intelligent and Personalized Clinical Decision Support Systems**. [S.l.]: IntechOpen, 2019. .

SOCIEDADE BRASILEIRA DE NEFROLOGIA. **Censo de Diálise: SBN 2013**. Disponível em: <http://arquivos.sbn.org.br/pdf/censo_2013_publico_leigo.pdf>. Acesso em: 26 nov. 2016.

_____. **O que é Nefrologia?** Disponível em: <https://sbn.org.br/publico/institucional/o-que-e-nefrologia/>. Acesso em: 2 jun. 2019.

SØILEN, K. S. Users' perceptions of Data as a Service (DaaS). **Journal of Intelligence Studies in Business**, v. 6, n. 2, 2016.

SOMMERVILLE, I. **Engenharia de software**. 8. ed. São Paulo: Addison Wesley, 2007.

SOUFFRONT, K. et al. Integrating a clinical decision support reminder to improve blood pressure reassessment for patients with uncontrolled hypertension. **Clinical Cardiology and Cardiovascular Interventions**, v. 2, 2019.

SOURI, A.; ASGHARI, P.; REZAEI, R. Software as a service based CRM providers in the cloud computing: challenges and technical issues. **Journal of Service Science Research**, v. 9, n. 2, p. 219–237, 2017.

SOUZA, C. A. de et al. Rede Bayesiana e Ontologia: Uma Abordagem no Domínio da Nefrologia. 2018, Fortaleza, CE: Sociedade Brasileira de Informática em Saúde (SBIS), 2018. p. 961–974.

SRIVASTAVA, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014.

STUNTEBECK, E. P. et al. HealthSense: Classification of health-related sensor data through user-assisted machine learning. 2008, New York, New York, USA: ACM Press, 2008. p. 1.

SUN, R.; MERRILL, E.; PETERSON, T. From implicit skills to explicit knowledge: A bottom-up model of skill learning. **Cognitive science**, v. 25, n. 2, p. 203–244, 2001.

SWAGGER. **Swagger Editor**. Disponível em: <https://swagger.io/tools/swagger-editor/>. Acesso em: 1 mar. 2018.

TAVARES, E. A. **Uma abordagem para suporte à decisão clínica baseada em semântica**

no domínio da nefrologia. 2016. 226 f. Universidade Federal da Paraíba, 2016.

THE APACHE SOFTWARE FOUNDATION. **Apache Jena: Jena Ontology API**. Disponível em: <https://jena.apache.org/documentation/ontology/>. Acesso em: 4 jan. 2018.

TSAFARA, A. et al. Cloud-Based Data and Knowledge Management for Multi-Centre Biomedical Studies. 2015, New York, New York, USA: ACM Press, 2015. p. 1–4.

VALENTIM, M. L. P. Inteligência competitiva em organizações: dado, informação e conhecimento. **DataGramaZero - Revista de Ciência da Informação**, v. 3, n. 4, p. 1–13, 2002.

VEINOT, T. C. et al. Leveling up: on the potential of upstream health informatics interventions to enhance health equity. **Medical care**, v. 57, p. S108--S114, 2019.

WANG, L. et al. Scientific Cloud Computing: Early Definition and Experience. set. 2008, [S.l.]: IEEE, set. 2008. p. 825–830.

WECHSLER, R. et al. A informática no consultório médico. **Jornal de Pediatria**, v. 79, p. S3–S12, 2003.

WITTEN, I. H. et al. **Data Mining: Practical Machine Learning Tools and Techniques**. 3rd Editio ed. [S.l.]: Elsevier, 2011.

WONG, S. C. et al. Understanding Data Augmentation for Classification: When to Warp? nov. 2016, [S.l.]: IEEE, nov. 2016. p. 1–6.

WORLD HEALTH ORGANIZATION. **Palliative Care Knowledge into Action Cancer Control WHO Guide for Effective Programmes**. . [S.l: s.n.], 2007.

WORLD WIDE WEB CONSORTIUM et al. **Resource description framework (RDF) schema specification**. Disponível em: <https://www.w3.org/TR/PR-rdf-syntax/>. Acesso em: 20 abr. 2020.

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

YU, T. et al. Research on the construction of knowledge service platform for TCM health preservation. out. 2017, [S.l.]: IEEE, out. 2017. p. 1–6.

ZGHEIB, R.; CONCHON, E.; BASTIDE, R. Engineering IoT Healthcare Applications: Towards a Semantic Data Driven Sustainable Architecture. **Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST**. [S.l: s.n.], 2017. v. 181 LNICST. p. 407–418.

ZHANG, Y. F. et al. Design and Development of a Sharable Clinical Decision Support System Based on a Semantic Web Service Framework. **Journal of Medical Systems**, v. 40, n. 5, p. 118, 22 maio 2016.

ZHAO, Z. et al. Knowledge-as-a-Service: A community knowledge base for research infrastructures in environmental and earth sciences. 2019, [S.l: s.n.], 2019. p. 127–132.

