UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Tese

Uma Proposta para Classificação de Recursos da IoT Explorando o Tratamento da Incerteza na Decisão Multicritério

Renato Marques Dilli

Pelotas, 2020

Renato Marques Dilli

Uma Proposta para Classificação de Recursos da IoT Explorando o Tratamento da Incerteza na Decisão Multicritério

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Adenauer Corrêa Yamin
Coorientadora: Profª. Drª. Renata Hax Sander Reiser

Pelotas, 2020

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

D578p Dilli, Renato Marques

Uma proposta para classificação de recursos da IoT explorando o tratamento da incerteza na decisão multicritério / Renato Marques Dilli ; Adenauer Corrêa Yamin, Renata Hax Sander Reiser, orientadores. — Pelotas, 2020.

120 f.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2020.

1. Iot. 2. Descoberta de recursos. 3. Classificação de recursos. 4. Incerteza. I. Yamin, Adenauer Corrêa, orient. II. Reiser, Renata Hax Sander, orient. III. Título.

CDD : 005

Elaborada por Maria Inez Figueiredo Figas Machado CRB: 10/1612

Renato Marques Dilli

Uma Proposta para Classificação de Recursos da IoT Explorando o Tratamento da Incerteza na Decisão Multicritério

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 8 de maio de 2020

Banca Examinadora:

Prof. Dr. Adenauer Corrêa Yamin (orientador)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Profª. Drª. Renata Hax Sander Reiser (coorientadora)
Doutora em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. André Du Bois
Doutor em Ciência da Computação pela Heriot-Watt University.

Prof. Dr. Claudio Fernando Resin Geyer
Doutor em Informática pela Université Joseph Fourier, Grenoble I.

Prof. Dr. João Ladislau Barbará Lopes
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

AGRADECIMENTOS

Ao Instituto Federal Sul-rio-grandense por investir em seus docentes e técnicos administrativos, possibilitando assim minha participação no curso de Doutorado do Programa de Pós-Graduação em Computação da UFPel.

Ao Prof. Dr. Adenauer Corrêa Yamin, meu orientador, que tem me acompanhado e orientado desde meu Mestrado. Agradeço pela sua amizade, sua incansável dedicação e disponibilidade em me auxiliar no desenvolvimento desta Tese.

À Profª. Drª. Renata Hax Sander Reiser por toda paciência e dedicação dada à revisão de artigos e às questões sobre tratamento de incerteza.

Agradeço aos colegas do PPGC da UFPel, em especial à Amanda Cardozo pelos momentos de estudo e pesquisa, que resultaram várias publicações importantes para o desenvolvimento desta Tese.

Ao Maurício Madruga e Patrick Fernandes do Programa de Pós-Graduação em Engenharia Eletrônica e Computação da UCPel pelas contribuições.

Aos funcionários e professores do PPGC pela seriedade e compromisso.

RESUMO

DILLI, Renato Marques. **Uma Proposta para Classificação de Recursos da IoT Explorando o Tratamento da Incerteza na Decisão Multicritério**. Orientador: Adenauer Corrêa Yamin. 2020. 120 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2020.

A Internet das coisas (IoT) é caracterizada por recursos conectados em rede, a maioria destes disponibilizando mais de um serviço. O processo de classificação para selecionar o recurso cujos serviços melhor atendem a requisição de um cliente constitui uma frente de pesquisa com diversos desafios atuais e de relevância internacional. Dentre os desafios destaca-se a elevada escalabilidade, a dinamicidade com que os recursos entram e saem do ambiente, bem como as incertezas do cliente no momento de especificar suas preferências.

Assim, devido a esta elevada escalabilidade da IoT, uma requisição de cliente pode retornar centenas, ou até milhares de recursos que atendem as especificações da requisição considerando seus parâmetros funcionais. Por sua vez, a revisão de literatura feita nesta Tese identificou a avaliação de parâmetros não funcionais como técnica recorrente nas diversas propostas para classificação de recursos.

Considerando este cenário, esta Tese tem por objetivo central apresentar uma proposta para classificação de recursos descobertos de forma autônoma, denominado EXEHDA-RR, que tem por base de concepção: (i) uma abordagem de decisão que pondera múltiplos critérios; e (ii) o tratamento da incerteza quando da definição dos diferentes parâmetros considerados.

Outro aspecto contemplado nesta Tese, está relacionado a minimização dos custos computacionais envolvidos no procedimento de classificação de recursos em ambientes dinâmicos como a IoT, onde a entrada e/ou a saída de recursos no ambiente computacional podem ocorrer a todo momento.

A avaliação do EXEHDA-RR foi realizada por meio de cenários de uso e a especificação dos recursos foi extraída do *dataset* QWS, que considera recursos reais disponibilizados na Internet. Os resultados obtidos mostram que é possível oferecer aos clientes os recursos mais adequados, de acordo com a preferência especificada. O emprego da Lógica Fuzzy dentre outros aspectos, possibilitou que a especificação das preferências por parte do cliente possa ser feita, de modo mais confortável, por meio de termos linguísticos, tais como, baixo, médio e alto, minimizando incertezas. Os resultados alcançados com os cenários de uso validam a proposta e apontam para a continuidade da pesquisa em trabalhos futuros.

Palavras-chave: IoT. Descoberta de Recursos. Classificação de Recursos. Incerteza.

ABSTRACT

DILLI, Renato Marques. **A Proposal for Classifying IoT Resources Exploring the Treatment of Uncertainty in Multicriteria Decision**. Advisor: Adenauer Corrêa Yamin. 2020. 120 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2020.

The Internet of things (IoT) is characterized by networked resources, most of which offer more than one service. The classification process for selecting the resource whose services best meet a client's request constitutes a research front with several current and internationally relevant challenges. Among the challenges we highlight the high scalability, the dynamism with which the resources enter and leave the environment, as well as the client's uncertainties when specifying their preferences. Thus, due to this high scalability of the IoT, a client request can return hundreds, or even thousands of resources that meet the specifications of the request considering its functional parameters. In turn, the literature review made in this Thesis identified the evaluation of non-functional parameters as a recurring technique in the various proposals for classification of resources. Considering this scenario, this Thesis has as main objective to present a proposal for classification of discovered resources in an autonomous way, called EXEHDA-RR, which is based on conception: (i) a decision approach that considers multiple criteria; and (ii) the treatment of uncertainty when defining the different parameters considered. Another aspect contemplated in this Thesis, is related to the minimization of the computational costs involved in the procedure of classification of resources in dynamic environments such as IoT, where the entry and/or exit of resources in the computing environment can occur at all times. The EXEHDA-RR evaluation was performed using usage scenarios and the resource specification was extracted from the QWS dataset, which considers real resources made available on the Internet. The results obtained show that it is possible to offer clients the most suitable resources, according to the specified preference. The use of Fuzzy Logic, among other aspects, made it possible for clients to specify their preferences more comfortably, using linguistic terms such as low, medium and high, minimizing uncertainties. The results achieved with the usage scenarios validate the proposal and point to the continuity of the research in future works.

Keywords: IoT. Resource Discovery. Resource Ranking. Uncertainty.

LISTA DE FIGURAS

1 Algoritmos MCDA 32
2 Módulos de um Controlador Fuzzy Tipo-1 38
3 Controlador Fuzzy Mamdani - Saídas Parciais das Regras 40
4 Controlador Fuzzy Mamdani - Saída Final das Regras 40
5 Conjuntos Fuzzy Tipo-1 e Tipo-2 41
6 Função de Pertinência de Conjunto Fuzzy Trapezoidal Tipo-2 42
7 FOU e Funções de Pertinência Superior e Inferior 43
8 Função de Pertinência de Conjunto Fuzzy Triangular Tipo-2 Intervalar 43
9 Módulos de um Controlador Fuzzy Tipo-2 44

10 RSL - Fluxo de Execução 48

11 Visão Geral da Arquitetura de Software do EXEHDA-RR 59
12 Ontologia Concebida para o Processador Semântico do EXEHDA-RR 62
13 Fluxo de Execução do Módulo Classificador de Recursos 64
14 Reclassificação da Saída Produzida pelo Algoritmo MCDA 66
15 Classificador Fuzzy - Funções de Pertinência Fuzzy Tipo-1 73
16 Classificador Fuzzy - Funções de Pertinência Fuzzy Tipo-2 Small . 74
17 Classificador Fuzzy - Funções de Pertinência Fuzzy Tipo-2 Large . 75
18 Comparação dos Resultados das Funções de Pertinência vs MCDA 75
19 Inferência Fuzzy - Processamento das Regras 16 a 18 81
20 Inferência Fuzzy - Processamento das Regras 25 a 27 82
21 Inferência Fuzzy - Resultado da Agregação de Regras 82
22 Inferência Fuzzy - Intervalo Resultante 83

23 Cenário 1 - Funções de Pertinência para RT e TH 87
24 Cenário 1 - Funções de Pertinência para AV Definidas pelo Especialista 87
25 Cenário 1 - Seleção da Preferência HML 88
26 Cenário 1 - Cadastro de Perfil do Processamento Fuzzy 88
27 Cenário 1 - Classificação da Preferência HML 89
28 Cenário 1 - Comparação dos Classificadores Fuzzy e WsRF 90
29 Cenário 2 - Busca por Recursos 92
30 Cenário 2 - Especificação da Preferência HLM 92
31 Cenário 2 - Resultado da Classificação com Preferência HLM . . . 93
32 Cenário 2 - Ranking do Classificador Fuzzy Considerando Preferências de Cliente 94
33 Cenário 3 - Busca por Recursos 95

34 Cenário 3 - Resultado da Classificação com Preferência HLM . . . 96
35 Cenário 3 - Ranking do Classificador Fuzzy Considerando Preferências do Cliente (Com a Inclusão de Novo Recurso) 98

36 Arquitetura de Software do *Middleware* EXEHDA 114
37 Organização Celular do Ambiente Ubíquo Gerenciado pelo EXEHDA na IoT 115

LISTA DE TABELAS

1 RSL - Bases de Dados e Strings de Pesquisa Utilizadas 47
2 RSL - Questões de Qualidade para Seleção 49
3 RSL - Seleção por Base de Dados dos Artigos 49
4 RSL - Seleção Final dos Artigos 50
5 RSL - Comparação dos Trabalhos Relacionados 56

6 Classificador MCDA - Atributos Selecionados do Dataset QWS . . . 66
7 Classificador MCDA - Dados Normalizados 67
8 Classificador MCDA - Dados Classificados 68
9 Pré-Classificador ML - Comparação da Acurácia dos Algoritmos . . 69
10 Pré-Classificador ML - Matriz de Confusão do Algoritmo LMT 69
11 Pré-Classificador ML - Matriz de Confusão do Algoritmo J48 69
12 Pré-Classificador ML - Avaliação da Precisão do Algoritmo LMT . . 71
13 Classificador Fuzzy - Variáveis Ling. Tipo-1: RT, AV e TH 73
14 Classificador Fuzzy - Variável Linguística Tipo-1: Weight 73
15 Classificador Fuzzy - Variáveis Linguísticas Tipo-2: RT, AV e TH . . 75
16 Classificador Fuzzy - Variável Linguística Tipo-2: Weight 76
17 Classificador Fuzzy - Base de Regras Mamdani 77
18 Classificador Fuzzy - Preferência de Cliente LHM 78
19 Classificador Fuzzy - Preferência de Cliente LHL 78
20 Classificador Fuzzy - Preferência de Cliente MHM 79
21 Classificador Fuzzy - Preferência de Cliente HLL 79
22 Classificador Fuzzy - Preferência de Cliente HHL 79
23 Classificador Fuzzy - Preferência de Cliente MLH 80
24 Classificador Fuzzy - Preferência de Cliente HLM 80
25 Classificador Fuzzy - Preferência de Cliente HML 81

26 Cenário 1 - Recursos Considerados para Classificação 86
27 Cenário 1 - Classificação da Preferência HML com WsRF 86
28 Cenário 2 - Recursos Selecionados pelo Classificador MCDA 91
29 Cenário 2 - Classificação com Preferências - HHH, LMH, LHL e MHM 91
30 Cenário 2 - Classificação com Preferências - HLL, HHL, MLH e HLM 93
31 Cenário 2 - Classificação Considerando todas Preferências de Cliente 94
32 Cenário 3 - Classificação com Preferências - HHH, LMH, LHL e MHM 96
33 Cenário 3 - Classificação com Preferências - HLL, HHL, MLH e HLM 97
34 Cenário 3 - Classificação Considerando todas Preferências de Cliente 97

35 Comparação com os Trabalhos Relacionados 101

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
AHP	*Analytic Hierarchy Process*
ANP	*Analytic Network Process*
AV	*Availability*
AVU	Ambiente Virtual do Usuário
BDA	Base da Dados das Aplicações
CC	Componente Cliente
CD	Componente Diretório
CIB	*Cell Information Base*
CPU	*Central Processing Unit*
CR	Componente Recurso
DEA	*Data Envelopment Analysis*
EXEHDA	*Execution Environment for Highly Distributed Applications*
EXEHDA-RR	EXEHDA - *Resource Ranking*
FDCRT	*Fuzzy Distance Correlation Ranking Technique*
FP	Função de Pertinência
FOU	*Footprint of Uncertainty*
FSIRT	*Fuzzy Subjective Interval-based Ranking Technique*
IFS	*Intuitionistic Fuzzy Sets*
IoT	*Internet of Things*
IP	*Internet Protocol*
JESS	*Java Expert System Shell*

KPI	*Key Performance Indicator*
LA	*Latency*
LMT	*Logistic Model Tree*
MADM	*Multiattribute Decison Making*
MAUT	*Multiple Attribute Utility Theory*
MCDA	*Multiple Criteria Decision Analysis*
MCSP	*Multicriteria Selection Problem*
MCMP	*Multicriteria Mathematical Problem*
MIT	*Massachusetts Institute of Technology*
ML	*Machine Learning*
MOA	*Massive Online Analysis*
MODM	*Multiobjective Decison Making*
MVC	*Model-view-controller*
NFPs	*Non-Functional Properties*
OWL	*Web Ontology Language*
P2P	*Peer-to-Peer*
QoS	*Quality of Service*
QWS	*Quality of Web Service*
RDF	*Resource Description Framework*
RE	*Reliability*
REST	*Representational State Transfer*
RMI	*Remote Method Invocation*
RRA	Repositório de Recursos Ativos
RRN	Repositório de Recursos Negados
RSL	Revisão Sistemática da Literatura
RT	*Response Time*
SAW	*Simple Additive Weight*
SBRF	Sistema Baseado em Regras Fuzzy
SBRF2	Sistema Baseado em Regras Fuzzy Tipo 2

SPARQL	*SPARQL Protocol and RDF Query Language*
SQL	*Structured Query Language*
TDB	*Trivial DataBase*
TCP	*Transmission Control Protocol*
TH	*Throughput*
TOPSIS	*Technique for Order of Preferences by Similarity to Ideal Solution*
UbiComp	Computação Ubíqua
UPnP	*Universal Plug and Play*
URI	*Uniform Resource Identifier*
WsRF	*Web Service Relevance Function*
VIKOR	*VlseKriterijumska Optimizacija Kompromisno Resenje*
XML	*eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO 16
1.1 Problema de Pesquisa 18
1.2 Objetivos 18
1.3 Organização do Documento 19

2 FUNDAMENTAÇÃO TEÓRICA 21
2.1 Internet das Coisas 21
2.1.1 Conceitos e Definições 22
2.1.2 Características Operacionais 23
2.2 Descoberta de Recursos 25
2.2.1 Categorias de Modelagem 26
2.2.2 Descoberta, Estimativa, Alocação e Monitoramento 27
2.3 Qualidade de Serviço 28
2.4 Classificação de Recursos 31
2.4.1 Algoritmos MCDA 31
2.4.2 Algoritmos Machine Learning 35
2.4.3 Lógica Fuzzy 37
2.4.4 Ordens Admissíveis 44
2.5 Considerações Sobre o Capítulo 45

3 ESTADO DA ARTE EM CLASSIFICAÇÃO DE RECURSOS 46
3.1 Revisão Sistemática da Literatura 46
3.1.1 Critérios de Inclusão 47
3.1.2 Critérios de Exclusão 48
3.2 Trabalhos Relacionados Identificados com a RSL 50
3.2.1 Discovery and Prioritization of Web Services Based on Fuzzy User Preferences for QoS 50
3.2.2 Prioritizing the Solution of Cloud Service Selection Using Integrated MCDM Methods Under Fuzzy Environment 51
3.2.3 Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences 52
3.2.4 Fuzzy AHP for Learning Service Selection in Context-Aware Ubiquitous Learning Systems 53
3.2.5 Combining Experts' Opinion with Consumers' Preference in Web Service QoS Selection 53
3.2.6 PuLSaR: Preference-based Cloud Service Selection for Cloud Service Brokers 54

3.2.7 Non Functional QoS Criterion Based Web Service Ranking 55
3.3 Análise dos Trabalhos Relacionados 56
3.4 Considerações Sobre o Capítulo 57

4 EXEHDA-RR: CONCEPÇÃO E FUNCIONALIDADES 58
4.1 Arquitetura de Software 58
4.1.1 CC - Componente Cliente 59
4.1.2 CR - Componente Recurso 60
4.1.3 CD - Componente Diretório 60
4.2 Módulo Classificador de Recursos 64
4.2.1 Serviço Classificador MCDA 64
4.2.2 Serviço Pré-Classificador Machine Learning 67
4.2.3 Serviço Classificador Fuzzy 71
4.3 Considerações Sobre o Capítulo 84

5 EXEHDA-RR: CENÁRIOS DE USO 85
5.1 Cenário 1 - Classificação de Recursos EXEHDA-RR vs WsRF 86
5.2 Cenário 2 - Classificação sem Persistência na Requisição do Cliente 90
5.3 Cenário 3 - Classificação Considerando Persistência na Requisição do Cliente 95
5.4 Considerações Sobre o Capítulo 98

6 CONSIDERAÇÕES FINAIS 99
6.1 Principais Conclusões 99
6.2 Publicações Realizadas 102
6.2.1 Publicações como Primeiro Autor 102
6.2.2 Publicações como Coautor 103
6.3 Trabalhos Futuros 104

REFERÊNCIAS 105

ANEXO A MIDDLEWARE EXEHDA 113
A.1 Arquitetura e Ambiente Computacional 113
A.2 Subsistemas do EXEHDA 116

1 INTRODUÇÃO

A Internet das Coisas (IoT - *Internet of Things*) caracteriza-se pela presença de uma grande quantidade de recursos que pelo seu perfil operacional realizam conexões transientes, com diferentes durações, totalizando uma grande quantidade de serviços cuja disponibilidade pode se alterar ao longo do tempo (PERERA, 2017).