ZHOU, Z. H. et al. Lung cancer cell identification based on artificial neural network ensembles. **Artificial Intelligence in Medicine**, v. 24, n. 1, p. 25–36, 1 jan. 2002.

APÊNDICE A: CONSULTAS EXECUTADAS NOS BUSCADORES

IEEE

URL: https://ieeexplore.ieee.org/search/advsearch.jsp?expression-builder

```
((("Document Title":"kaas" OR "Document Title":"knowledge as a service" OR "Document Title":"knowledge-as-a-service" OR "Document Title":"kgaas" OR "Document Title":"knowledge graph as a service" OR "Document Title":"knowledge-graph-as-a-service") OR (("Document Title":"knowledge" OR "Document Title":"ontology" OR "Document Title":"ontologies" OR "Document Title":"semantic") AND ("Document Title":"architecture" OR "Document Title":"paradigm" OR "Document Title":"service" OR "Document Title":"cloud" OR "Document Title":"sharing" OR "Document Title":"share"))) AND ("health" OR "e-health" OR "health domain" OR "health care" OR "healthcare" OR "medicine"))
```

ACM

URL: https://dl.acm.org/advsearch.cfm?coll=DL&dl=ACM

```
(((acmdlTitle:"kaas" OR acmdlTitle:"knowledge as a service" OR acmdlTitle:"knowledge-as-a-service" OR acmdlTitle:"kgaas" OR acmdlTitle:"knowledge graph as a service" OR acmdlTitle:"knowledge-graph-as-a-service") OR ((acmdlTitle:"knowledge" OR acmdlTitle:"ontology" OR acmdlTitle:"ontologies" OR acmdlTitle:"semantic") AND (acmdlTitle:"architecture" OR acmdlTitle:"paradigm" OR acmdlTitle:"service" OR acmdlTitle:"cloud" OR acmdlTitle:"sharing" OR acmdlTitle:"share"))) AND ("health" OR "e-health" OR "health domain" OR "health care" OR "healthcare" OR "medicine"))
```

PUBMED

URL: https://www.ncbi.nlm.nih.gov/pubmed/advanced

```
(("kaas"[Title] OR "knowledge as a service"[Title] OR "knowledge-as-a-service"[Title] OR (("knowledge"[Title] OR "knowledge"[Title]) AND graph[Title] AND service[Title])) OR (("knowledge"[Title] OR "ontology"[Title] OR "ontologies"[Title] OR "semantic"[Title]) AND ("architecture"[Title] OR "paradigm"[Title] OR "service"[Title] OR "cloud"[Title] OR "sharing"[Title] OR "share"[Title]))) AND ("health"[All Fields] OR "e-health"[All Fields] OR "health domain"[All Fields] OR "health care"[All Fields] OR "healthcare"[All Fields] OR "medicine"[All Fields])
```

SCORPUS

URL: https://www.scopus.com/search/form.uri?display=basic

```
(((TITLE("kaas") OR TITLE("knowledge as a service") OR TITLE("knowledge-as-a-service") OR TITLE("kgaas") OR TITLE("knowledge graph as a service") OR TITLE("knowledge-graph-as-a-service")) OR ((TITLE("knowledge") OR TITLE("ontology") OR TITLE("ontologies") OR TITLE("semantic")) AND (TITLE("architecture") OR TITLE("paradigm") OR TITLE("service") OR TITLE("cloud") OR TITLE("sharing") OR TITLE("share")))) AND (TITLE-ABS-KEY("health") OR TITLE-ABS-KEY("e-health") OR TITLE-ABS-KEY("health domain") OR TITLE-ABS-KEY("health care") OR TITLE-ABS-KEY("healthcare") OR TITLE-ABS-KEY("medicine"))) AND (LIMIT-TO(LANGUAGE,"English")) AND (LIMIT-TO(SUBJAREA,"MEDI") OR LIMIT-TO (SUBJAREA,"COMP"))
```

APÊNDICE B: NÚMERO DE ARTIGOS PUBLICADOS EM CADA BANCO DE DADOS

ANO	BUSCADOR				TOTAL
	IEEE	ACM	SCOPUS	PUBMED	
1970	0	0	0	0	0
1971	0	0	0	0	0
1972	0	0	1	0	1
1973	0	0	1	0	1
1974	0	0	1	0	1
1975	0	0	0	0	0
1976	0	0	0	2	2
1977	0	0	1	2	3
1978	0	0	0	0	0
1979	0	0	3	0	3
1980	0	0	0	0	0
1981	0	0	2	1	3
1982	0	0	0	0	0
1983	0	0	1	0	1
1984	0	0	0	0	0
1985	0	0	2	0	2
1986	0	0	2	0	2
1987	0	0	2	0	2
1988	0	0	1	0	1
1989	0	0	3	0	3
1990	1	0	1	1	3
1991	0	0	1	1	2
1992	0	0	2	1	3
1993	0	0	3	2	5
1994	0	0	12	3	15
1995	0	0	4	2	6
1996	1	0	4	2	7
1997	0	0	6	4	10
1998	0	0	14	4	18
1999	1	1	5	4	11
2000	1	0	8	3	12
2001	0	0	9	3	12
2002	0	0	7	4	11
2003	2	0	16	8	26
2004	0	1	14	7	22
2005	6	0	15	10	31
2006	3	1	33	10	47
2007	0	2	28	11	41