O cenário atual contabiliza mais de vinte bilhões de recursos que podem se conectar à Internet para disponibilizar serviços a potenciais clientes. A previsão de aumento médio desse número é ao redor de 15% a cada ano, podendo chegar a 125 bilhões de recursos até 2030 (IHS Markit, 2017) (RESEARCH, 2017).

A seleção adequada de recursos que satisfaçam as exigências de um cliente, considerando a disponibilidade de um grande número de serviços aptos a atendê-lo, constitui uma tarefa com diferentes desafios de pesquisa. Tal situação aponta para a importância de trabalhos que possam contribuir com mecanismos para descoberta de recursos, na perspectiva de cenários computacionais com elevada escalabilidade, e que também considere um perfil de composição dinâmica, como acontece na IoT (BARNAGHI; SHETH, 2016).

Um desafio a ser vencido após a descoberta de recursos, neste cenário de grande escalabilidade, é o de classificá-los para selecionar aqueles que melhor possam atender à solicitação de clientes (GARCÍA, 2012).

Nesta Tese é considerado como cliente a entidade que realiza requisições por recursos no ambiente computacional. Sendo assim, o cliente pode ser representado por uma aplicação, um componente de software executando em um dispositivo IoT ou um usuário. É oportuno registrar que, em algumas literaturas, o termo “cliente” pode ser nomeado como “consumidor” (DELICATO; PIRES; BATISTA, 2017).

Por sua vez, a definição de recurso considerada é aquela apresentada por (TANENBAUM; BOS, 2014): “recurso é qualquer objeto que pode ser alocado dentro da infraestrutura computacional”. Os recursos muitas vezes são compostos de vários dispositivos constituídos por hardwares heterogêneos, tais como sensores e atuadores (RAZZAQUE et al., 2016). O acesso aos recursos geralmente é realizado por meio de serviços.

No trabalho de pesquisa desta Tese, entende-se que um processo de classificação de recursos deve considerar de forma sinérgica, tanto os requisitos funcionais, como os não funcionais (KHUTADE; PHALNIKAR, 2014).

Enquanto os requisitos funcionais definem o que o serviço disponibilizado pelo recurso efetivamente realiza, os requisitos não funcionais, especificam o nível de qualidade dos serviços oferecidos. Usualmente, os requisitos não funcionais são especificados por métricas de QoS (Quality of Service), por exemplo, o tempo necessário para responder às requisições e a porcentagem de tempo que um determinado serviço permanece no modo operacional.

Nessa perspectiva, as solicitações dos clientes descrevem as suas preferências com relação aos requisitos que devem ser atendidos pelos serviços providos pelos recursos.

Como premissa operacional, via de regra, os procedimentos de descoberta têm por base avaliar a disponibilidade ou não dos requisitos funcionais solicitados pelo cliente. Uma vez descobertos esses recursos, os procedimentos de classificação, devem identificar aqueles mais apropriados, considerando os parâmetros de QoS associados a estes.

Assim, as preferências dos clientes definem critérios de otimalidade, caracterizando o processo de classificação como um problema de otimização, que obtém o melhor serviço a partir de preferências estabelecidas (GARCÍA et al., 2008).

Nesta Tese, considerando o aspecto dinâmico da IoT, entende-se que muitas vezes não é possível ter uma definição a respeito dos atributos associados aos serviços prestados pelos recursos (SALAH; SAADI, 2016). Como consequência, decorre a necessidade de lidar com a incerteza. De modo mais específico, a classificação dos serviços oferecidos pelos recursos pode ser afetada por várias fontes de incertezas, as quais foram classificadas por (PLATENIUS et al., 2013) do seguinte modo: (i) incerteza decorrente de informações incompletas sobre os serviços, que deveriam ser fornecidas pelo desenvolvedor do recurso; (ii) incerteza decorrente das pressuposições adotadas para a avaliação dos serviços pela técnica de classificação aplicada; (iii) incerteza introduzida pela a ausência de um ou mais atributos de QoS envolvidos; por fim, (iv) incerteza dos clientes sobre a adequação dos valores dos atributos praticados pelos serviços disponibilizados pelos recursos.

As várias técnicas de classificação de recursos identificadas na revisão de literatura, apresentadas no Capítulo 3, diferentemente do proposto nesta Tese, não consideram a incerteza no momento da especificação das preferências. Isso ocorre tanto por parte dos clientes interessados n219,43os recursos, como na construção da opinião de especialistas que, neste trabalho, traduzem-se em funções de pertinência Fuzzy. Vale lembrar que o termo especialista, adotado nesta Tese, é o de alguém com profundo conhecimento técnico-científico, adquirido ao longo do tempo, sobre

os mecanismos de funcionamento de um determinado fenômeno ou processo (LIMA, 2015).

A seleção de recursos é um problema de decisão que ganha complexidade em um regime que se precisa considerar incertezas de especificação. Nesse contexto, a teoria dos conjuntos fuzzy vem sendo utilizada com sucesso para lidar com a aproximação e a incerteza em diferentes abordagens para tomada de decisão (KOHAR, 2018).

De forma complementar, o *Machine Learning* (ML) permite que um sistema examine dados e deduza o conhecimento. Em essência, o intuito do ML é identificar e explorar padrões ocultos nos dados. Os padrões descobertos são utilizados para analisar novos dados desconhecidos, para que possam ser mapeados para os grupos conhecidos (BOUTABA et al., 2018). Este fato mostra-se oportuno para otimizar procedimentos de comparação, reduzindo o esforço inerente a comparações extensivas a todo conjunto de dados.

Todos esses aspectos, somados às características operacionais da IoT, motivam o problema de pesquisa considerado neste trabalho.

1.1 Problema de Pesquisa

O problema de pesquisa central desta Tese consiste em: Como classificar recursos descobertos na infraestrutura computacional provida pela IoT?

Este problema de pesquisa se desdobra nos seguintes aspectos: (i) como gerenciar a grande quantidade de recursos, típica da IoT; (ii) como tratar a incerteza do cliente na especificação das suas preferências; (iii) como considerar os diferentes atributos de qualidade dos recursos no processo de sua classificação; e (iv) como promover a inclusão de novos recursos, considerando aspectos de escalabilidade.

Outrossim, registre-se que, além dos aspectos citados, no cenário de elevada escalabilidade da IoT, muitos recursos são semelhantes, introduzindo complexidade na definição de quais seriam os mais oportunos para as demandas dos diferentes clientes.

Por fim, é importante considerar que os esforços da pesquisa desenvolvidos tiveram como premissa contribuir com a arquitetura de software do *middleware* EXEHDA. Do problema de pesquisa e seus desdobramentos, foram consolidados os objetivos desta Tese, discutidos a seguir.

1.2 Objetivos

O objetivo geral desta Tese, considerando o problema de pesquisa elencado, é conceber uma proposta para classificação de recursos descobertos na IoT, denominada

EXEHDA-RR (*Resource Ranking*). Essa proposta classifica os recursos descobertos tendo por base tanto as preferências dos clientes, como os parâmetros funcionais e não funcionais dos recursos.

Tendo por base uma abordagem que facultasse um crescimento gradativo na especificidade da pesquisa desenvolvida, foram sistematizados os seguintes objetivos específicos:

- revisar, na perspectiva da IoT, os principais conceitos relacionados à Descoberta e Classificação de Recursos;
- identificar o estado da arte em classificação de recursos, empregando uma Revisão Sistemática da Literatura;
- conceber uma proposta para classificação de recursos, considerando o problema de pesquisa elencado;
- especificar componentes e funcionalidades a serem providos pela arquitetura de software que dará suporte à proposta;
- integrar uma abordagem para tomada de decisões na presença de critérios múltiplos na classificação de requisitos não funcionais dos recursos;
- explorar a Lógica Fuzzy no tratamento da incerteza no procedimento de classificação;
- integrar *Machine Learning* no procedimento de pré-classificação de recursos com o objetivo de otimização do esforço computacional, quando da classificação de recursos sob a condição de elevada escalabilidade;
- considerar na concepção da proposta que a mesma será integrada ao *middleware* EXEHDA;
- desenvolver cenários de uso que explorem os diferentes aspectos da proposta concebida;
- divulgar, ante a comunidade científica, os resultados atingidos pela pesquisa por meio de publicações em conferências e/ou periódicos especializados da área.

1.3 Organização do Documento

Este documento está organizado em seis capítulos, caracterizados a seguir:

- O Capítulo 2 apresenta a Fundamentação Teórica, introduzindo conceitos sobre a Internet das Coisas, a Descoberta de Recursos e a Qualidade de Serviço. Algumas abordagens para classificação de recursos são igualmente expostas.

- No Capítulo 3, é apresentada a Revisão Sistemática da Literatura, realizada para a identificação e a seleção do estado da arte em classificação de recursos. Dessa forma, são elencadas as etapas da revisão realizada, as principais características dos trabalhos selecionados e uma análise comparativa entre estes.

- O Capítulo 4 introduz a proposta de classificação de recursos EXEHDA-RR. Nele, ainda são mostradas a modelagem e as funcionalidades da arquitetura de software para descoberta e classificação de recursos para o *middleware* EXEHDA.

- No Capítulo 5 estão expostos três estudos de caso para avaliar a proposta de classificação de recursos EXEHDA-RR. Os aspectos dos cenários de avaliação, tecnologias utilizadas, protótipos desenvolvidos, testes realizados e os resultados também são apresentados.

- No Capítulo 6 encontram-se as Considerações Finais desta Tese, as publicações realizadas até o momento e a sugestão de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta um resumo dos estudos que fundamentam a modelagem do EXEHDA-RR. Nele, são destacados conceitos e, principalmente, alguns desafios inerentes da IoT, sendo discutidos os requisitos que devem ser atendidos no processo de descoberta de recursos da IoT. São também apresentados conceitos referentes à Qualidade de Serviço (QoS - *Quality of Service*) com a finalidade de integrar QoS ao processo de classificação e recursos na IoT. Por fim, são tratadas abordagens envolvendo técnicas e algoritmos para classificação de recursos, incluindo um estudo sobre Lógica Fuzzy, discutindo-se seu potencial para tratar incertezas no processo de classificação de recursos.

2.1 Internet das Coisas

A expressão Internet das Coisas (IoT) surgiu no laboratório de pesquisa Auto-ID Center do MIT (*Massachusetts Institute of Technology*) e foi apresentada, em 1999, por Kevin Ashton, em um relato intitulado “A Internet das Coisas tem o potencial de mudar o mundo, assim como a Internet fez. Talvez até mais” (ASHTON, 2009).

Ainda sob uma perspectiva histórica, em 2001 essa expressão foi descrita no artigo intitulado “*The Electronic Product Code*: *A naming Scheme for Physical Objects*” (BROCK, 2001), em que o autor descreve que a visão dos estudiosos era a de criar um “*Smart World*”, isto é, uma infraestrutura inteligente que conectaria vários objetos, informações e pessoas através da rede de computadores. Tal infraestrutura, por sua vez, permitiria o controle dos recursos físicos por meio do monitoramento remoto pelas pessoas e máquinas.

No início de 2015, o mundo já registrava 25 bilhões de dispositivos conectados à Internet, segundo a CES 2015 (*Consumer Electronics Show 2015*), mostrando que a Internet das Coisas, vinha tendo um destaque importante, sendo considerada a revolução tecnológica do futuro entre as infraestruturas de computação e comunicação.

O surgimento do paradigma da Internet das Coisas permite a criação de cenários de comunicação ubíquas, em que todos os objetos endereçáveis no ambiente podem

se comunicar e cooperar (RAY, 2018).

2.1.1 Conceitos e Definições

É importante apresentar alguns conceitos e definições sobre a Internet das Coisas. Estes foram selecionados na literatura da área, tendo como critério central a sua adoção tanto no cenário nacional, como internacional.

- A Internet das Coisas consiste em milhares de objetos interconectados através da Internet; um cenário em que objetos e pessoas se comunicam-se entre si, sem a intervenção direta do ser humano, possibilitando facilidades na vida diária das pessoas envolvidas com esse tipo de tecnologia, as quais usufruem de uma rede totalmente conectada, coletando informações de lugares, objetos e situações (PERERA et al., 2014).

- Internet das coisas é uma associação da computação ubíqua com a Internet do futuro. Ela interage com sensores heterogêneos, agregadores, atuadores e um domínio no contexto de aplicações, preservando a segurança (SOMA et al., 2011).

- A Internet das Coisas é um paradigma que prescreve um mundo de objetos físicos embarcados com sensores e atuadores, interligados por redes sem fio e que se comunicam utilizando a Internet, criando uma rede de objetos inteligentes capazes de executar processamentos, coletar variáveis ambientais e reagir a estímulos externos (ATZORI; IERA; MORABITO, 2010).

- Internet das coisas refere-se ao endereçamento de objetos e suas representações virtuais em uma estrutura igual à Internet. Tais objetos podem interligar informações sobre si mesmo ou podem transmitir dados de sensores sobre seu estado, em tempo real (AGGARWAL, 2013).

- A IoT é uma tecnologia que contribui para concretizar novos domínios de aplicação das tecnologias de informação e comunicação (TICs), a exemplo das cidades inteligentes, onde o uso de tecnologias avançadas de comunicação e sensoriamento objetiva promover serviços de valor agregado para os órgãos administrativos destas cidades e para as pessoas (ANDREA et al., 2014).

- A Internet das Coisas é uma importante evolução que a Internet obteve. Com ela, as pessoas não são as únicas a produzirem a ação, pois criam a possibilidade de os objetos poderem mediar suas vidas. Toda essa evolução promove a possibilidade de análise populacional, pois dá condições de coletar e distribuir dados, a fim de que sejam obtidos novos conhecimentos e novas informações (EVANS, 2011).

- A Internet das Coisas permite que objetos possam se conectar a qualquer momento, em qualquer lugar, com qualquer coisa, de preferência utilizando qualquer caminho ou rede e qualquer serviço (SAINT-EXUPERY, 2009).

- Os objetos inteligentes ou simplesmente "coisas"devem ser capazes de detectar fenômenos (físicos ou lógicos) e/ou disparar ações sobre o ambiente; possuir alguma capacidade computacional, ter capacidade de comunicação e identificador único (MIORANDI et al., 2012).

Tendo como base essas conceituações para a IoT, destacamos quatro aspectos que podem ser empregados para caracterizar a Internet das Coisas: (i) os sistemas são independentes; (ii) a IoT é instalada em cooperação com os novos serviços; (iii) a comunicação é feita de objetos-para-pessoas e de objetos-para-objetos; e (iv) o acesso à rede pode ser a todos ou restrito a alguns.

Com o avanço tecnológico que acontece no cenário mundial, surge o aparecimento da IoT, das redes de sensores sem fio, computação móvel e computação ubíqua. Porém, há uma série de desafios que deverão ser superados para a disseminação deste assunto, principalmente com relação ao desenvolvimento das aplicações e da alta heterogeneidade de hardware e software do ambiente em questão.

2.1.2 Características Operacionais

A partir da literatura da área, é possível se identificar diversas características operacionais que devem ser consideradas no desenvolvimento de sistemas de suporte para a Internet das Coisas (PAULO F. et al., 2015) (FLÁVIA C.; PAULO F.; THAIS, 2013) (GUBBI et al., 2013) (MIORANDI et al., 2012). Dentre elas, considerando o foco desta Tese, destacaríamos as seguintes:

- Heterogeneidade de dispositivos - a grande heterogeneidade de dispositivos físicos é uma das características centrais da Internet das Coisas, os quais poderão estar no ambiente de interesse das aplicações dos usuários. Essa heterogeneidade diz respeito às diferentes capacidades e recursos tecnológicos disponíveis em cada dispositivo, sejam de software, sejam de hardware. Nos cenários da IoT, com um elevado número de dispositivos, não é desejável que o desenvolvedor de aplicações ubíquas envolva-se no tratamento dos aspectos tecnológicos de baixo nível. Com isso, as soluções de *middleware* podem tratar a heterogeneidade e possibilitar a interoperabilidade e integração dos dispositivos e serviços que estão no ambiente. O grande propósito, nesse caso, consiste em gerenciar aspectos de baixo nível desses dispositivos da IoT e, ao mesmo tempo, diminuir as demandas de alto nível das aplicações da UbiComp.

- Escalabilidade - o número de dispositivos e a quantidade de informações produzidas envolvidas podem ser muito elevados, e isso acarreta em problemas de escalabilidade que englobam diferentes naturezas, incluindo: (i) nomeação e endereçamento, devido ao vasto número de dispositivos envolvidos; (ii) comunicação, pela quantidade de dados permutados na rede; (iii) tratamento de um volume enorme de dados; e (iv) gerenciamento de serviços, devido ao grande número de execuções.

- Eficiência energética - quanto à eficiência energética, muitos dispositivos da IoT operam com baterias, como nodos sensores, por exemplo. Isso demonstra que esses dispositivos apresentam uma quantidade limitada de energia, sendo assim, seu tempo de funcionamento, muitas vezes, está ligado à durabilidade dessa energia. Mesmo com o emprego de métodos de captação de energia que possam ser utilizados, como a utilização de painéis solares, por exemplo, os dispositivos nem sempre resolvem os problemas de energia. Por isso, o uso eficiente dessa energia é importante para prolongar a vida útil desses mecanismos, uma vez que nem sempre substituir ou recarregar a bateria é uma atividade simples. Nesse sentido, as estratégias de diminuição do consumo de energia geralmente estão ligadas à otimização da utilização das interfaces de comunicações sem fio e do processamento. Com isso, independentemente da utilização de baterias e do grande número de dispositivos contemplados em abordagens IoT, é necessário que, para soluções deste cenário, seja necessário o cuidado com relação ao consumo de energia desses dispositivos.

- Auto-organização - as características dos ambientes da IoT são as seguintes: (i) complexidade; e (ii) dinamismo. Em redes de sensores, esses dispositivos estão sempre saindo e entrando na rede, por término de energia, perda de sinal de rádio ou pela entrada de um novo dispositivo, devido à substituição por defeito ou necessidade de inserir outro ponto de monitoramento (JENNIFER; BISWANATH; Ghosal Dipak, 2008). Com o grande número de dispositivos envolvidos na rede é necessário que se tenha o mínimo de intervenção humana nos procedimentos de gerenciamento. Por isso, o uso de mecanismos de descoberta de dispositivos e serviços são vitais nestes cenários.

- Interoperabilidade semântica e gerenciamento de dados - o volume elevado de dados que são trocados na IoT necessita de tratamento. Para que esses dados possam resultar em informações úteis que sejam capazes de serem usadas por diferentes aplicações, é necessário disponibilizá-las em formatos adequados e padronizados, fornecendo modelos e descrições semânticas do conteúdo, chamadas de meta-dados, utilizando-se formatos e linguagens bem definidos. Isso

possibilita que sejam usados métodos de raciocínio sobre esses dados produzidos.

2.2 Descoberta de Recursos

Esta Seção caracteriza a descoberta de recursos no contexto da IoT e apresenta aspectos importantes que devem ser considerados durante o processo de descoberta. Também são apontados desafios de pesquisa que devem ser observados, tais como disponibilidade, escalabilidade e dinamicidade, em que os recursos entram e saem do ambiente computacional.

Há vários desafios a serem tratados no processo de descoberta de recursos na IoT. A disponibilidade é um deles, pois a infraestrutura e o ambiente da IoT são dinâmicos e a participação humana no processo de descoberta é inviável, portanto, a automatização é um requisito importante. Quando não há nenhuma infraestrutura de rede, cada dispositivo deve anunciar a sua presença e os recursos que oferece. Esse é um modelo diferente daqueles sistemas distribuídos centralizados, em que a publicação, a descoberta e a comunicação de recursos geralmente são gerenciados por um servidor dedicado. Mecanismos de descoberta também precisam escalar bem, no que diz respeito à quantidade de recursos, além de haver uma distribuição eficiente da carga computacional inerente ao esforço de descoberta, dada a composição de dispositivos com recursos limitados na IoT.

Além da disponibilidade e escalabilidade dos recursos, os trabalhos de (GOMES et al., 2016) e (PAGANELLI; PARLANTI, 2012) destacam os seguintes requisitos que devem ser atendidos na descoberta de recursos para IoT:

- Esquema de Identificação Flexível - transparência em relação ao esquema de identificação adotado (por exemplo, URIs, endereços IP, etc.), usado para endereçar recursos e serviços.
- Consulta de vários atributos - capacidade de encontrar recursos ao fazer consultas com base em uma correspondência exata de um ou mais atributos.
- Consulta de Intervalo - capacidade de encontrar recursos com base em consultas, especificando limites em atributos únicos ou múltiplos.
- Lidar com vários provedores - várias entidades podem produzir e publicar informações sobre um dado recurso.
- APIs de gerenciamento - capacidade de gerenciar informações associadas a um determinado recurso.
- Descrição semântica de recursos e serviços - uso de ontologias para modelar semanticamente recursos e serviços expostos.

- Modelagem de critérios QoC - modelagem de indicadores de QoC associados a dados de contexto.
- Interações síncronas e assíncronas - capacidade de lidar com interações síncronas e assíncronas com os clientes.

As principais atividades de um sistema de gerenciamento de recursos na IoT são as seguintes (DELICATO; PIRES; BATISTA, 2017): modelagem, descoberta, estimativa, alocação e monitoramento. Considerando que o tema central de pesquisa em desenvolvimento é a classificação de recursos, enquanto parte importante do gerenciamento, entendemos que a modelagem de recursos é uma atividade essencial na melhoria do processo de descoberta e classificação de recursos. Por esse motivo, é discutida com mais detalhes que as outras atividades.

2.2.1 Categorias de Modelagem

A atividade de modelagem de recursos pode ser agrupada em diferentes categorias, considerando suas abordagens e primitivas usadas para representar recursos. A modelagem, nesta Tese, foi classificada em três categorias: baseada em atributos, baseada em semântica e baseada em virtualização. Cada categoria difere entre si quanto ao grau de abstração, granularidade e formalismo com que os elementos são representados nos modelos (DELICATO; PIRES; BATISTA, 2017).

Categoria de Modelagem Baseada em Atributos

Pode ser considerada a mais direta. Uma vez que os recursos são descritos como um conjunto de atributos, esta categoria permite uma granularidade mais fina, já que os atributos descrevem características específicas do hardware do dispositivo, tais como CPU e capacidade de armazenamento. Tais características são modeladas em termos do que o recurso é capaz de fornecer.

Em relação ao grau de abstração, esta categoria tem um grau que pode ser considerado baixo, pois os atributos estão diretamente relacionados às características físicas dos dispositivos. Ou há um mapeamento um-para-um entre o atributo e o recurso físico que representa, ou um-para-muitos no caso de recursos virtuais. Em qualquer caso, os elementos utilizados sempre referem-se a atributos de baixo nível de dispositivos.

Categoria de Modelagem Baseada em Semântica

Mecanismos mais sofisticados para modelagem de recursos são baseados no uso de tecnologias semânticas, tais como ontologias. Esses mecanismos são frequentemente adotados por abordagens orientadas a serviços e promovem um alto nível de

abstração para representar os recursos da IoT. Nesta categoria, os modelos podem ser criados a partir de uma perspectiva de nível superior e os recursos podem ser consumidos como um serviço.

As ontologias são baseadas na lógica de descrição, portanto, proporcionam, inerentemente, um alto grau de formalismo na representação de recursos. Uma ontologia é uma representação formal de um conjunto de conceitos dentro de um domínio e das relações entre esses conceitos. As ontologias são muito expressivas e flexíveis, de modo que permitem representar hardware e software, recursos físicos e virtuais, e em uma variedade de granularidades. Conceitos podem ser definidos para modelar um dispositivo IoT de forma única e seus componentes físicos, juntamente com as relações entre eles, evitando a ambiguidade de descrição.

Categoria de Modelagem Baseada em Virtualização

Esta categoria de modelagem consiste em utilizar abordagens de virtualização, que podem fornecer um alto grau de abstração ao modelo de recursos IoT. Essas abordagens podem se concentrar na virtualização de dispositivos, virtualização de rede ou virtualização do sistema como um todo, fornecendo assim diferentes granularidades para modelagem. Nesse contexto, as técnicas definidas por software são mecanismos principais para abstrair componentes e funcionalidades de baixo nível (recursos), permitindo seu provisionamento e gerenciamento através de APIs de alto nível e bem definidas.

Em relação ao rigor/precisão dos modelos adotados para representar recursos e aplicações, pode-se considerar que eles têm um baixo grau de formalismo (DELICATO; PIRES; BATISTA, 2017). Basicamente, os recursos IoT são descritos como unidades definidas por softwares e expostos por meio de APIs bem definidas. Essas unidades definidas por softwares abrangem aspectos funcionais e não funcionais dos recursos IoT e são configuradas com base em um conjunto de políticas.

2.2.2 Descoberta, Estimativa, Alocação e Monitoramento

Antes que os recursos possam ser alocados, eles precisam ser descobertos no ecossistema IoT. Em um sistema heterogêneo como este, tal descoberta é, em si, um desafio agravado pela atual falta de padronização de protocolos e formatos. Além disso, a alta escalabilidade da IoT requer que o ambiente computacional forneça algum mecanismo capaz de registrar e descobrir recursos e serviços de forma autoconfigurada, eficiente e dinâmica.

Por vezes, é útil tentar estimar a quantidade de recursos a serem utilizados, a fim de garantir que todos aqueles necessários a uma aplicação em execução sejam providos, e as estratégias dessa estimação são geralmente baseadas na manutenção

de dados históricos do consumo de recursos, obtidos através de atividades de monitoramento. Nos casos em que existe uma estimativa e, possivelmente, reserva de recursos, pode ocorrer de, eventualmente, estes não serem utilizados (por exemplo, porque houve excesso de provisionamento ou porque as condições de execução mudaram). Portanto, é possível que seja necessário o sistema ter de recuperar esses recursos não utilizados, devolvendo-os ao conjunto de recursos disponíveis.

A alocação de recursos na infraestrutura da IoT tem requisitos adicionais introduzidos pelos dispositivos deste tipo de infraestrutura, cada um com uma natureza específica dos dados produzidos, caracterizados como fluxos de dados. O processo de alocação deve considerar tanto os fluxos dos dispositivos como dos nós intermediários, tais como *Gateways*. O objetivo principal da alocação de recursos é garantir a adequada reunião de requisitos de aplicação, com os dispositivos providos pela infraestrutura, minimizando o seu custo operacional.

O monitoramento de recursos, essencial para proporcionar um serviço de qualidade, é uma atividade que deve tratar a natureza altamente dinâmica dos sistemas IoT, considerando aspectos tanto no que diz respeito à disponibilidade de conexões de rede, à presença de dispositivos, bem como às variações de interesse do usuário. Além disso, o monitoramento sobre o uso de recursos atuais também é necessário para manter uma alocação de recursos ótima ou próxima à otimizada, e isso implica que a gestão de recursos para a IoT deve estar ciente do contexto e que o processo de alocação não pode ser estático, porém dinâmico e adaptativo para acomodar tais variações no contexto de execução. As soluções para detectar alterações ambientais e as adaptações a elas permitirão a prestação de serviços baseados em contexto, ajudando a fornecer a prestação de serviços mais adequados, de acordo com cada situação que se apresentar.

2.3 Qualidade de Serviço

A Qualidade de Serviço (QoS) consiste de uma combinação de várias qualidades ou propriedades de um determinado serviço, tais como disponibilidade, tempo de resposta, precisão e vazão (MENASCÉ; RUAN; GOMAA, 2007).

Enquanto os requisitos funcionais definem o que os serviços oferecidos pelos recursos deverão fazer, os requisitos não funcionais, em particular, a QoS, especifica o nível de qualidade dos serviços, definindo, por exemplo, o tempo necessário para um serviço responder a várias requisições e a porcentagem de tempo em que um serviço está funcionando.

A qualidade do serviço como um componente não funcional é a “capacidade de fornecer um serviço satisfatório” por meio de diferentes recursos. Devido à natureza heterogênea da IoT, a QoS em IoT apresenta muita flutuação, pois os recursos en-

volvidos apresentam perfis operacionais bastantes particulares, desde sensores de pequeno porte até servidores que aglutinam dados e disponibilizam informações já processadas de elevado valor agregado.

Assim, a aplicabilidade de um conjunto de parâmetros de QoS irá depender do domínio específico da aplicação IoT, em combinação com as diferentes tecnologias envolvidas (BHADDURGATTE; BP, 2015).

Os principais requisitos de QoS para os serviços disponibilizados por recursos apresentam-se sumarizados a seguir (LEE et al., 2003):

1. *Performance* - o desempenho de um serviço representa a rapidez com que uma solicitação de serviço pode ser concluída, podendo ser medido em termos de taxa de transferência, tempo de resposta, latência, tempo de execução e tempo de transação. Taxa de transferência refere-se ao número de solicitações de serviço da web, atendidas em um determinado intervalo de tempo. O tempo de resposta é o tempo necessário para concluir uma solicitação de serviço da web. Por sua vez, a latência é o atraso de ida e volta entre enviar uma solicitação e receber a resposta. Já o tempo de execução é o tempo gasto por um serviço para processar sua sequência de atividades. Finalmente, o tempo de transação representa o tempo que transcorre enquanto o serviço está concluindo uma transação completa.

2. *Reliability* - os serviços devem ser fornecidos com alta confiabilidade, representada, aqui, como a capacidade que um serviço tem para executar suas funções requeridas sob condições estabelecidas por um intervalo de tempo especificado. A confiabilidade é a medida geral de um serviço para manter sua qualidade de serviço, já que a medida geral de um serviço está relacionada ao número de falhas por dia, semana, mês ou ano. A confiabilidade também está relacionada à entrega garantida e ordenada de mensagens transmitidas e recebidas por solicitantes de serviços e provedores de serviços.

3. *Scalability* - os serviços devem ser fornecidos com alta escalabilidade, que representa a capacidade de aumentar a capacidade de computação do provedor de serviços e a capacidade do sistema de processar mais solicitações, operações ou transações de usuários em um determinado intervalo de tempo. Esse requisito também está relacionado ao desempenho. Os serviços devem ser escalonáveis em termos do número de operações ou transações suportadas.

4. *Capacity* - os serviços devem ser fornecidos com a capacidade necessária à demanda. Essa capacidade diz respeito ao limite do número de solicitações simultâneas que devem ser fornecidas com desempenho garantido. Os serviços, nesse caso, devem suportar o número necessário de conexões simultâneas.

5. *Robustness* - os serviços devem ser fornecidos com alta robustez, aqui representando o grau em que um serviço pode funcionar corretamente, mesmo na presença de entradas inválidas, incompletas ou conflitantes. Esses serviços devem, ainda, funcionar mesmo se parâmetros incompletos forem fornecidos para a chamada da solicitação de serviço.

6. *Exception Handling* - os serviços devem ser fornecidos com a funcionalidade de manipulação de exceção. Como não é possível ao projetista de serviços especificar todos os possíveis resultados e alternativas, as exceções devem ser tratadas adequadamente, sendo que esse tratamento está relacionado à maneira como o serviço lida com essas exceções.

7. *Accuracy* - os serviços devem ser fornecidos com alta precisão, definida aqui como a taxa de erro gerada pelo serviço, sendo que o número de erros gerado ao longo de um intervalo de tempo deve ser minimizado.

8. *Integrity* - a integridade dos serviços deve ser fornecida para que um sistema ou componente possa impedir o acesso não autorizado ou a modificação de programas ou dados de computador. Pode haver dois tipos de integridade: integridade de dados, que define se os dados transferidos são modificados em trânsito, e a integridade transacional, que se refere a um procedimento, somente, ou mesmo a um conjunto destes, o que garante preservar a integridade do banco de dados em uma transação.

9. *Accessibility* - os serviços da Web devem ser fornecidos com alta acessibilidade, representada, aqui, quando o serviço da web é capaz de atender às solicitações do cliente. Alta acessibilidade pode ser alcançada, por exemplo, construindo-se sistemas altamente escaláveis.

10. *Availability* - o serviço deve estar pronto (isto é, disponível) para consumo imediato. Essa disponibilidade é a probabilidade de o sistema estar ativo e confiável e, por isso mesmo, o tempo para reparo está associado à disponibilidade, já que o serviço deve estar disponível imediatamente quando for invocado.

11. *Interoperability* - os serviços devem ser interoperáveis entre os diferentes ambientes de desenvolvimento usados para implementá-los, de modo que seus desenvolvedores não precisem pensar em qual linguagem de programação ou qual sistema operacional estão hospedados esses serviços.

12. *Security* - os serviços devem ser fornecidos com a segurança necessária. Com o aumento de serviços que são fornecidos pela Internet, há uma preocupação crescente com a segurança. O provedor de serviços pode aplicar diferentes

abordagens e níveis de fornecimento de diretivas de segurança, dependendo do solicitante do serviço.