2008	12	2	38	18	70
2009	9	4	27	12	52
2010	8	2	49	17	76
2011	10	2	38	11	61
2012	11	5	42	22	80
2013	9	1	56	23	89
2014	4	2	52	26	84
2015	16	2	63	17	98
2016	4	2	58	18	82
2017	5	0	52	18	75
2018	6	4	66	28	104
2019	1	0	18	10	29
TOTAL	110	31	761	305	1207

APÊNDICE C: MODELO DE FORMULÁRIO USADO PARA AVALIAR TÍTULOS, ABSTRACTS E PALAVRAS-CHAVE

BUSCADOR	TÍTULO DA PUBLICAÇÃO	ANO	RESULTADO
ACM	Cloud-Based Data and Knowledge Management for Multi-Centre Biomedical Studies	2015	
Resumo: We have developed a cloud-based (AWS and IBM SoftLayer) knowledge environment for scalable semantic mining of scientific literature and PTM integrative knowledge discovery in precision medicine, building upon our novel natural language processing (NLP) technologies and bioinformatics infrastructure. We provided semantic integration of full-scale PubMed mining results from disparate text mining tools, along with kinase-substrate data from iPTMnet, and PTM proteoforms and their relations from Protein Ontology (PRO). We shared the digital objects of those applications in multiple interoperable formats and have registered them in bioCADDIE using CEDAR. We experimented with multiple system setups using operating system, programming language, web server, or database server that best fits each application. We evaluated the cost effectiveness of cloud computing by only paying for what we use and readily experimenting with additional services. A web portal is available for accessing our cloud-based knowledge environment at https://proteininformationresource.org/cloud/. **Palavras-Chave:** Artificial intelligence; Knowledge representation and reasoning; Health care information systems; Life and medical sciences.			
IEEE	Knowledge framework for clinical processes architecture and analysis	2015	
Resumo: Last decades have introduced different improvements into healthcare information systems domain including features like open interfaces to clinical facilities, improved patients registers, connectivity to insurance companies, and clinical pathways and workflows. The last named - clinical pathways and workflows - is a part of a complex area known as process modeling and architecture and except the control and management it also offers additional advantages like process optimization, re-engineering, analysis, and automatized process execution. This paper discusses application of explicit knowledge profiles based on process meta-model within clinical processes, alignment with visual process modeling, and further analysis with simulation and reverse engineering methods. **Palavras-Chave:** Unified modeling language; Analytical models; Medical services; Adaptation models; OWL; Reverse engineering; Mathematical model.			
Scopus	Restful web services composition using semantic ontology for elderly living assistance services	2018	
Resumo: Recent advances in medical science have made people live longer, which has affected many aspects of life, such as caregiver burden, increasing cost of healthcare, increasing number of disabled and depressive disorder persons, and so on. Researchers are now focused on elderly living assistance services in smart home environments. In recent years, assisted living technologies have rapidly grown due to a faster growing aging society. Many smart devices are now interconnected within the home network environment and such a home setup supports collaborations between those devices based on the Internet of Things (IoT). One of the major challenges in providing elderly living assistance services is to consider each individual's requirements of different needs. In order to solve this, the virtualization of physical things, as well as the collaboration and composition of services provided by these physical things should be considered. In order to meet these challenges, Web of Objects (WoO) focuses on the implementation aspects of IoT to bring the assorted real world objects with the web applications. We proposed a semantic modelling technique for manual and semiautomated service composition. The aim of this work is to propose a framework to enable RESTful web services composition using semantic ontology for elderly living assistance services creation in WoO based smart home environment. **Palavras-Chave:** Internet of Things; Service composition; Web of objects.			