2.4 Classificação de Recursos

A classificação de recursos, tópico central desta Tese, tem como tônica central permitir uma seleção de recursos que melhor atendam demandas dos diferentes clientes. Neste sentido, as solicitações dos clientes descrevem os requisitos e as suas preferências com relação aos serviços oferecidos pelos recursos. No que diz respeito aos requisitos, eles indicam as funcionalidades necessárias que um cliente está procurando, de modo que estão vinculados ao processo de descoberta. Por sua vez, as preferências são interpretadas unicamente por mecanismos de classificação para ordenar serviços dos recursos que foram descobertos. As exigências dos clientes são geralmente suportadas por *middlewares*, cujas propostas de descoberta dependem de seus modelos.

A classificação dos serviços disponibilizados pelos recursos pode ser afetada por certas fontes de incertezas. O trabalho de (KAKARONTZAS; GEROGIANNIS, 2016) lida com esse assunto, com os Conjuntos Fuzzy Intuicionais (IFSs - *Intuitionistic Fuzzy Sets*), como segue: (i) utiliza IFSs para refletir avaliações faltantes de serviços; (ii) quando há informações desconhecidas sobre pesos de critérios de QoS, são calculados pesos de entropia (ou seja, objetivos) para concluir com uma classificação final dos serviços; e (iii) utiliza IFSs refletindo o acordo ou desacordo dos clientes sobre os valores max e min avaliados dos atributos de benefício e qualidade de custo.

A revisão de literatura, registrada no Capítulo 3, permitiu identificar abordagens passíveis de serem utilizadas para classificação de recursos em cenários típicos de sistemas distribuídos. Essas abordagens estão organizadas nas Seções que seguem.

A Seção 2.4.1 trata dos métodos baseados na Análise de Decisão de Múltiplos Critérios (MCDA *Multiple Criteria Decision Analysis*), a Seção 2.4.2 discute os algoritmos baseados em *Machine Learning*, e por sua vez a Seção 2.4.3 apresenta a Teoria dos Conjuntos Fuzzy.

Outrossim, enquanto uma extensão ao tratamento da incerteza oferecido pela Lógica Fuzzy, a Seção 2.4.4 introduz o conceito geral pertinente às Ordens Admissíveis.

2.4.1 Algoritmos MCDA

A tomada de decisões na presença de critérios múltiplos, geralmente conflitantes, é denominada de Análise de Decisão de Múltiplos Critérios (MCDA - *Multiple Criteria Decision Analysis*). Os algoritmos MCDA utilizam um conjunto de objetivos e critérios e visam auxiliar no julgamento da tomada de decisão, já que estimam seus pesos de importância relativa e estabelecem a contribuição de cada opção em relação a cada

critério de desempenho. MCDA não é apenas um conjunto de teorias, metodologias e técnicas, mas, também, uma perspectiva específica para lidar com problemas de tomada de decisão (GRECO; EHRGOTT; FIGUEIRA, 2016).

O trabalho de (WHAIDUZZAMAN et al., 2014) apresenta uma revisão bibliográfica sobre algoritmos MCDA, organizando-os em problema-base, tipos, métodos e categorias. A Figura 1 apresenta os algoritmos, destacando os métodos mais utilizados.

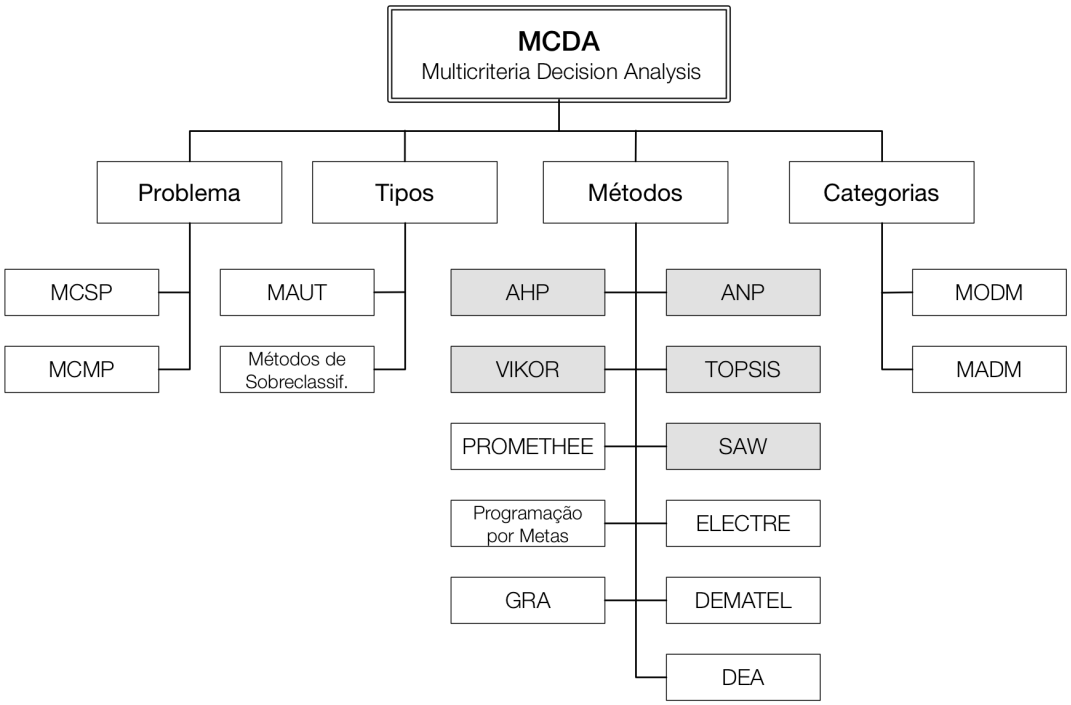

Figura 1 – Algoritmos MCDA
Fonte: Adaptado de (WHAIDUZZAMAN et al., 2014)

Os algoritmos foram classificados conforme o problema-base a ser resolvido: (i) Problema de Seleção Multicritério (MCSP - *Multicriteria Selection Problem*), que seleciona a melhor alternativa a partir de um conjunto finito de alternativas, todas conhecidas a priori; e (ii) Problema Matemático Multicritério (MCMP - *Multicriteria Mathematical Problem*), que seleciona a melhor alternativa a partir de um conjunto muito grande ou infinito de alternativas, nem todas são conhecidas a priori.

Métodos MCDA podem ser categorizados em dois tipos (WHAIDUZZAMAN et al., 2014): (i) por Teoria da Utilidade Multiatributo (MAUT - *Multiple Attribute Utility Theory*), que tem origem na Teoria da Utilidade e aborda situações com múltiplos objetivos, representados por atributos; e (ii) por métodos de sobreclassificação (*outranking*). A MAUT tenta encontrar uma função que reflita a utilidade de uma alternativa particular. A cada ação é atribuída uma utilidade marginal, com um número real representando a preferência da ação considerada. A utilidade retornada é a soma dessas utilidades

marginais. Métodos de sobreclassificação decidem se uma alternativa é classificada com grau mais alto que outro (métodos ELECTRE e PROMETHEE).

Os algoritmos foram organizados em duas categorias: tomada de decisão multi-atributo (MADM - *Multiattribute Decision Making*) e tomada de decisão multiobjetiva (MODM - *Multiobjective Decision Making*). De acordo com um levantamento realizado por (WHAIDUZZAMAN et al., 2014) para avaliar o uso de diferentes métodos, especialmente para melhorar a qualidade, o fuzzy foi o mais utilizado, representando 40% do total, seguido por AHP e ANP juntos em aproximadamente 20%. Em contraste, DEA - *Data Envelopment Analysis*, Programação por Metas e ELECTRE foram empregados raramente (MOHAMMADSHAHI, 2013). Os métodos mais utilizados, segundo (WHAIDUZZAMAN et al., 2014) e (NUNES et al., 2016), são apresentados resumidamente a seguir.

Analytic Hierarchy Process (AHP)

Em 1980, Thomas L. Saaty descobriu o AHP, um método popular e amplamente utilizado para MCDA, que permite o uso de critérios qualitativos e quantitativos ao avaliar alternativas. O AHP é baseado numa comparação parcial, cujos atributos não são inteiramente independentes uns dos outros, ou seja, estão estruturados em uma relação hierárquica. A hierarquia começa do nível superior, passando pelos níveis mais baixos, que correspondem aos critérios, subcritérios, e assim por diante, até chegar ao objetivo. Nessa árvore de hierarquia, o processo começa a partir de nós/folha e progride até o nível superior. Cada nível de saída representa a hierarquia correspondente ao peso ou à influência de diferentes ramos originários desse nível. Finalmente, após realizar as comparações, a melhor alternativa em relação a cada atributo é geralmente selecionada (CRISTÓBAL, 2011).

Analytic Network Process (ANP)

Extensão do AHP, a ANP, proposta por Thomas L. Saaty em 1996, refere-se a uma técnica abrangente de tomada de decisão destinada a superar o problema da dependência e do *feedback* entre os critérios, utilizando relações hierárquicas unidirecionais entre os níveis de decisão. Para lidar com a interdependência entre os elementos, a ANP deriva uma “supermatriz” que contém pesos compostos (LIANG et al., 2013). Para a obtenção da supermatriz, assume-se que, para um sistema de n agrupamentos ou componentes, os elementos de cada componente interagem ou impactam ou, ainda, são influenciados por alguns ou por todos os elementos do próprio ou de outro agrupamento.

Technique for Order of Preferences by Similarity to Ideal Solution (TOPSIS)

Essa técnica mostra preferência pela semelhança com uma solução ideal, que tenta selecionar uma alternativa que mais se aproxima desta e, simultaneamente, mais se distancia da solução anti-ideal. Assim, a matriz de decisão é inicialmente normalizada usando a normalização do vetor, e as soluções ideais e anti-ideais são identificadas dentro da matriz de decisão normalizada. TOPSIS foi desenvolvido em 1981 por Hwang e Yoon e seleciona alternativas com o menor distanciamento da solução ideal positiva e o distanciamento maior da solução ideal negativa (MOHAMMADSHAHI, 2013). É um método de critérios múltiplos para identificar soluções a partir de um conjunto finito de alternativas. As soluções ótimas devem ter a menor distância da solução ideal positiva e a mais distante da solução ideal negativa (CRISTÓBAL, 2011). O método TOPSIS introduz uma função de agregação, incluindo as distâncias do ponto ideal e do ponto ideal negativo, sem considerar sua importância relativa. No entanto, o ponto de referência pode ser uma grande preocupação na tomada de decisão, e deve ser o mais próximo possível da solução ideal (REHMAN; HUSSAIN; HUSSAIN, 2012) (CRISTÓBAL, 2011).

VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR)

VIKOR, também conhecido como o método de classificação de compromisso, é uma ferramenta eficaz na tomada de decisão multicritério. O seu índice de classificação multicritério baseia-se numa medida de “proximidade” à solução ideal. VIKOR emprega normalização linear, mas os valores normalizados não dependem da unidade de avaliação de um critério. Uma função de agregação equilibra a distância da solução ideal entre satisfação individual e ideal (CRISTÓBAL, 2011). É uma técnica eficiente de MCDA para classificar e, em seguida, selecionar a partir de um conjunto de alternativas conflitantes (ALABOOL; MAHMOOD, 2013).

Simple Additive Weight (SAW)

O algoritmo SAW usa uma pontuação de avaliação, obtida através de valores de critérios normalizados multiplicados por um peso, para classificar cada opção disponível. As opções são classificadas em ordem decrescente de acordo com sua pontuação final, que é a soma das pontuações para cada critério. SAW pode ser resumido em três passos (TZENG; HUANG, 2011):

1. Normalizar a matriz Q (1) de acordo com a Equação (2), se o critério deve ser maximizado, ou Equação (3), se o critério deve ser minimizado.

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1j} \\ q_{21} & q_{22} & q_{23} & \cdots & q_{2j} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ q_{i1} & q_{i2} & q_{i3} & \cdots & q_{ij} \end{bmatrix} \quad (1)$$

$$q'_{ij} = \frac{q_{ij} - q_j^{min}}{q_j^{max} - q_j^{min}} \quad (2) \qquad q'_{ij} = \frac{q_j^{max} - q_{ij}}{q_j^{max} - q_j^{min}} \quad (3)$$

2. Calcular o vetor pontuação de cada opção disponível. Cada pontuação pode ser calculada usando a Equação (4), em que w_j corresponde ao peso do critério e N representa o número de critérios na matriz de avaliação.

$$\phi(q_i) = \sum_{j=1}^{N} w_j . q'_{ij} \quad (4)$$

3. Classificar q_i em ordem decrescente de acordo com a pontuação $\phi(q'_i)$ para obter o ranqueamento resultante.

O trabalho de (NUNES et al., 2016) avalia os algoritmos *The Simple Additive Weighting* (SAW), *Technique for the Order of Prioritisation by Similarity to Ideal Solution* (TOPSIS) e *VIseKriterijumska Optimizacija I Kompromisno Resenje* (VIKOR). O critério Ótimo de Pareto, que lida com o conceito de dominância para determinar quando uma solução é melhor que outra, foi utilizado para comparar a qualidade das soluções obtidas por método. Os autores concluíram que SAW obteve um desempenho melhor ou igual, confrontado aos algoritmos TOPSIS e VIKTOR.

Sumarizadas as técnicas baseadas em especificação, a seguir é contemplado o estudo realizado sobre técnicas baseadas em *Machine Learning* para classificação de recursos.

2.4.2 Algoritmos Machine Learning

O *Machine Learning* (ML) visa compreender a lógica dos programas para melhorar o desempenho das máquinas em determinadas tarefas, através da experiência anterior (SIMON, 1983). Para tal, empregam um princípio denominado indução, no qual se obtêm conclusões genéricas a partir de um conjunto particular de exemplos.

Assim, algoritmos de Aprendizado de Máquina aprendem a induzir uma função ou hipótese capaz de resolver um problema a partir de dados que representam instâncias do problema a ser resolvido (FACELI et al., 2011). Desse modo, os algoritmos de

aprendizado automático têm-se revelado extremamente úteis em diversos domínios, desde logo têm sido especialmente importantes na resolução de problemas de mineração de dados, onde universos de dados extensos poderão conter, implicitamente, informações consideradas de valor e que poderão ser descobertas automaticamente.

Os algoritmos de *Machine Learning* podem ser divididos em três etapas: análise dos dados, também chamado de pré-processamento, métodos de aprendizado e análise dos resultados ou pós-processamento.

O *Machine Learning* não oferece por si só as saídas para todo e qualquer problema. De forma geral, não é possível aplicar um método de ML sobre uma massa de dados em seu formato original. Isso acontece devido às diferentes características, dimensões e formatos que os dados podem apresentar. Nesse sentido, o pré-processamento de dados consiste em retirar ou substituir instâncias ou padrões que possuam valores de atributos faltosos, aberrantes ou inconsistentes, tornando a base de dados mais suscetível à utilização de um método de ML.

Métodos de *Machine Learning* são modelos computacionais que observam um certo grupo de padrões (instâncias) e tentam realizar a definição de uma resolução geral para um problema. Apesar de serem uma ferramenta com uma definição resumida comum, há diversos tipos de métodos de ML. O aprendizado indutivo possui vários tipos de acordo com as tarefas de aprendizado, e é dividido em aprendizado supervisionado (preditivo) e não supervisionado (descritivo). As tarefas supervisionadas se dividem pelo tipo dos rótulos: classificação, para os rótulos discretos, e regressão, para os rótulos contínuos. Já as tarefas descritivas são divididas em: agrupamento, sumarização e associação. No agrupamento os dados são agrupados de acordo com a sua similaridade; na sumarização, o objetivo é encontrar uma descrição simples e compacta para os dados; já na associação, deve-se encontrar padrões para as associações entre os atributos dos dados. Em tarefas de previsão, o objetivo é encontrar uma função (hipótese) a partir dos dados de treinamento, é rotular ou dar uma "classe" a um padrão (instância) qualquer.

Uma das técnicas utilizadas para aprendizado supervisionado são as Árvores de Decisão (WITTEN; FRANK; HALL, 2011). Essa técnica utiliza a abordagem de dividir para conquistar, e gera uma árvore a partir do conjunto de treinamento, sendo que essa árvore é utilizada para classificar as novas instâncias.

A análise dos resultados para modelos preditivos é realizada segundo diferentes aspectos, tais como taxa de acerto do modelo gerado, compreensibilidade do conhecimento extraído e tempo de aprendizado. A maioria das medidas de análise de algoritmos preditivos está relacionada com o desempenho das predições realizadas. Medidas de erro, por exemplo, são largamente utilizadas na avaliação dos métodos preditores em ML. Esse tipo de métrica usa uma taxa de erro ou classificações incorretas para medir o desempenho do classificador. Técnicas de amostragem também

são usadas para se obter estimativas de desempenho preditivo mais confiáveis. As principais são *Holdout*, Amostragem Aleatória, Validação Cruzada e *Bootstrap*.

2.4.3 Lógica Fuzzy

A teoria dos conjuntos fuzzy (ou difusos), que tem sido amplamente aplicada para modelar as ambiguidades do pensamento humano, também resolve efetivamente as incertezas nas informações disponíveis para tomada de decisão baseada em múltiplos critérios. O peso de significância de critérios são avaliados em termos de valores linguísticos representados por números difusos. No conjunto fuzzy, as variáveis linguísticas são usadas para descrever termos difusos que mapeiam variáveis numéricas. No processo de tomada de decisão os valores de verdade da lógica booleana são substituídos por valores numéricos no intervalo unitário $[0, 1]$ (ALABOOL; MAHMOOD, 2013).