APÊNDICE D: ARTIGOS INCLUÍDOS NA REVISÃO DA LITERATURA APÓS A AVALIAÇÃO DOS TÍTULOS, ABSTRACTS E KEYWORDS

BUSCADOR	TÍTULO DA PUBLICAÇÃO	ANO	AVAL. 1	AVAL. 2	RESULTADO
ACM	Cloud-Based Data and Knowledge Management for Multi-Centre Biomedical Studies	2015	INCLUIR	INCLUIR	INCLUIR
IEEE	Research on the construction of knowledge service platform for TCM health preservation	2017	INCLUIR	INCLUIR	INCLUIR
IEEE	Technical specifications of a service-oriented architecture for semantic interoperability of EHR — electronic health records	2017	INCLUIR	INCLUIR	INCLUIR
IEEE	An Ontology-Oriented Architecture for Dealing With Heterogeneous Data Applied to Telemedicine Systems	2018	INCLUIR	INCLUIR	INCLUIR
IEEE	K4ThaiHealth: A Prototype for Thai Routine Medical Research Knowledge Extraction Sharing	2018	INCLUIR	INCLUIR	INCLUIR
IEEE	A semantic smart interconnected healthcare system using ontology and cloud computing	2018	INCLUIR	INCLUIR	INCLUIR
IEEE	Healthcare event aggregation lab (HEAL), a knowledge sharing platform for anomaly detection and prediction	2015	INCLUIR	INCLUIR	INCLUIR
PubMed	Development of a Service-Oriented Sharable Clinical Decision Support System Based on Ontology for Chronic Disease	2017	INCLUIR	INCLUIR	INCLUIR
PubMed	Design and Development of a Sharable Clinical Decision Support System Based on a Semantic Web Service Framework	2016	INCLUIR	INCLUIR	INCLUIR
Scopus	Clinical decision support based on OWL queries in a knowledge-as-a-service architecture	2018	INCLUIR	INCLUIR	INCLUIR
Scopus	Restful web services composition using semantic ontology for elderly living assistance services	2018	EXCLUIR	INCLUIR	INCLUIR
Scopus	Supporting digital healthcare services using semantic web technologies	2018	INCLUIR	INCLUIR	INCLUIR
Scopus	Linking health web services as resource graph by semantic REST resource tagging	2018	INCLUIR	EXCLUIR	INCLUIR
Scopus	A semantic web services for medical analysis in health care domain	2017	EXCLUIR	INCLUIR	INCLUIR
Scopus	Engineering IoT healthcare applications: Towards a semantic data driven sustainable architecture	2017	INCLUIR	EXCLUIR	INCLUIR
Scopus	Semantic interoperability in electronic health record databases: Standards, architecture and e-health systems	2017	EXCLUIR	INCLUIR	INCLUIR

Scopus	Application of semantic integration methods for cross-agency Information sharing in healthcare	2017	INCLUIR	INCLUIR	INCLUIR
Scopus	Graph-Based Semantic Web Service Composition for Healthcare Data Integration	2017	INCLUIR	INCLUIR	INCLUIR
Scopus	Health service knowledge management to support medical group decision making	2016	INCLUIR	INCLUIR	INCLUIR
Scopus	Ontology-Based Knowledge Representation for Secure Self-Diagnosis in Patient-Centered Teleheath with Cloud Systems	2015	INCLUIR	INCLUIR	INCLUIR
Scopus	Towards an ontological framework for knowledge sharing in healthcare systems	2016	EXCLUIR	INCLUIR	INCLUIR
Scopus	A model of medical practice for contextual knowledge sharing in collaborative healthcare	2016	INCLUIR	INCLUIR	INCLUIR
Scopus	Context-aware clinical knowledge Sharing in cross-boundary e-health: A conceptual model	2015	INCLUIR	EXCLUIR	INCLUIR
Scopus	Ontology and Knowledge Sharing in E-Health	2015	EXCLUIR	INCLUIR	INCLUIR
Scopus	Mobile cloud-based depression diagnosis using an ontology and a Bayesian network	2015	INCLUIR	INCLUIR	INCLUIR
Scopus	OpenClinical.net: A platform for creating and sharing knowledge and promoting best practice in healthcare	2015	INCLUIR	INCLUIR	INCLUIR
Scopus	Towards an architecture for managing semantic knowledge in semantic repositories	2015	INCLUIR	EXCLUIR	INCLUIR
Scopus	REHABROBO-ONTO: Design, development and maintenance of a rehabilitation robotics ontology on the cloud	2015	INCLUIR	INCLUIR	INCLUIR