A principal vantagem de se utilizar a Lógica Fuzzy deve-se a sua capacidade de lidar com incertezas, raciocínio aproximado, termos vagos e ambíguos, o que não é possível de se fazer com as lógicas clássicas. O raciocínio humano envolve todos esses elementos tratados pela Lógica Fuzzy, por isso ela é de suma importância no contexto da computação, especialmente na inteligência artificial, que procura representar o raciocínio e o conhecimento humano da forma mais realista possível.

Nos últimos anos, o potencial de manuseio de incertezas e de controle de sistemas complexos tem-se tornado possível com o uso da Lógica Fuzzy. Estes sistemas estão sendo combinados com métodos de *Machine Learning*, que possuem características de adaptação e aprendizado. A palavra certa para isso é simbiose, que vem gerando novas classes de sistemas e de controladores neurodifusos, combinando, dessa forma, os potenciais e as características individuais em sistemas adaptativos e inteligentes (GOMIDE; GUDWIN; TANSCHEIT, 1988).

Assim, matematicamente, um conjunto é definido como uma coleção finita, infinita, ou contável infinita de elementos. Em cada caso, cada elemento é um membro do conjunto ou não, no entanto, na Teoria dos Conjuntos Fuzzy, todos os elementos do universo considerado pertencem a conjuntos fuzzy, com graus possivelmente diferentes. Assim, a resposta à pergunta: “x é membro de um conjunto A” não tem uma resposta definitiva, como verdadeiro ou falso.

Um conjunto fuzzy A definido no universo de discurso $U \neq \emptyset$ é caracterizado por uma função de pertinência $\mu_A : U \rightarrow [0, 1]$, e definido pela expressão:

$$A = \{(x, \mu_A(x)) \mid \ x \in U \wedge \mu_A(x) \in [0, 1]\}$$

Para cada $x \in X$, $\mu_A(x)$ representa o grau de pertinência de x em A.

$$x \in (A, \mu) \Leftrightarrow x \in A \wedge \mu_A(x) \in [0, 1]$$

Desse modo, pela função de pertinência cada elemento $x \in U$ tem um grau de pertinência em cada conjunto A, expressando o quanto é possível para o elemento x pertencer ao conjunto A. Assim, quando o grau de pertinência de um elemento em um conjunto fuzzy for igual a 0, na lógica clássica significa que ele não pertence ao conjunto. E, na construção dual, se um elemento tem grau de pertinência 1 em um conjunto fuzzy, significa que na abordagem clássica este elemento pertence ao conjunto.

Sistema Baseado em Regras Fuzzy Tipo-1

Um Sistema Baseado em Regras Fuzzy (SBRF), é um sistema que se utiliza da Lógica Fuzzy para produzir saídas para cada entrada desse sistema. A entrada e a saída representam a “condição” e a “ação", respectivamente e, diante dessa conotação para as entradas e saídas, os SBRF são denominados de Controladores Fuzzy (DUBOIS; PRADE, 2012).

Geralmente, para um sistema fuzzy qualquer, a cada entrada deve-se corresponder uma saída. E para o desenvolvimento desse sistema, são necessários quatro módulos, que compõem o Controlador Fuzzy. Os módulos são: fuzzificação, base de regras, inferência fuzzy e de defuzzificação. A funcionalidade de cada um será descrita a seguir.

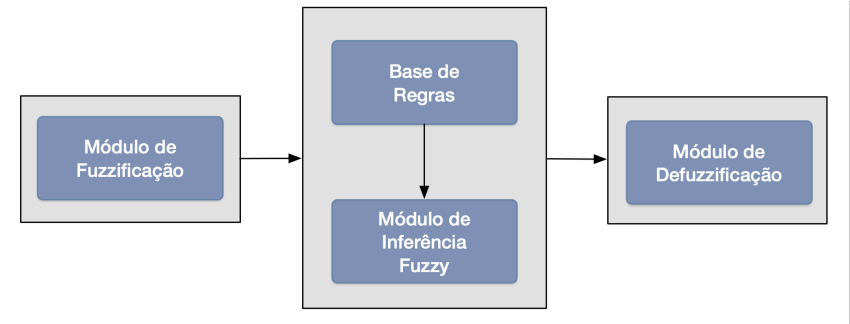

Figura 2 – Módulos de um Controlador Fuzzy Tipo-1

Fuzzificação

A fuzzificação, também conhecida por “Processador de Entrada”, é o módulo responsável por modelar as entradas do sistema por conjuntos fuzzy. As funções de pertinência são concebidas para cada conjunto fuzzy e podem ser construídas de três maneiras diferentes (GOMIDE; GUDWIN; TANSCHEIT, 1988): (i) analíticas; (ii) lineares por partes, resultando em formas triangulares ou trapezoidais; e (iii) discretizadas. Para a seleção de serviços usando-se algoritmos fuzzy, a maior parte dos trabalhos na literatura usam funções de associação triangular ou trapezoidal (ALMULLA; YAHYA-

OUI; AL-MATORI, 2015). A função de associação trapezoidal com os três valores difusos baixo (ruim), médio (normal) e alto (bom), com suas interseções específicas, geram cinco regiões que representam a opinião do especialista sobre o valor da variável linguística em questão (por exemplo, relevância do serviço com as necessidades do usuário).

Base de Regras

Os Sistemas Baseados em Regras Fuzzy são executados mais rapidamente do que os sistemas convencionais baseados em regras, pois são fáceis de entender, ler, adicionar, modificar serviços, apresentam conhecimento natural em representação uniforme e também separam conhecimento do processamento, reduzindo a complexidade do tempo (PRIYA; CHANDRAMATHI, 2014).

O módulo da Base de Regras por sua vez, é composto por proposições fuzzy, sendo que cada uma destas, pode ser representada, numa construção de múltiplos dados de entrada (RT, AV e TH) em um único dado de saída (Weight) da seguinte forma:

If RT is Low and AV is Low and TH is Low then Weight is Low

Esses dados de entrada e saída, os quais correspondem a valores associados a variáveis linguísticas, são modelados por funções de pertinência.

Inferência Fuzzy

O módulo de Inferência Fuzzy apresenta o objetivo de “traduzir" matematicamente cada proposição fuzzy por meio de técnicas da Lógica Fuzzy. Nesse módulo são definidos quais procedimentos e regras de inferência serão selecionadas para se obter a relação fuzzy que modela a Base de Regras. O módulo também fornece o controle fuzzy a partir de cada entrada.

O Método de Inferência de Mamdani, desenvolvido por Ebrahim Mandani (MAMDANI; ASSILIAN, 1975), é baseado na regra de composição de inferência max-min. O processamento das regras é realizado da seguinte forma: (i) para cada regra M_j, da base de regras, a condição “se x é A_j então u é B_j", é modelada pela aplicação de uma relação fuzzy; (ii) utiliza-se a t-norma $\wedge$ (mínimo) para o conectivo lógico “e"; (iii) utiliza-se a t-conorma $\vee$ (máximo) para o conectivo lógico “ou", sendo que este liga as regras fuzzy da base de regras.

Considerando-se um controlador fuzzy com duas entradas e uma saída, conforme a duas regras a seguir: Se x_1 é A_{11} e x_2 é A_{12} então u é B_1; e Se x_1 é A_{21} e x_2 é A_{22} então u é B_2, têm-se graficamente as saídas parciais e a saída final do controlador fuzzy de Mamdani, respectivamente na Figura 3 e na Figura 4.

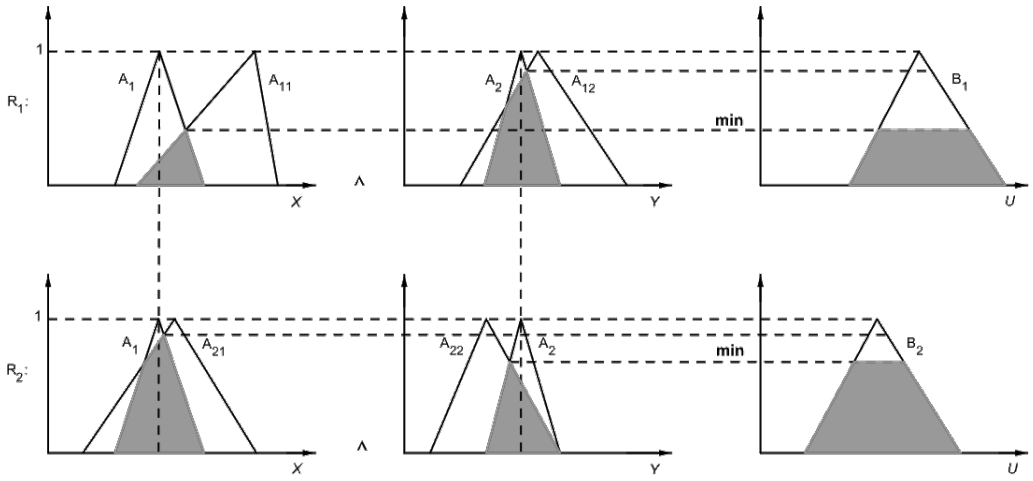

Figura 3 – Controlador Fuzzy Mamdani - Saídas Parciais das Regras
Fonte: BARROS; BASSANEZI, 2010

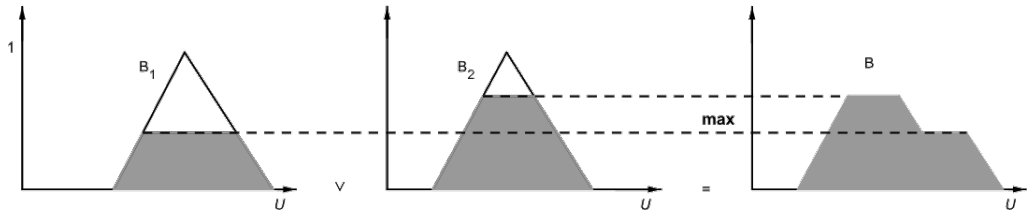

Figura 4 – Controlador Fuzzy Mamdani - Saída Final das Regras
Fonte: BARROS; BASSANEZI, 2010

O gráfico da Figura 4 representa a função de pertinência μ_B que foi gerado através do conectivo $\vee$, t-conorma do máximo.

De um modo formal, a relação fuzzy M é o subconjunto de $X \times U$ e sua função de pertinência é representada por

$$\mu_M(x,u) = \max_{1\leq j\leq r}[\mu_{R_j}(x,u)] = \max_{1\leq j\leq r}[\mu_{A_j}(x) \wedge \mu_{B_j}(u)],$$

no qual R representa o número de regras que compõem a base de regras e A_j e B_j são os subconjuntos da regra j. Cada um desses valores $\mu_{A_j}(x)$ e $\mu_{B_j}(u)$ são interpretados como graus com que x e u pertencem aos subconjuntos A_i e B_i respectivamente, e M é a união dos produtos cartesianos fuzzy.

Defuzzificação

O módulo de Defuzzificação, também conhecido como Processador de Saída, funciona de modo que a cada entrada fuzzy, o módulo de inferência produz uma saída fuzzy. Se a entrada for um número real, espera-se que a saída também seja um número real, mas, geralmente, isso não ocorre em controladores fuzzy pois, mesmo a entrada sendo crisp, a saída é fuzzy. Com isso, deve-se utilizar um método para defuzzificar a saída e gerar um número real.

Entre vários métodos de defuzzificação, um dos mais utilizados é o Método do Centro de Gravidade $(G(Y))$, também conhecido como Centroide. Esse método transforma a saída fuzzy em saída discreta, encontrando-se a média aritmética entre os centros de gravidade dos conjuntos fuzzy aos quais o elemento pertence, ponderados pelo grau de pertinência. Então, para um domínio discreto, tem-se a seguinte equação:

$G(Y) = \frac{\sum_{i=1}^{n} \mu_i Y_i}{\sum_{i=1}^{n} \mu_i}$, onde: G(Y) é a saída discreta; n é a quantidade de conjuntos fuzzy a que o elemento pertence; μ_i é o grau de pertinência com que o elemento pertence ao i-ésimo conjunto fuzzy; e Y_i é o centro de gravidade do i-ésimo conjunto fuzzy a que o elemento pertence.

Sistema Baseado em Regras Fuzzy Tipo-2

A Lógica Fuzzy Tipo-2, foi introduzida por Lotfi A. Zadeh como uma extensão da Lógica Fuzzy Tipo-1 (ZADEH, 1971, 1975). A origem da Lógica Fuzzy Tipo-2 está relacionada com a insuficiência da Lógica Fuzzy Tipo-1 em modelar as incertezas referentes à definição das funções de pertinência dos antecedentes e consequentes em um sistema de inferência fuzzy.

Na Lógica Fuzzy Tipo-2 o grau de pertinência de um elemento pode ser dado por outro conjunto fuzzy. Neste novo contexto, mesmo que a função de pertinência Tipo-2 seja precisa, ela possui uma “mancha" de incerteza (FOU $-$ *Footprint of Uncertainty*). A Figura 5, a seguir, exibe a diferença de uma função de pertinência de um conjunto fuzzy triangular Tipo-1 do Tipo-2.

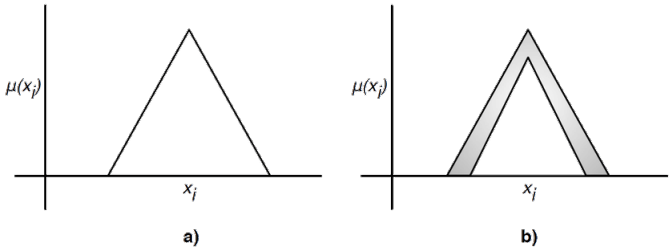

Figura 5 – Conjuntos Fuzzy Tipo-1 e Tipo-2
Fonte: CASTAÑÓN-PUGA et al., 2015

Para representar um conjunto Fuzzy Tipo-2 bidimensionalmente, é utilizada a mancha de incerteza (FOU), que representa a incerteza dos limites dos conjuntos Fuzzy Tipo-2 por meio de uma área desfocada em torno da curva que retrata a função de pertinência principal ou através de uma área hachurada.

A diferença entre sistemas fuzzy do Tipo-1 e do Tipo-2 está associada à natureza

das funções de pertinência e não às regras. Por isso, as regras permanecem as mesmas, tanto para o Tipo-1 quanto para o Tipo-2.

Conjuntos Fuzzy Tipo-2 possuem como grau de pertinência conjuntos Fuzzy Tipo-1 e não apenas um único valor. Esses conjuntos podem ser utilizados em problemas em que existe incerteza referente aos graus de pertinência, incerteza em alguns parâmetros das funções de pertinência ou incerteza no formato dessas funções de pertinência.

Um conjunto fuzzy do tipo-2, $\widetilde{A}$ sobre X, é representado por uma função de pertinência tipo-2, $\mu_{\widetilde{A}}(x,u)$, onde $x \in X$ e $u \in J_x \subseteq [0,1]$, ou seja,

$$\widetilde{A} = \{[(x,u), \mu_{\widetilde{A}}(x,u)] \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\},$$
onde $0 \leq \mu_{\widetilde{A}}(x,u) \leq 1$.

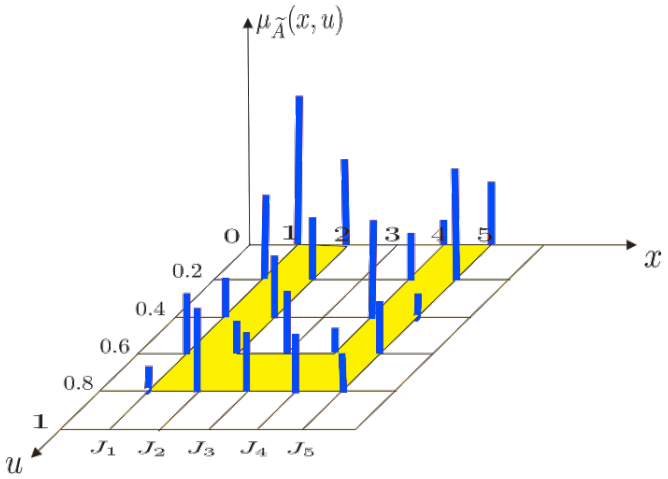

Figura 6 – Função de Pertinência de Conjunto Fuzzy Trapezoidal Tipo-2
Fonte: CABRERA, 2014

A Figura 6 ilustra tridimensionalmente a função de pertinência de um conjunto fuzzy trapezoidal de Tipo-2, facilitando, assim, a caracterização da incerteza, representada pelo eixo vertical e a “mancha" de incerteza caracterizada pela área quadrada, preenchida.

A pertinência primária de X (J_X) é definida como domínio da função de pertinência secundária para um valor de x, com $J_x \subseteq [0,1]$, $\forall x \in X$, onde $J_x = [\underline{J}_x, \overline{J}_x]$. A função de pertinência secundária é definida como o corte vertical $\mu_{\widetilde{A}}(x,u)$ em um determinado valor de $x = x'$, ou seja, é $\mu_{\widetilde{A}}(x = x', u)$ (CABRERA, 2014).

É denominado Grau Primário o valor do domínio de $u_x \in [0,1]$. Já o Grau Secundário é determinado pela amplitude de uma função de pertinência secundária, ou seja, $\mu_{\widetilde{A}}(x,u)$ é o grau de pertinência denominado grau secundário.