APÊNDICE E: TRABALHOS INCLUÍDOS NA ETAPA DE EXTRAÇÃO DE DADOS

CÓDIGO	TÍTULO DA PUBLICAÇÃO	ANO	BUSCADOR	REFERÊNCIA
TR-01	Cloud-Based Data and Knowledge Management for Multi-Centre Biomedical Studies	2015	ACM	TSAFARA et al., 2015
TR-02	A semantic smart interconnected healthcare system using ontology and cloud computing	2018	IEEE	ABATAL; KHALLOUKI; BAHAJ, 2018
TR-03	An Ontology-Oriented Architecture for Dealing With Heterogeneous Data Applied to Telemedicine Systems	2018	IEEE	PERAL et al., 2018
TR-04	Healthcare event aggregation lab (HEAL), a knowledge sharing platform for anomaly detection and prediction	2015	IEEE	MANASHTY; LIGHT; YADAV, 2015
TR-05	K4ThaiHealth: A Prototype for Thai Routine Medical Research Knowledge Extraction Sharing	2018	IEEE	MITRPANONT et al., 2018
TR-06	Research on the construction of knowledge service platform for TCM health preservation	2017	IEEE	YU et al., 2017
TR-07	Technical specifications of a service-oriented architecture for semantic interoperability of EHR — electronic health records	2017	IEEE	CHAIM; OLIVEIRA; ARAUJO, 2017
TR-08	Design and Development of a Sharable Clinical Decision Support System Based on a Semantic Web Service Framework	2016	PubMed	ZHANG et al., 2016
TR-09	Development of a Service-Oriented Sharable Clinical Decision Support System Based on Ontology for Chronic Disease	2017	PubMed	SHANG et al., 2017
TR-10	A semantic web services for medical analysis in health care domain	2017	Scopus	SETHURAMAN; SNEHA; SWETHA BHARGAVI, 2017
TR-11	Application of semantic integration methods for cross-agency Information sharing in healthcare	2017	Scopus	AKATKIN et al., 2017
TR-12	Clinical decision support based on OWL queries in a knowledge-as-a-service architecture	2018	Scopus	BARRETO; AVERSARI; et al., 2018a
TR-13	Context-aware clinical knowledge Sharing in cross-boundary e-health: A conceptual model	2015	Scopus	ANYA; TAWFIK; AL-JUMEILY, 2015
TR-14	Engineering IoT healthcare applications: Towards a semantic data driven sustainable architecture	2017	Scopus	ZGHEIB; CONCHON; BASTIDE, 2017

TR-15	Graph-Based Semantic Web Service Composition for Healthcare Data Integration	2017	Scopus	ARCH-INT et al., 2017
TR-16	Health service knowledge management to support medical group decision making	2016	Scopus	LEE; SOHN, 2016
TR-17	Linking health web services as resource graph by semantic REST resource tagging	2018	Scopus	PENG; GOSWAMI; BAI, 2018
TR-18	Mobile cloud-based depression diagnosis using an ontology and a Bayesian network	2015	Scopus	CHANG et al., 2015
TR-19	Ontology and Knowledge Sharing in E-Health	2015	Scopus	MASSOUD; IKRAM, 2015
TR-20	Ontology-Based Knowledge Representation for Secure Self-Diagnosis in Patient-Centered Teleheath with Cloud Systems	2015	Scopus	GAI et al., 2015
TR-21	OpenClinical.net: A platform for creating and sharing knowledge and promoting best practice in healthcare	2015	Scopus	FOX, J. et al., 2015
TR-22	REHABROBO-ONTO: Design, development and maintenance of a rehabilitation robotics ontology on the cloud	2015	Scopus	DOGMUS; ERDEM; PATOGLU, 2015
TR-23	Restful web services composition using semantic ontology for elderly living assistance services	2018	Scopus	FATTAH; CHONG, 2018
TR-24	Supporting digital healthcare services using semantic web technologies	2018	Scopus	BARISEVIČIUS et al., 2018
TR-25	Towards an architecture for managing semantic knowledge in semantic repositories	2015	Scopus	ALAMRI; BERTOK; FAHAD, 2015
TR-26	Towards an ontological framework for knowledge sharing in healthcare systems	2016	Scopus	HAMOUDA; TANTAN; BOUGHZALA, 2016

APÊNDICE F: DADOS EXTRAÍDOS: METADADOS

CÓDIGO	ANO	BUSCADOR	LOCAL DE PUBLICAÇÃO	TIPO	QUALIS
TR-01	2015	ACM	Proceedings of the International Conference on Knowledge Capture, K-CAP 2015	Conferência	Indisponível
TR-02	2018	IEEE	Proceedings of the 2018 International Conference on Optimization and Applications, ICOA 2018	Conferência	Indisponível
TR-03	2018	IEEE	IEEE Access	Periódico	B3
TR-04	2015	IEEE	2015 17th International Conference on E-Health Networking, Application and Services, Healthcom 2015	Conferência	B2
TR-05	2018	IEEE	Seventh ICT International Student Project Conference (ICT-ISPC)	Conferência	Indisponível
TR-06	2017	IEEE	IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)	Conferência	B2
TR-07	2017	IEEE	12th Iberian Conference on Information Systems and Technologies (CISTI)	Conferência	Indisponível
TR-08	2016	PubMed	Journal of Medical Systems	Periódico	C
TR-09	2017	PubMed	Studies in health technology and informatics	Periódico	B5
TR-10	2017	Scopus	2017 International Conference on Information Communication and Embedded Systems, ICICES 2017	Conferência	Indisponível
TR-11	2017	Scopus	Proceedings of the XXVI International Symposium on Nuclear Electronics & Computing (NEC'2017)	Conferência	Indisponível
TR-12	2018	Scopus	International Joint Conference on Rules and Reasoning	Conferência	B1
TR-13	2015	Scopus	2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing Context-aware	Conferência	B1
TR-14	2017	Scopus	Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering	Capítulo de Livro	Indisponível