Em um conjunto Fuzzy Tipo-2, a incerteza das funções de pertinências primárias é representada pela FOU ou “mancha" de incerteza. Quando essa incerteza some, o conjunto Fuzzy Tipo-2 passa a ser um conjunto Fuzzy Tipo-1. A “mancha"

de incerteza é definida como a união de todas as pertinências primárias, $FOU(\widetilde{A}) = \bigcup_{x \in X}(x, J_x)$.

A FOU de um conjunto Fuzzy Tipo-2 é delimitada por uma função de pertinência tipo-1 superior e inferior. A Figura 7 apresenta a FOU e as funções de pertinência superior e inferior de um conjunto Fuzzy Tipo-2 trapezoidal.

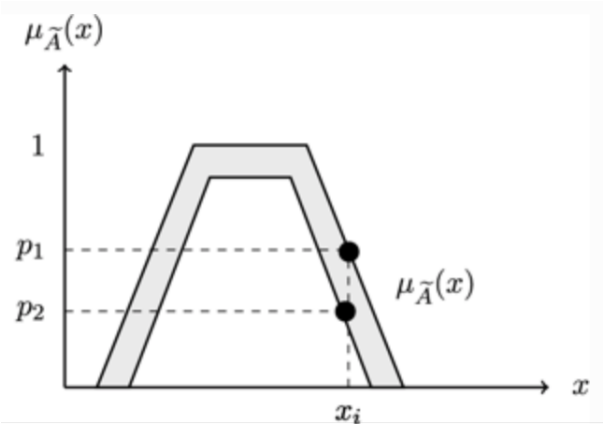

Figura 7 – FOU e Funções de Pertinência Superior e Inferior
Fonte: MENDEL, 2016

A função de pertinência superior é exibida na forma $\overline{\mu}_{\widetilde{A}}(x)$, $\forall x \in X$ e a função de pertinência inferior, função mais interna que limita a *FOU*$(\widetilde{A})$ e dada na forma de $\underline{\mu}_{\widetilde{A}}(x)$, $\forall x \in X$, respectivamente representadas por $\overline{\mu}_{\widetilde{A}}(x) = \bigcup_{x \in X}(x, \overline{J}_x)$, $\underline{\mu}_{\widetilde{A}}(x) = \bigcup_{x \in X}(x, \underline{J}_x)$, onde $\bigcup$ representa a união.

Um conjunto Fuzzy Tipo-2 Intervalar é um conjunto Fuzzy tipo-2 em que todos os graus secundários são 1, ou seja, $\mu_{\widetilde{A}}(x, u) = 1, \quad \forall x \in X, \quad \forall u \in J_x$. A Figura 8 apresenta a função de pertinência triangular de um conjunto Fuzzy Tipo-2 Intervalar.

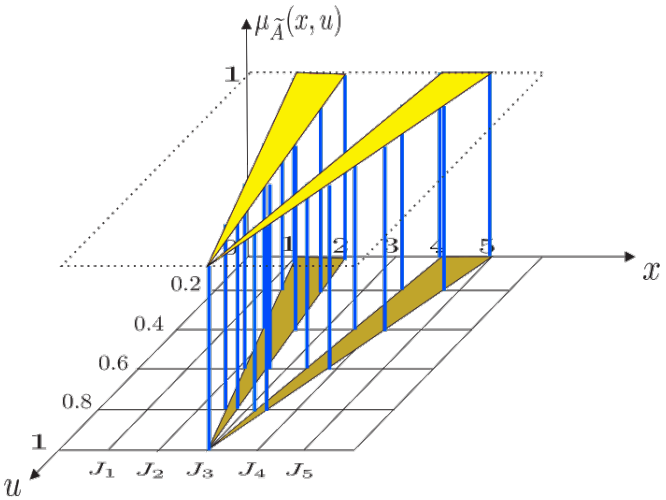

Figura 8 – Função de Pertinência de Conjunto Fuzzy Triangular Tipo-2 Intervalar
Fonte: CABRERA, 2014

O Sistema Baseado em Regras Fuzzy Tipo-2, SBRF2, é aquele em que pelo me-

nos um dos seus conjuntos fuzzy presentes no antecedente ou consequente de uma regra são conjuntos fuzzy do Tipo-2. Esse sistema, o SBRF2, é formado por cinco componentes denominados de fuzzificador, inferência, base de regras, redutor de tipo e defuzzificador. A Figura 9 exibe a estrutura geral desse sistema.

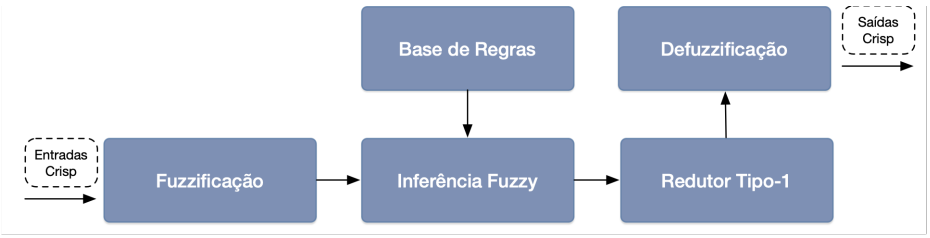

Figura 9 – Módulos de um Controlador Fuzzy Tipo-2

Os componentes utilizados pelo SBRF2 são os mesmos utilizados pelo SBRF, portanto, apenas o componente " Redutor Tipo-1" será apresentado.

O objetivo do "Redutor de Tipo" é transformar um conjunto Fuzzy Tipo-2 em conjunto fuzzy do Tipo-1, ou seja, verifica o conjunto Fuzzy Tipo-1 que melhor representa o conjunto Fuzzy Tipo-2 e que precisa satisfazer à seguinte premissa: quando toda incerteza desaparecer, então o resultado do SBRF2 é reduzido ao SBRF (MENDEL, 2017).

Uma das restrições quanto à aplicação da Lógica Fuzzy Tipo-2 refere-se a este módulo que reduz o resultado final a um número. Embora seja fácil de analisar e comparar, este resultado numérico não é capaz de refletir no resultado a avaliação da incerteza para tomada de decisão baseada em SBRF2.

Isso porque nem sempre conjuntos podem ser comparados, pois são reticulados definidos por ordens parciais. No caso intervalar, $L([0,1]) = ([0,1], \leq)$ tem-se o conjunto de todos os subintervalos do intervalo unitário, considerando a ordem parcial usual: $X \leq_{L([0,1])} Y \Leftrightarrow \underline{X} \leq \underline{Y} \wedge \overline{X} \leq \overline{Y}$, $\forall X = [\underline{X}, \overline{X}], Y = [\underline{Y}, \overline{Y}] \in L([0,1])$.

Para contornar esta situação consideramos o uso de ordens admissíveis em $L([0,1])$.

2.4.4 Ordens Admissíveis

Além do emprego da abordagem de Sistema Baseado em Regras Fuzzy Tipo-2, também considera-se o uso de uma ordem total para intervalos por meio das Ordens Admissíveis, pois, nem sempre uma lista de intervalos gerada com a saída do SBRF2 pode ser comparável através dos métodos convencionais. Uma ordem linear $\leq$ acima de P é uma relação binária transitiva, antissimétrica e total. Equivalentemente, uma ordem linear é uma ordem parcial sob a qual cada par de elementos é comparável.

Alguns exemplos de ordens lineares em L ([0,1]) são as Ordens Lexicográficas, com $\leq_{Lex1}$ relacionado à primeira variável e $\leq_{Lex2}$ para a segunda variável, respectiva-

mente definidas, $\forall X = [\underline{X}, \overline{X}], Y = [\underline{Y}, \overline{Y}] \in L([0,1])$, da seguinte forma:

$$[\underline{X}, \overline{X}] \leq_{Lex1} [\underline{Y}, \overline{Y}] \Leftrightarrow \begin{cases} \underline{X} < \underline{Y}; \text{ ou} \\ \underline{X} = \underline{Y} \text{ e } \overline{X} \leq \underline{Y}; \end{cases}$$

$$[\underline{X}, \overline{X}] \leq_{Lex2} [\underline{Y}, \overline{Y}] \Leftrightarrow \begin{cases} \overline{X} < \overline{Y}; \text{ ou} \\ \overline{X} = \overline{Y} \text{ e } \underline{X} \leq \underline{Y}. \end{cases}$$

Portanto, esta proposta considera a ordem de Xu e Yager (XU; YAGER, 2006) definida, para todos os $X, Y \in \mathbb{U}$, da seguinte forma:

$$[\underline{X}, \overline{X}] \leq_{XY} [\underline{Y}, \overline{Y}] \Leftrightarrow \begin{cases} \underline{X} + \overline{X} < \underline{Y} + \overline{Y}; \text{ ou} \\ \underline{X} + \overline{X} = \underline{Y} + \overline{Y} \text{ e } \overline{X} - \underline{X} \leq \overline{Y} - \underline{Y}. \end{cases} \quad (5)$$

2.5 Considerações Sobre o Capítulo

Este Capítulo discutiu conceitos fundamentais para o desenvolvimento do trabalho de pesquisa desta Tese. Considerando que a IoT disponibiliza inúmeros recursos, muitos com características próximas, é possível inferir que para atender uma demanda específica de um cliente, tais recursos necessitam de um tratamento de, no mínimo, três etapas: serem descobertos, identificados e classificados.

A revisão de literatura apontou que um mecanismo de descoberta de recursos deve atender vários requisitos para adequar-se à dinamicidade e à escalabilidade da IoT. Em especial, considerando-se trabalhos anteriores do grupo de pesquisa, destaca-se a importância da descrição semântica para padronização e compartilhamento das nomenclaturas atribuídas aos recursos. Por sua vez, foi possível identificar que a classificação de recursos que empregam requisitos não funcionais pode ser implementada através da avaliação dos seus atributos de qualidade (QoS).

A discussão das abordagens para classificação de recursos trouxe a perspectiva de que é possível avaliar os atributos de QoS empregando-se Algoritmos de Decisão Multicritério, o que promove a integração de duas frentes de estudo já consolidadas.

A revisão de conceitos referentes à Lógica Fuzzy e *Machine Learning*, no que lhe concerne, consolidam a perspectiva do seu emprego para o tratamento de incertezas e/ou otimização de procedimentos de clusterização de recursos em grupos.

O próximo Capítulo apresenta a Revisão Sistemática da Literatura realizada durante o doutorado. Esta revisão foi desenvolvida com a intenção de selecionar trabalhos que apresentam propostas para classificação de recursos, considerando o tratamento da incerteza.

3 ESTADO DA ARTE EM CLASSIFICAÇÃO DE RECURSOS

Este Capítulo discute a Revisão Sistemática da Literatura (RSL), desenvolvida durante o curso de doutorado. O resultado central desta revisão foi a seleção dos principais trabalhos de pesquisa no tema desta Tese. Outro ganho decorrente da revisão sistemática foi a identificação dos métodos utilizados no tratamento da incerteza durante o processo de classificação de recursos, e que possam ser aplicados na infraestrutura computacional provida pela Internet das Coisas.

3.1 Revisão Sistemática da Literatura

A Revisão Sistemática da Literatura (XIAO; WATSON, 2019) apresentada nesta Tese foi desenvolvida com suporte da ferramenta online Parsifal (Parsifal, 2018), que foi projetada para auxiliar os pesquisadores na condução de revisões sistemáticas da literatura. Desta forma, ela possui todas as etapas expostas por esta Tese, permitindo que os pesquisadores trabalhem colaborativamente de forma mais simples e sem limitação de plataforma.

Esta RSL foi desenvolvida tendo como base cinco etapas principais a serem seguidas: (i) definição de questões de pesquisa; (ii) realização da pesquisa de estudos primários relevantes; (iii) triagem dos documentos; (iv) *keywording* dos resumos; e (v) extração de dados.

A primeira etapa do desenvolvimento de uma revisão sistemática é a definição das questões de pesquisa que fundamentam a mesma. Assim, com a premissa de circunstanciar a identificação dos trabalhos relacionados, foram elaboradas as seguintes questões de pesquisa:

- Quais são os trabalhos atuais e relevantes sobre descoberta e classificação de recursos que consideram os requisitos não funcionais dos recursos?
- Quais são as incertezas a serem consideradas no processo de classificação e quais métodos utilizados para tratá-las?
- Quais algoritmos e técnicas podem ser utilizados no processo de classificação?

A partir da definição dessas questões de pesquisa, foram definidas as palavras-chave que respondam as questões elaboradas. Inicialmente, foram realizados testes com as palavras candidatas a palavras-chave na busca. Tais testes consistiram na combinação dos seguintes termos: (i) descoberta; (ii) ranqueamento; (iii) classificação; (iv) seleção; (v) recursos; (vi) serviços; (vii) QoS; e (viii) incerteza. Após a definição dos termos foram elaboradas as strings para pesquisa nas bases de dados que são apresentadas na Tabela 1, executando-se, assim, o segundo estágio desta pesquisa.

Tabela 1 – RSL - Bases de Dados e Strings de Pesquisa Utilizadas

Base de Dados	**String de Pesquisa**
ACM Digital Library	acmdlTitle:((resource ranking OR service ranking OR resource classification OR service classification OR resource selection OR service selection)) AND ((QoS OR attribute)) AND ((Fuzzy OR Uncertainly OR non-functional))
IEEE Digital Library	((resource ranking OR service ranking OR resource classification OR service classification OR resource selection OR service selection)) AND ((QoS OR attribute)) AND ((Fuzzy OR Uncertainly OR non-functional))
Science@Direct	(((ranking OR classification OR selection)) AND ((services OR resources)) AND ((QoS OR Attribute)) AND ((Fuzzy OR Uncertainly OR non-functional)))
Scopus	TITLE-ABS-KEY(((resource ranking OR service ranking OR resource classification OR service classification OR resource selection OR service selection)) AND ((QoS OR attribute)) AND ((Fuzzy OR Uncertainly OR non-functional)))
Springer Link	(("resource ranking" OR "service ranking" OR "resource classification" OR "service classification" OR "resource selection" OR "service selection")) AND ((QoS OR attribute)) AND ((Fuzzy OR Uncertainly OR non-functional))

A partir da seleção inicial realizada nas cinco bases de dados, foram elaboradas etapas, atendendo-se a critérios para inclusão ou exclusão dos artigos.

A Figura 10 apresenta o fluxo de execução da RSL.

3.1.1 Critérios de Inclusão

Foram definidos os seguintes critérios de inclusão:

- trabalhos que propusessem estratégias para classificação de recursos em grande escala, considerando-se atributos de QoS;
- emprego de técnicas para tratamento da incerteza no processo de classificação de recursos;
- trabalhos publicados entre 2013 e 2018.

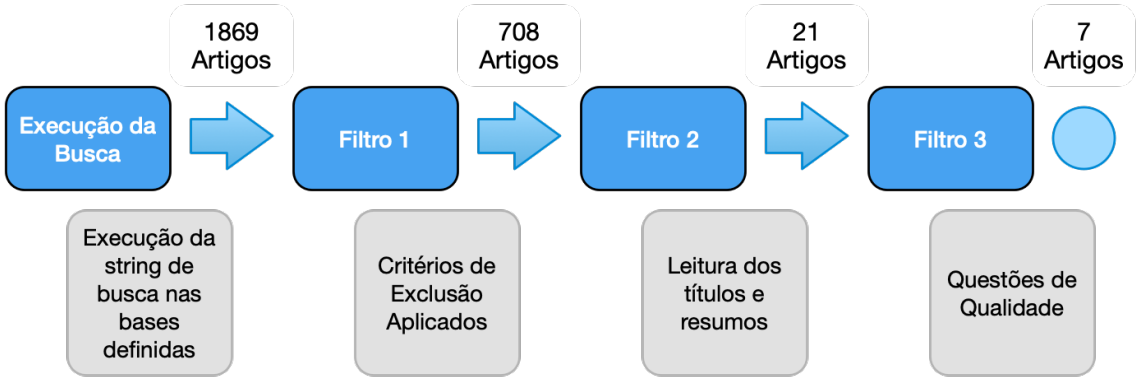

Figura 10 – RSL - Fluxo de Execução

3.1.2 Critérios de Exclusão

Foi aplicada a pesquisa inicial nas bases de dados, utilizando-se as strings de pesquisa, excluindo-se trabalhos que se enquadrassem nos seguintes critérios de exclusão:

- remoção de artigos anteriores a 2013;
- trabalhos que não foram publicados em inglês ou português;
- artigos que não apresentavam a string de busca no resumo;
- artigos duplicados;
- artigos de autoria própria;
- artigos que apresentavam pequenas modificações de outros artigo;
- trabalhos que não eram artigos publicados em eventos ou periódicos;
- trabalhos voltados para protocolos de rede ou infraestrutura de rede;
- trabalhos direcionados à jogos;
- trabalhos com foco em algoritmos genéticos.

Os resultados obtidos na execução da busca em cada base de dados foram submetidos no formato .bib na ferramenta Parsifal. Foram realizadas as seguintes etapas no processo de seleção de artigos:

1. Execução da string de pesquisa na base de dados.
2. Aplicação do “Filtro 1”, ou seja, definição dos critérios de exclusão.

3. Remoção dos artigos duplicados e de autoria própria dentro da ferramenta Parsifal. Após, foi realizada a leitura dos títulos e resumos para aceitar ou rejeitar o artigo.

4. As questões de qualidade são respondidas no resultado do “Filtro 2”, definidas na Tabela 2. Após todas as questões de qualidade terem sido respondidas, foi gerada uma pontuação para o artigo.

Tabela 2 – RSL - Questões de Qualidade para Seleção

ID	Questão de Qualidade
QQ1	Utiliza algoritmos MCDA?
QQ2	Trata de alguma forma a incerteza?
QQ3	Emprega técnicas para predição e/ou pré-classificação de recursos?
QQ4	Emprega tecnologias de Web Semântica?
QQ5	Apresenta resultados da classificação?
QQ6	Utiliza técnicas híbridas (objetivas e subjetivas) de classificação?
QQ7	Considera as preferências do cliente?

5. O “Filtro 3” foi realizado através da definição da pontuação mínima aceitável para a seleção final. Desse modo, foi definida a pontuação mínima para aceite de “4” e pontuação máxima “7”.

6. A última etapa foi realizar a extração de dados dos artigos selecionados, executada dentro da ferramenta Parsifal, considerando-se as seguintes questões:

   (a) Qual o problema que o artigo aborda?

   (b) Qual a solução proposta para o problema?

   (c) Como foi feita a validação?

   (d) Quais são os trabalhos futuros?

O quantitativo de artigos aceitos, por base de dados, em cada etapa desta Revisão Sistemática, é apresentada na Tabela 3.

Tabela 3 – RSL - Seleção por Base de Dados dos Artigos

Base de Dados	Execução da Busca	Filtro 1	Filtro 2	Filtro 3
ACM Digital Library	364	108	6	0
IEEE Digital Library	431	150	8	4
Science@Direct	69	34	1	0
Scopus	75	31	3	0
Springer Link	930	385	3	3

Os artigos selecionados estão relacionados na Tabela 4, a seguir, e estão apresentados com um identificador, título e autores correspondentes.

Tabela 4 – RSL - Seleção Final dos Artigos

ID	Nome do Artigo	Autores
A1	Discovery and prioritization of web services based on fuzzy user preferences for QoS	(GOHAR; PUROHIT, 2016)
A2	Prioritizing the solution of cloud service selection using integrated MCDM methods under Fuzzy environment	(KUMAR; MISHRA; KUMAR, 2017)
A3	Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences	(WANG et al., 2015)
A4	Fuzzy AHP for Learning Service Selection in Context-Aware Ubiquitous Learning Systems	(SALAH; SAADI, 2016)
A5	Combining experts' opinion with consumers' preference in web service QoS selection	(LIU; XIAO; LIN, 2013)
A6	PuLSaR: preference-based cloud service selection for cloud service brokers	(PATINIOTAKIS; VERGINADIS; MENTZAS, 2015)
A7	Non Functional QoS Criterion Based Web Service Ranking	(SUCHITHRA; RAMAKRISHNAN, 2016)

3.2 Trabalhos Relacionados Identificados com a RSL

A última etapa da Revisão Sistemática da Literatura foi a organização das características dos Trabalhos Relacionados que foram identificados, estimando-se o problema de Pesquisa abordado, a proposta de seu tratamento, a abordagem para validação explorada e a indicação de trabalhos futuros. Essa organização está contemplada nas Seções a seguir.

3.2.1 Discovery and Prioritization of Web Services Based on Fuzzy User Preferences for QoS (GOHAR; PUROHIT, 2016)

- **Problema:** desafio da seleção e descoberta de serviços na web, que envolve a correspondência e a busca do serviço web mais adequado a partir de uma grande coleção de serviços web equivalentes funcionais.

- **Proposta:** foi desenvolvida uma abordagem baseada em fuzzy para a descoberta de serviços na web que modelam a classificação de serviços com reconhecimento de QoS como um problema de tomada de decisão multicritério. Para descrever os serviços da Web disponíveis no registro, foi criada uma ontologia para cada serviço e para representar as preferências de Quality of Service (QoS) funcionais e imprecisas, tanto do consumidor do serviço web quanto do provedor em termos linguísticos. A base da regra fuzzy é criada com a ajuda da API Java Expert System Shell (JESS). Para tomar decisões sobre requisitos de QoS

múltiplos e conflitantes, foi adotado o modelo de organização de classificação de preferência melhorada para avaliação de enriquecimentos (PROMETHEE) para a classificação dos serviços.

- **Validação:** para demonstrar as habilidades do *framework*, um sistema baseado na web foi projetado para recrutamento de candidatos. Muitas empresas não conseguem recrutar candidatos adequados para constituir sua equipe de trabalho e, por outro lado, os candidatos também não conseguem obter a colocação que merecem. Assim, foi implementada uma abordagem para o recrutamento de candidatos adequados no local apropriado, criando-se um sistema de Recrutamento Eletrônico. Por exemplo, para definir o conjunto de habilidades necessárias para recrutar o candidato apropriado para um trabalho específico, o recrutador usou termos linguísticos que podem ser representados pelo conjunto fuzzy apropriado para habilidade de comunicação que deve ser muito pobre, pobre, médio, bom, muito bom ou superior. O Recrutador (provedor) também pode descrever o parâmetro de QoS de seus serviços da Web em termos linguísticos, como um conjunto fuzzy para descrever a segurança de um serviço da Web, que deve ser muito baixo, médio, alto ou muito alto.

- **Trabalhos Futuros:** não apresentam trabalhos futuros.

3.2.2 Prioritizing the Solution of Cloud Service Selection Using Integrated MCDM Methods Under Fuzzy Environment (KUMAR; MISHRA; KUMAR, 2017)

- **Problema:** a seleção do serviço em nuvem desempenha um papel crucial em termos de seleção de serviço *on-demand* em uma base de assinatura. Como resultado da ampla disponibilidade de serviços em nuvem com funcionalidades semelhantes, é muito importante determinar qual serviço atende melhor aos desejos e objetivos do usuário.

- **Proposta:** foi projetado um novo modelo de seleção de servidor de nuvem sob o ambiente fuzzy, utilizando-se o processo de hierarquia analítica (AHP) e técnica fuzzy para preferência de ordem por similaridade à solução ideal (TOPSIS). O método AHP é implementado para configurar a estrutura do problema da seleção do serviço da nuvem e para estimular o peso dos critérios, usando-se comparações e o método TOPSIS para a classificação final da solução. No modelo proposto, os requisitos de qualidade de serviço não funcionais são levados em consideração para selecionar o serviço apropriado. Além disso, explora um conjunto de variáveis linguísticas pré-definidas, parametrizadas por números fuzzy triangulares para avaliar cada peso de critérios.

- **Validação:** os resultados calculados foram comparados com outros métodos MCDM existentes, como AHP e TOPSIS melhorado.

- **Trabalhos Futuros:** analisar a integração do método Fuzzy TOPSIS com diferentes sistemas de pesos difusos para o problema da seleção do serviço na nuvem.

3.2.3 Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences (WANG et al., 2015)

- **Problema:** existe um número crescente de provedores de serviços que oferecem funcionalidades semelhantes. A seleção de serviços com as propriedades não funcionais desejadas do usuário (NFPs) tem importância significativa, mas desencadeia uma série de problemas de pesquisa relacionados à grande quantidade de dados. Em primeiro lugar, a decisão de seleção deve lidar com um grande volume de dados NFP de serviços. Em segundo lugar, a seleção de serviços precisa refletir diversas preferências de usuários, tanto qualitativas como quantitativas. Em terceiro lugar, a incerteza da rede e da carga do serviço leva a uma grande variabilidade nos NFPs. Em quarto lugar, à medida que os valores de confiança dos NFPs de serviço são coletados através dos comentários históricos do usuário, traz a dimensão de veracidade aos NFPs dos serviços. Em quinto lugar, os objetivos de decisão múltiplos e às vezes conflitantes para uma seleção de serviço ideal devem ser equilibrados.

- **Proposta:** foi abordada a preferência heterogênea e a seleção de serviços com base na confiança. Desenvolveram uma nova abordagem de otimização multiobjetivo para tomar uma decisão de compensação entre o valor de confiança do serviço e a preferência QoS do usuário para classificar os serviços da nuvem em busca dos graus de correspondência com os requisitos dos usuários.

- **Validação:** através de experimentos para avaliar a abordagem de seleção de serviços proposta utilizando-se o *dataset* QWS.

- **Trabalhos Futuros:** devem ser integrados métodos para impedir *feedbacks* mal intencionados na seleção de serviços baseados em confiança e reputação. Abordagens de filtragem colaborativa relacionadas ao perfil de usuário e ao perfil de uso devem ser levados em consideração na seleção de serviços. O trabalho está sob a hipótese de que os usuários forneceriam informações suficientes que contivessem a preferência do usuário e a avaliação dos serviços para ajudar a tomar decisões. Uma questão importante é como promover os usuários para dar *feedback* sobre os serviços que eles invocaram e como obter essas informações na seleção de serviços convenientemente e eficazmente.

3.2.4 Fuzzy AHP for Learning Service Selection in Context-Aware Ubiquitous Learning Systems (SALAH; SAADI, 2016)

- **Problema:** a seleção de serviços em ambientes de aprendizagem é baseada nos requisitos e necessidades QoS do aluno. No entanto, estes últimos são baseados em termos linguísticos vagos que devem ser representados por uma teoria de conjuntos fuzzy.

- **Proposta:** foi proposta uma abordagem de seleção de serviço AHP fuzzy que considera a identificação da situação de aprendizagem, avalia a qualidade do contexto (QoC) e a qualidade do serviço (QoS) e satisfaz os requisitos e necessidades de qualidade do aluno. O uso do método de AHP fuzzy permitiu assegurar uma seleção melhorada do serviço de aprendizagem, levando em consideração vários critérios de avaliação.

- **Validação:** foi simulado um cenário de aprendizagem ciente de contexto em um campus universitário para o qual foi desenvolvido um módulo funcional para ilustrar o papel e a importância da QoC e QoS na seleção de serviços de aprendizagem ubíquos com contexto. No estudo de caso de "configuração do endereço TCP/IP de uma impressora" para auxiliar os alunos a aprender a instalar e configurar a impressora IP, fornecendo-lhes os recursos adequados ao seu contexto e satisfazendo suas necessidades, assegurando a qualidade do contexto detectado e o serviço selecionado.

- **Trabalhos Futuros:** otimização do processo de seleção de serviços de aprendizagem, considerando-se o *feedback* dos alunos e as experiências passadas em uma seleção de raciocínio baseado em casos.

3.2.5 Combining Experts' Opinion with Consumers' Preference in Web Service QoS Selection (LIU; XIAO; LIN, 2013)

- **Problema:** incerteza na seleção do melhor serviço que atenda à requisição do cliente, considerando as definições dos especialistas e as preferências do cliente.

- **Proposta:** foi proposto um modelo de decisão multicritério fuzzy para resolver o problema de seleção de QoS. Leva em conta a opinião dos especialistas como base de dados para tomada de decisão e considera também a preferência do consumidor para lhe oferecer o melhor serviço. O *feedback* dos consumidores também é considerado para ajudar a fazer a escolha certa.

- **Validação:** foi simulado o método proposto utilizando-se quatro especialistas e três serviços com 16 atributos de QoS.

- **Trabalhos Futuros:** não apresentaram trabalhos futuros.

3.2.6 PuLSaR: Preference-based Cloud Service Selection for Cloud Service Brokers (PATINIOTAKIS; VERGINADIS; MENTZAS, 2015)

- **Problema:** a avaliação e seleção dos serviços da nuvem desejados é uma tarefa incômoda para os consumidores de serviços. Portanto, há uma necessidade crescente de orientação e intermediação do usuário durante o processo de seleção do serviço, mas também durante o consumo do serviço na nuvem que sempre deve se referir à melhor escolha possível com base nas preferências do usuário.

- **Proposta:** foi discutido o cLoud Service Recommender baseado em preferências (PuLSaR) que usa uma abordagem holística de decisão multicritério (MCDM) para oferecer otimização como um serviço broker. Os detalhes de especificação e implementação deste mecanismo de software proposto são discutidos minuciosamente enquanto o método de fundo usado é resumido. Tanto o método como o serviço broker permitem a avaliação multiobjetivo de serviços da nuvem de forma unificada, levando em consideração métricas precisas e imprecisas e lidando com a sua falta de expressão. Lidam com a confusão de métricas imprecisas, no sentido de que essa abordagem trata de preferências linguisticamente expressadas e características de serviços em nuvem que não possuem um valor fixo ou preciso e que implicam em um nível de imprecisão que só pode ser capturado usando a Teoria dos Conjuntos Fuzzy de Zadeh.

- **Validação:** foi realizada uma série de experimentos e mediram certos indicadores-chave de desempenho (KPI's) que avaliam o comportamento do PuLSaR. Essas experiências envolvem um número cada vez maior de perfis de preferências de consumidores fabricados (ou seja, 1-1000) que desencadeiam PuLSaR para emitir recomendações para encontrar as ofertas normais da nuvem em um mercado de nuvem hipotético. O tamanho das ofertas disponíveis também varia (ou seja, 1-1000), enquanto os atributos linguísticos (imprecisos) são usados para descrever as ofertas, uma vez que exigem operações mais complexas do que atributos numéricos durante o processamento. Os dois aspectos da variante (perfis e serviços na nuvem, mantendo os atributos selecionados fixados em 10) constituem oito experimentos. Para cada uma dessas variantes, a CPU, a memória e o tempo consumido para a emissão de recomendações PuLSaR foram medidos. Especificamente, essas 8 experiências se agruparam em duas séries de experiências.

- **Trabalhos Futuros:** avaliação das capacidades de otimização do PuLSaR em cenários com brokers reais usando plataformas de hospedagem heterogêneas

e enriquecendo-os com valiosas capacidades dos brokers. As capacidades do PuLSaR serão aprimoradas em plataformas de nuvem reais que aspiram a adicionar funcionalidades broker em suas ofertas. Além disso, PuLSaR será estendido para suportar vários nós de processamento PuLSaR funcionando em paralelo, aumentando ainda mais a escalabilidade. Eventualmente, as extensões do método proposto são consideradas na direção de reduzir o número de comparações em nuvem oferecidas por esta e o cálculo mais rápido de pesos relativos.

3.2.7 Non Functional QoS Criterion Based Web Service Ranking (SUCHITHRA; RAMAKRISHNAN, 2016)

- **Problema:** como vários serviços da Web existem na Internet, a seleção do serviço apropriado é vital em muitos aplicativos de serviços dessa rede. A qualidade de serviço (QoS) é o parâmetro predominante usado para selecionar um serviço em termos de qualidade. Com base na preferência do usuário pela qualidade do serviço, os serviços são classificados e os melhores, selecionados. No entanto, a complexidade com essa abordagem reside no fato que é difícil definir precisamente a propriedade de QoS.

- **Proposta:** foi apresentado um algoritmo de tomada de decisão multiatributo fuzzy para seleção de serviços é apresentado neste trabalho. O método proposto coleta periodicamente o *feedback* do usuário para atualizar a classificação de QoS do serviço.

- **Validação:** os autores prototiparam a abordagem em linguagem Python e *framework* Django. A interface é de tal forma que os parâmetros necessários são claramente explicados quando a página é visualizada pelo usuário sem qualquer explicação adicional. O Django fornece um controlador MVC que ajuda a separar a lógica de negócios da apresentação. Isso facilita o processo de design para manipular alterações dinâmicas em todo o aplicativo da web. Na lógica de classificação, usa a atualização dinâmica das restrições de QoS do serviço. Essas mudanças dinâmicas são baseadas no *feedback* do usuário em relação ao serviço.

- **Trabalhos Futuros:** no futuro, o processo de validação de QoS deve ser automatizado, o que tornará o processo de seleção mais fácil, sem qualquer intervenção humana.

3.3 Análise dos Trabalhos Relacionados

A Tabela 5 apresenta uma comparação dos Trabalhos Relacionados que foram selecionados com a RSL. Os seguintes critérios foram analisados: (i) algoritmo(s) MCDA utilizado(s) como base; (ii) como é realizado o tratamento da incerteza; (iii) expressividade na representação de recursos; (iv) emprego ou não de técnicas para pré-classificação de recursos; e (v) utilização ou não de termos linguísticos para expressar preferências e/ou opiniões do cliente no processo de classificação.

Tabela 5 – RSL - Comparação dos Trabalhos Relacionados

Trabalhos	Algoritmos	Incerteza	Exp.	Pré-classif.	Termos Ling.
A1	PROMETHEE	Fuzzy	Sim	Não	Não
A2	AHP, TOPSIS	Fuzzy	Não	Não	Não
A3	AHP, ANP	Não	Não	Não	Não
A4	AHP	Fuzzy	Não	Sim	Sim
A5	Multi-criteria Group Decision	Intuitionistic Fuzzy	Não	Não	Sim
A6	AHP	Fuzzy	Sim	Não	Sim
A7	SAW	Fuzzy	Não	Não	Não

Pode-se observar que apenas o trabalho A3 (WANG et al., 2015) não aborda técnicas para tratar a incerteza. Todos os outros trabalhos tratam a incerteza com o emprego de Lógica Fuzzy.