TR-15	2017	Scopus	Journal of Healthcare Engineering	Periódico	Indisponível
TR-16	2016	Scopus	2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)	Conferência	A2
TR-17	2018	Scopus	Procedia Computer Science	Periódico	C
TR-18	2015	Scopus	Future Generation Computer Systems	Periódico	A2
TR-19	2015	Scopus	2015 International Conference on Healthcare Informatics	Conferência	Indisponível
TR-20	2015	Scopus	2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing	Conferência	Indisponível
TR-21	2015	Scopus	Computers in Industry	Periódico	A2
TR-22	2015	Scopus	Robotics and Computer-Integrated Manufacturing	Periódico	A2
TR-23	2018	Scopus	Journal of Information Processing Systems	Periódico	Indisponível
TR-24	2018	Scopus	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)	Periódico	C
TR-25	2015	Scopus	International Journal of Parallel, Emergent and Distributed Systems	Periódico	B1
TR-26	2016	Scopus	Pacific Asia Conference on Information Systems (PACIS)	Conferência	B1

APÊNDICE G: DADOS EXTRAÍDOS: PARADIGMA, PRINCIPAIS COMPONENTES E DOMÍNIO

CÓD.	PARADIGMA / ARQUITETURA	COMPONENTES	DOMÍNIO	REUTILISÁVEL EM OUTROS DOMÍNIOS
TR-01	Cloud based	Cloud API, Knowledge based, platform manager, study manager, user interface	Saúde	Não definido
TR-02	Não definido	Interaction Manager, Ontology Manager, Service Selector, Medical Databases, Service Repository	Saúde	Não
TR-03	Ontology-Oriented Architecture	Knowledge Base, Telemedicine system, Data Sources	Saúde, Medicina Personalizada	Não definido
TR-04	Cloud based	Service Layer, Control Layer, Distributed cloud based data providers (predictors and aggregators)	Saúde, Internet das Coisas	Não
TR-05	Não definido	ThaiDOSO, Website K4ThaiHealth	Saúde	Não
TR-06	Não definido	Knowledge Base, Knowledge Service Website, Web service interface, Knowledge Engineering platform	Saúde, Medicina Tradicional Chinesa	Não
TR-07	SOA	Barramento do serviço de saúde; Sistema de informação clínica, Framework de segurança, Framework semânticos, serviços de interoperabilidade, demográficos e repositório de informação clinica	Saúde	Não
TR-08	SOA, Semantic Web Services	Módulos de Conhecimento, Base de Conhecimento para Suporte a Decisão, Aplicativo Cliente	Saúde, Sistemas de Decisão Clínica	Não
TR-09	SOA	Knowledge Base, CDSS web service and Clients	Saúde, Tratamento de doenças crônicas	Não definido
TR-10	Não definido	Interface gráfica, Ontologias e algoritmo de aprendizagem de máquina	Saúde	Não
TR-11	Integração Semântica	Modelo de dados de domínio, Mediadores, Clientes de Serviços web	Saúde	Não definido
TR-12	SOA, Knowledge as a Service, REST	Extrator de Conhecimento; Serviço de Conhecimento; Fontes de Conhecimento; Consumidores de Conhecimento	Saúde	Não
TR-13	Practice-centered awareness model	Ontological Context, Stereotyped Context, Situated Context	Saúde	Não

TR-14	SOA, Semantic Data Driven Architecture, Message Oriented Middleware	Semantic Publisher, Semantic Middleware, Semantic message broker, Semantic Subscriber	Saúde, Internet das Coisas	Sim
TR-15	SOA, Semantic Web Service	Management-time subsystem, Run-time subsystem	Saúde	Sim
TR-16	Ontology-Oriented Architecture	Health Service ontology	Saúde	Não
TR-17	SOA, REST	Domínio conceitual, domínio de serviço, domínio de representação e domínio semântico	Geral	Sim
TR-18	Cloud-Services	User Agent, Inference agent, Recording Agent, Ontology creation agent	Saúde, Diagnóstico de Depressão	Não
TR-19	Open Grid Services Architecture	Modelo Organizacional, Modelo de conhecimento colaborativo e Arquitetura de Serviço	Saúde	Não
TR-20	Cloud-Services	Ontology set, Alignment set, parameter set and resources	Saúde	Não
TR-21	SOA, Knowledge Based Service	Reviewer database, Public Web Repository, PROforma applications	Saúde	Não
TR-22	SOA	RehabRobo-Onto, RehabRobo-query	Saúde, Robôs para Reabilitação	Não
TR-23	SOA, REST, Web of Objects	Servidor da Aplicação, Home Server, Objetos virtuais e Objetos virtuais compostos	Saúde, Internet das Coisas	Sim
TR-24	SOA, REST	Servidor de Serviço, ontologia de nível superior, Base de Conhecimento	Saúde	Não
TR-25	Semantic Knowledge	Inference Layer, Schema Layer, Lookup Layer, Data Layer	Saúde	Sim
TR-26	Não definido	Não especificado	Saúde	Não

APÊNDICE H: DADOS EXTRAÍDOS: FONTES DE CONHECIMENTO

CÓD.	FONTES DE CONHECIMENTOO	MÚLTIPLAS FONTES	COMUNICAÇÃO COM AS FONTES	FONTES EXTERNAS	COMO PODE SER ADICIONADA UMA NOVA FONTE
TR-01	Formulários e questionários criados via interface gráfica do administrador	Não	HTTP	Não	Não definido
TR-02	Ontologia, Sensores	Não	Não definido	Não	Não definido
TR-03	Internamente usa uma ontologia. Externamente as fontes são sensores, redes sociais, dados não estruturados, etc.	Sim	Processador de linguagem natural	Sim	Processamento de linguagem natural, indexação e armazenamento em ontologia de domínio e integração com a ontologia principal
TR-04	Sensores, dados brutos, algoritmos de aprendizagem de máquina	Sim	Rest, SPARQL e outros	Sim	Dados de novos sensores são agregados e os *predictors* são refinados
TR-05	Texto (Routine to Research Data), ICD10TM, MedicineNet.com	Sim	Extração de conhecimento a partir de Texto, Acesso direto aos arquivos brutos	Sim	Via processamento do texto bruto
TR-06	Knowledge base interna criada a partir de documentos, livros e experts do domínio	Não	Acesso direto	Não	Via extração semi-automatizada de conhecimento
TR-07	Diversos serviços de saúde e documentos estruturados compatíveis com o formato do barramento e OpenEHR	Sim	OpenEHR	Sim	Basta ser compatível com o barramento de serviço de saúde e OpenEHR
TR-08	Ontologias	Sim	HL7 vMR, Semantic Web Rule Language, SPARQL,	Sim	Via modularização de ontologias
TR-09	Ontologia de Domínio, Instancias dessa ontologia e Regras	Não	Jena API, SPARQL	Não	Modelagem de novo conhecimento na ontologia, criação de novas instancias
TR-10	OWL-RDF	Não	SPARQL e Jena API	Não	Não definido
TR-11	Serviços externos	Sim	SOAP	Sim	Integração com o modelo de dados de domínio
TR-12	Guidelines Médicos, Ontologias, Testes Clínicos, Literatura Médica, etc.	Sim	API Rest	Sim	Via escrita de um novo extrator de conhecimento
TR-13	Sensors, actuators, user interfaces	Sim	Não definido	Sim	Implementação de várias camadas de extração de dados e conhecimento

TR-14	Sensores Físicos, Sensores Virtuais, Sensor Semântico	Sim	OWL	Sim	Os dados brutos dos sensores são envelopados utilizando OWL
TR-15	Ontologia, Grafo de dependências, Repositório de Serviços	Sim	Jena API, OWL	Sim	Anotação manual pelo provedor do serviço e criação de grafos pelo administrador
TR-16	Health Service Ontology, Patient-context data and Doctor context data	Sim	OWL	Sim	Adição de novas ontologias
TR-17	Dados Abertos Ligados na área médica, Vocabulários médicos, Ontologias	Sim	HTTP Rest e Consultas SPARQL	Sim	Via mapeamento de termos
TR-18	Ontologia, Rede Bayesiana	Sim	OWL, Smile Library	Não	Desenvolvimento de nova ontologia e nova rede bayesiana
TR-19	Ontologias	Não	Não definido	Não especificado	Não definido
TR-20	Ontologias	Sim	Acesso direto	Não	Utilizando uma técnica chamada de semantic ontology matching, gerando uma nova fonte de conhecimento
TR-21	Guidelines formalizadas na linguagem PROforma	Sim	PROforma Language	Não	Formalização de guidelines
TR-22	Ontologias	Sim	SPARQL	Sim, outras ontologias na área da saúde	Integração com a RehabRobo-Onto
TR-23	Classes OWL, Instâncias OWL	Não	Não definido	Não	Método semiautomático via modelagem dos Objetos virtuais e suas funcionalidades
TR-24	RDF/OWL, XML, CSV, TSV	Sim	Extração e conversão para RDF	Sim	Manual via conversão e alinhamento ao formato RDF especificado
TR-25	OWL-RDF	Sim	OWL API e Jena API	Não	Indexação, extração e armazenamento dos modelos semânticos
TR-26	Ontologias	Não definido	Não definido	Não definido	Não definido

APÊNDICE I: DADOS EXTRAÍDOS: PERSISTÊNCIA, CONSUMIDORES DE CONHECIMENTO, SEGURANÇA E VALIDAÇÃO

CÓD.	MECANISMO DE PERSISTÊNCIA	HISTÓRICO DE CONSULTAS E RESULTADOS	CONSUMIDORES DE CONHECIMENTO	COMUNICAÇÃO COM OS CONSUMIDORES	CONTROLE DE ACESSO	FORMA DE VALIDAÇÃO
TR-01	Não definido	Não definido	Interfaces de usuário (aplicativos móveis, websites)	HTTP	Autenticação por senha, login único, sistema de permissão baseado em papeis	Protótipo sem medidas de performance
TR-02	Banco de dados	Sim	Paciente, Assistente de Saúde	HTTP	Login dos usuários no consumidor, sistema de autorização	Implementação de dois protótipos, um website e uma aplicação móvel.
TR-03	Ontologia	Não definido	Sistemas de Telemedicina	Não definido	Não definido	Estudo de caso focado no tratamento de diabetes com detalhes sobre a extração do conhecimento e avaliação de algumas métricas
TR-04	Historical Data Warehouse	Sim	Outros provedores de dados na área da saúde	Não definido	Não definido	Protótipo sem medidas de performance
TR-05	CSV, JSON	Não	Websites	Não definido	Público	Implementação de um website com um sistema de busca de termos sem medidas de performance
TR-06	Não Definido	Não	Usuário da Internet, Aplicativo Móvel, Website	JSON, HTTP	Login dos usuários no consumidor de conhecimento com registro aberto ao público	Implementação do banco de conhecimento e serviço proposto
TR-07	OpenEHR, Outros	Sim	Sistemas de informações clínicas (portais governamentais voltadas para o paciente e para profissionais de saúde)	XML, SOAP, HTTP	Id de usuário, Autenticação, autorização, sistema de controle de acesso, etc.	Implementado utilizando a infraestrutura do ministério da saúde
TR-08	Ontologia	Sim	Diversos Aplicativos Clientes HTTP	XML, REST, HTTP	Não definido	Múltiplas implementações, validações do modelo e medidas de performance
TR-09	Instâncias OWL	Sim	Diversos Aplicativos Clientes HTTP	REST, HTTP	Não definido	Implementação de dois aplicativos consumidores validados com um caso clinico
TR-10	OWL-RDF	Não	Interface Gráfica	Não definido	Não definido	Estudo de Caso com validação do especialista
TR-11	Diversos: ontologias, banco de dados relacional, RDF, etc.	Sim	Clientes de Serviços Web Externos	XML, SOAP, HTTP	Não definido	Protótipo Simples
TR-12	Não definido	Não	Sistemas especialistas voltados para profissionais da atenção primária e pacientes	REST, HTTP	API Key, login do usuário	Estudo de Caso na área da Nefrologia

TR-13	Não definido	Não definido	Médicos, especialistas	Não definido	Não definido	Exemplo de caso de uso executado dentro de um protótipo
TR-14	OWL	Não	Aplicações diversas de monitoramento e sistemas de decisão	OWL, Fila de Mensagens	Não definido	Implementação de Protótipo sem medidas de performance
TR-15	OWL, RDF, Outros	Não	Interfaces gráficas de usuário	OWL-S	Login do usuário, sistema de permissão	Implementação de protótipo e validação utilizando o tempo de execução e corretude
TR-16	Instâncias OWL	Sim	Clinicas, Médicos, profissionais de saúde	Não definido	Não definido	Estudo anterior que mostra que a abordagem funciona e um exemplo de execução
TR-17	RDF	Não	Clientes HTTP	JSON, HTTP	Não definido	Estudo de caso sem medidas de performance
TR-18	Instâncias OWL, Cloud based database	Sim	Aplicativos móveis	Não definido	Não definido	Implementação de um protótipo, Testes de latência e uma survey
TR-19	Não definido	Não definido	Organizações na área da Saúde	Não definido	Não definido	Não definido
TR-20	Não definido	Não	Não definido	Não definido	Não definido	Avaliação da precisão, Recall e F-Measure do retorno direto das fontes de conhecimento
TR-21	Não definido	Não	PROforma applications	Não definido	Não definido	Descreve duas aplicações já implementadas
TR-22	OWL 2 DL, SQL	Não definido	Robot designers, physical therapist, medical doctors	Web Interface, HTTP	Registro de usuários com confirmação do e-mail, sistema de permissão, autorização manual de modificações, captcha system	Implementação de protótipo com validação de especialista
TR-23	Instâncias OWL armazenadas em um banco de triplas	Sim	Aplicações baseadas em Internet das Coisas, Desenvolvedores, Aplicativos android, Especialistas de domínio, etc.	REST, HTTP	Não definido	Implementação parcial de protótipo sem medidas de performance
TR-24	RDF em uma banco de Triplas GraphDB	Não definido	Aplicações para Pacientes e médicos	REST, HTTP	Não definido	Estudo de Caso com validação do especialista
TR-25	Instâncias OWL armazenadas em um banco de triplas	Não definido	Não definido	SPARQL	Não definido	Response time Benchmark
TR-26	OpenEHR formatted data	Sim	Não definido	Não definido	Extensão do modelo Role Based Access Control	Não definido