O algoritmo MCDA AHP foi utilizado por vários trabalhos: A2 - (KUMAR; MISHRA; KUMAR, 2017), A3 - (WANG et al., 2015), A4 - (SALAH; SAADI, 2016) e A6 - (PATINIOTAKIS; VERGINADIS; MENTZAS, 2015). Apenas o A5 (LIU; XIAO; LIN, 2013) não utilizou um algoritmo MCDA tradicional, optando por um classificador fuzzy proposto pelos autores.

A utilização de tecnologias de Web Semântica foi observada nos trabalhos de A1 (GOHAR; PUROHIT, 2016) e A6 (PATINIOTAKIS; VERGINADIS; MENTZAS, 2015).

Somente o trabalho A4 (SALAH; SAADI, 2016) propõe uma pré-classificação de recursos, que seleciona os recursos apenas pelos requisitos funcionais. Caso o resultado não satisfaça a requisição, então são avaliados os requisitos não funcionais na classificação final.

Os trabalhos de A4 (SALAH; SAADI, 2016), A5 (LIU; XIAO; LIN, 2013) e A6 (PATINIOTAKIS; VERGINADIS; MENTZAS, 2015) utilizam Termos Linguísticos na especificação das preferências e/ou avaliação do cliente no processo de classificação.

3.4 Considerações Sobre o Capítulo

O estudo dos Trabalhos Relacionados que foram identificados com a realização da Revisão Sistemática da Literatura possibilitou identificar as técnicas atualmente utilizadas para os procedimentos de classificação de recursos, bem como os respectivos mecanismos utilizados para o tratamento de incertezas inerentes a estes procedimentos.

A partir desta Revisão foi identificada como principal técnica para tratamento de incertezas o emprego de Lógica Fuzzy e a decorrente utilização de variáveis linguísticas para representar a informação. Outrossim, foi possível identificar que classificação de recursos considerando os seus atributos de QoS, via de regra acontece com o emprego de Algoritmos de Decisão Multicritério.

No próximo Capítulo, é apresentada a concepção do EXEHDA-RR, tendo por base tanto a apropriação conceitual realizada, como a discussão relacionada à literatura da área.

4 EXEHDA-RR: CONCEPÇÃO E FUNCIONALIDADES

Este Capítulo apresenta a concepção do EXEHDA-RR (EXEHDA-*Resource Ranking*), com destaque a sua arquitetura de software, projetada com o intuito de classificar recursos descobertos na IoT, perseguindo a premissa de que sejam os mais adequados para atender as requisições da aplicação de um determinado cliente.

A partir dos Trabalhos Relacionados identificados pela RSL discutida no Capítulo 3, foi articulado o esforço de sistematização das funcionalidades do EXEHDA-RR. Este esforço teve por base as técnicas atualmente empregadas no tratamento da incerteza quando da classificação de recursos, bem como a perspectiva das elevadas escalabilidade e dinamicidade da IoT. A revisão de literatura foi o foco central para a concepção e amadurecimento das características propostas para o EXEHDA-RR, considerando sua atuação em cenários computacionais modernos.

O problema de pesquisa se desdobra nos seguintes aspectos: (i) como gerenciar a grande quantidade de recursos típica da IoT; (ii) como tratar a incerteza do cliente na especificação das suas preferências; (iii) como considerar os diferentes atributos de qualidade dos recursos no processo de sua classificação; e (iv) como promover a inclusão de novos recursos, considerando aspectos de escalabilidade.

4.1 Arquitetura de Software

O EXEHDA-RR é uma proposta para classificação de recursos descobertos que, dentre as suas premissas de pesquisa, está atender as demandas do cenário da IoT. Sua concepção considerou aspectos gerais da organização do middleware EXEHDA, em particular, do ambiente ubíquo provido por ele, cuja organização é baseada em células de execução (YAMIN, 2004), (LOPES, 2016), (SOUZA, 2017).

A literatura aponta que a organização da infraestrutura computacional baseada em células tem sido uma tendência na IoT (ATZORI; LERA; MORABITO, 2018). Tendo isso em conta, entende-se que o EXEHDA-RR tem potencial de ser explorado em outras propostas de middleware.

A arquitetura de software do EXEHDA-RR considera a dinamicidade com que os re-

cursos entram e saem do ambiente computacional, controlando, portanto, a presença de dispositivos, inclusive com baixo poder computacional.

O aumento da expressividade na consulta por recursos foi provida pelo emprego de tecnologias baseadas em Web Semântica, com o intuito de potencializar a descrição dos recursos existentes no ambiente.

A arquitetura do EXEHDA-RR foi concebida tendo por base a abstração "Componente de Software", de forma análoga a outros esforços de pesquisa relacionados ao *middleware* EXEHDA. Esta organização, dentre outros aspectos, contribui para a modularidade da arquitetura, o que é oportuno para futuros esforços de expansão. Outro aspecto a ser destacado é que esta abordagem também tem o potencial de promover para uma melhor relação entre concepção e desenvolvimento, pois estimula aspectos de coesão quanto às funcionalidades envolvidas.

Nesse sentido, a arquitetura de software do EXEHDA-RR (Vide Figura 11) foi concebida tendo por base três componentes principais: (CC) Componente Cliente, (CR) Componente Recurso e (CD) Componente Diretório.

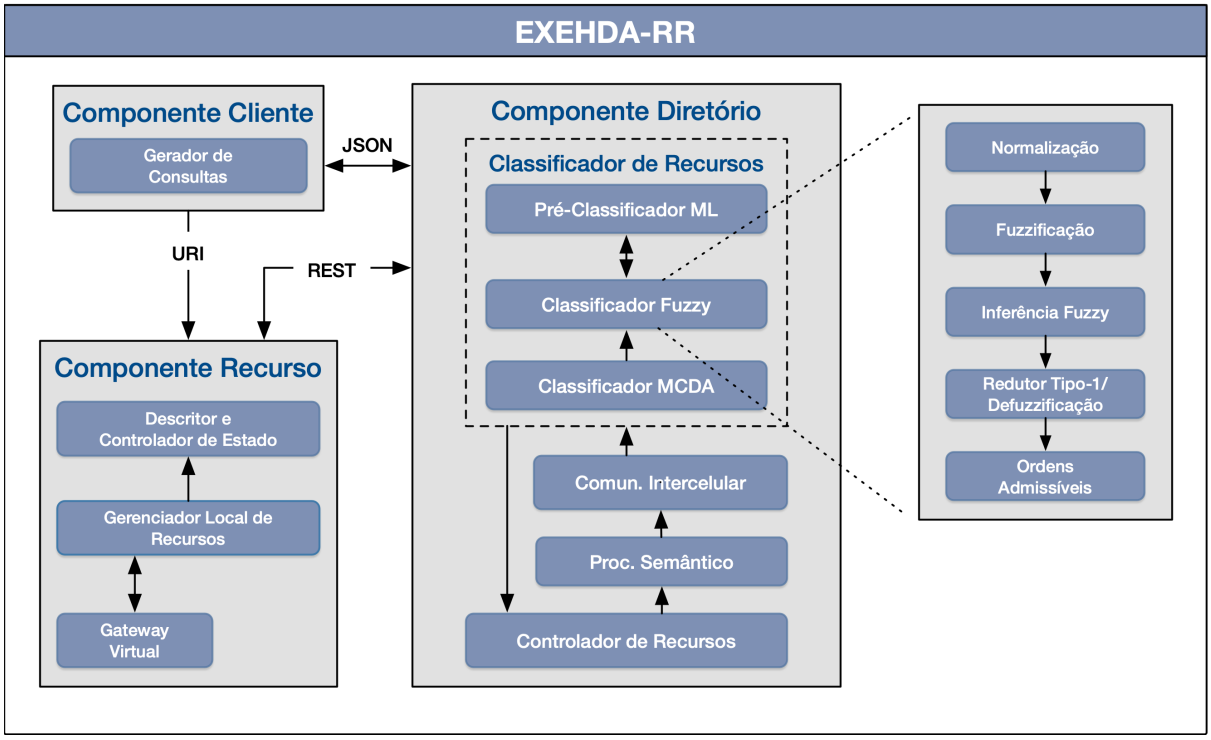

Figura 11 – Visão Geral da Arquitetura de Software do EXEHDA-RR

4.1.1 CC - Componente Cliente

O Componente Cliente (CC) é responsável pelo disparo da consulta por recursos, empregando critérios pertinentes às demandas do cliente. Esse componente que se comunica diretamente com as aplicações que fazem operações de consultas afim de localizar serviços oferecidos pelo recursos presentes no ambiente computacional ge-

renciado pelo middleware EXEHDA. Este componente possui um módulo Gerador de Consultas, responsável por interpretar a descrição dos recursos necessários às aplicações no formato da linguagem SPARQL. Essa comunicação se dá por meio de métodos que seguem o padrão arquitetural REST, que organizam as informações recebidas no formato JSON, incluindo as URIs necessárias para as aplicações acessarem os recursos.

4.1.2 CR - Componente Recurso

O Componente Recurso (CR) é responsável por notificar a presença do recurso. Isso é feito por troca de mensagens com o CD, dentro de um intervalo de tempo. Quando o recurso ingressa pela primeira vez em uma célula, o CR envia a descrição dos seus recursos para o CD da célula onde está localizado o recurso. Este, por sua vez, gera uma instância do recurso na ontologia local. O CR é composto pelos módulos Descritor e Controlador de Estado, e pelo Gerenciador Local de Recursos.

O módulo Descritor e Controlador de Estado é responsável por enviar mensagens ao CD com as características do recurso e anunciar sua presença no ambiente. O módulo Gerenciador Local de Recursos tem por objetivo realizar a descoberta de *Gateways* através do protocolo UPnP. Os tipos de *Gateways* e Componentes do *middleware* EXEHDA são apresentados no Anexo A.

O CR é ativado em nodos EXEHDABorda, cuja finalidade é anunciar a presença dos dispositivos confinados que estão conectados aos *Gateways* da célula.

4.1.3 CD - Componente Diretório

O Componente Diretório (CD), instanciado no EXEHDABase, é formado por três módulos:

1. Controlador de Recursos - esse módulo tem 5 objetivos: (i) realizar a manutenção do estado dos recursos; (ii) gerenciar o Repositório de Recursos Ativos (RRA); (iii) gerenciar o Repositório de Recursos Negados (RRN); (iv) receber as consultas dos Componentes Cliente; e (v) receber as mensagens enviadas pelos Componentes Recurso. O *lease* é um intervalo de tempo, gerenciado pelo Controlador de Recursos. O Componente Recurso precisa renovar, periodicamente, o *lease* do recurso localizado no CD, caso contrário, será removido do Repositório de Recursos Ativos (RRA). Quando se referir aos recursos presentes no ambiente computacional, a modelagem parte da premissa de que todo e qualquer recurso, quando descoberto pela primeira vez, tem suas características extraídas de seu arquivo de configuração para armazenamento na ontologia. Uma vez presente no ambiente, há troca de mensagens entre o Componente Diretório e o Componente Recurso, garantindo que todos os recursos que não tenham seu *lease* renovado no período previsto, sejam desativados.

A modelagem proposta por este módulo, defende o uso de três possíveis valores para o estado atual de um recurso presente no ambiente:

- Inativo - estado de um recurso que se encontra no ambiente mas não teve seu estado renovado no limite de tempo previsto para sua renovação na arquitetura.
- Ativo - estado de um recurso que se fez descoberto recentemente pela arquitetura e que pode ser acessado por aplicações.
- Negado - estado de recurso definido de forma manual na arquitetura. Neste estado, todo e qualquer recurso neste estado será ignorado pela aplicação.

Para auxiliar com o controle do estado de recursos, são mantidas informações do estado atual de cada recurso presente no ambiente, bem como informações referentes à última data que determinado recurso enviou uma mensagem, autoafirmando-se ativo no ambiente computacional. Por sua vez, o módulo Controlador de Recursos possui um serviço, denominado Controlador de Estado, que utiliza os seguintes repositórios: Repositório de Recursos Ativos (RRA) e Repositório de Recursos Negados (RRN).

O serviço Controlador de Estado é responsável por verificar periodicamente todos os recursos que não tiveram seu estado renovado, removendo-os do Repositório de Recursos Ativos. Esse serviço implementa uma funcionalidade de autoconfiguração, visto que não é necessária a iteração do usuário para realizar a manutenção do estado do recurso.

O RRA é um repositório em base de dados relacional Postgres que mantém armazenados todos os recursos que se encontram ativos no ambiente computacional gerenciado pelo *middleware* EXEHDA, seu identificador e data de última atualização, tempo este utilizado para validação de seu estado atual.

O RRN, de igual modo que o RRA, também é um repositório em base de dados relacional Postgres, com a diferença, que mantém armazenados os recursos no primeiro momento em que eles forem anunciados pelo Componente Recurso, e que não podem ser retornados em consultas realizadas por aplicações externas.

2. Processador Semântico - responsável pelo processamento das consultas em linguagem SPARQL, segue o padrão arquitetural REST com a integração do Apache Jena Fuseki (DILLI et al., 2017). A base de dados TDB realiza a persistência de dados, sendo que o motor de inferência foi configurado para utilizar o raciocinador genérico do Apache Jena. Com isso, é possível definir regras adicionais para aumentar a expressividade nas consultas. Por meio desse módulo, são realizadas as consultas à ontologia e são utilizados raciocinadores com o intuito

de aumentar a expressividade das consultas, gerando, por sua vez, melhores resultados (AZEVEDO, 2017).

O Processador Semântico atual foi aperfeiçoado e sua ontologia (Figura 12) remodelada para fazer uso do servidor Apache Fuseki, base de dados TDB e linguagem de consulta SPARQL.

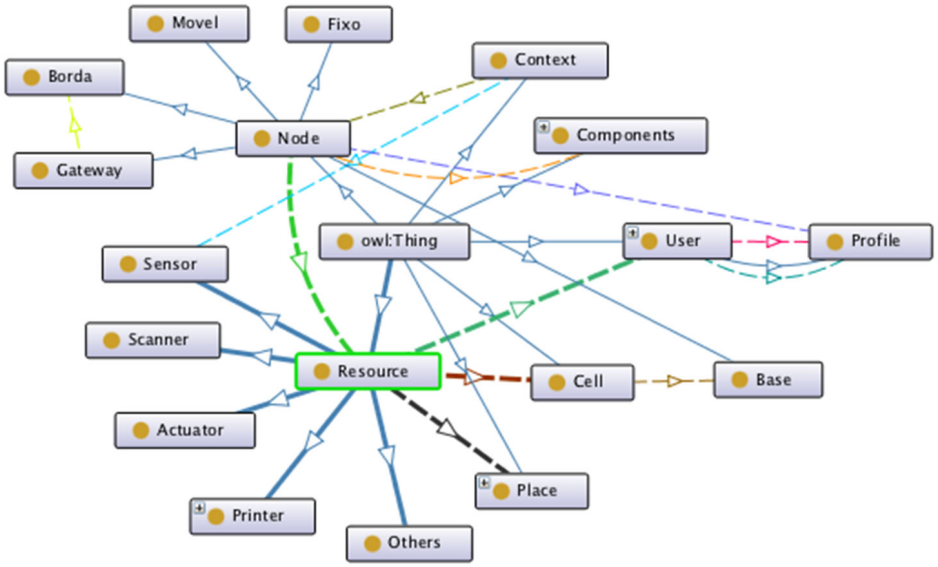

Figura 12 – Ontologia Concebida para o Processador Semântico do EXEHDA-RR

O Apache Fuseki é um componente do Apache Jena que engloba vários de seus componentes, a fim de disponibilizar um servidor SPARQL *standalone*, permitindo que sejam utilizadas as demais funcionalidades providas pelo Apache Jena (HOFEREK, 2012). A ferramenta Jena oferece as seguintes funcionalidades:

- leitura, processamento e escrita de dados RDF em XML, em formato de triplas;
- uma API de ontologia para manipular OWL, com uso de RDF;
- motor de inferência de raciocinadores RDF sobre dados OWL;
- armazenamento de grandes quantidades de triplas RDF, armazenadas em disco;
- motor de consulta com base nas últimas especificações da linguagem SPARQL;
- conjunto de configurações internas que permite que dados RDF sejam publicados por outras aplicações, utilizando uma variedade de protocolos, incluindo SPARQL.

Em adição a todas essas características, o Fuseki permite que as consultas, as atualizações e os uploads de arquivos SPARQL sejam direcionadas a uma base de dados. Também conta com a presença de validadores para consultas e atualizações SPARQL, expandindo a comunicação para formatos além de RDF ou XML (XIN et al., 2013).

O TDB é uma base de dados que armazena informações em pares, contendo chaves e valores, sendo seus dados armazenados em formato binário. No Apache Jena, ele é o componente responsável por armazenar os dados RDF e realizar as consultas por meio da linguagem SQL. O Apache Fuseki admite a realização de procedimentos com a linguagem SPARQL no conjunto de dados presentes no TDB, baseado em padrões REST sobre o protocolo HTTP (XIN et al., 2013).

De acordo com a W3C, SPARQL contém um conjunto de padrões de triplas, denominado padrão básico de grafos. O resultado de dados de consulta SPARQL pode retornar conjuntos de grafos RDF. É considerada uma linguagem para trabalhar com dados RDF pela possibilidade de gerar consultas mais complexas, relacionadas diretamente com expressividade em resultados de buscas (PRUD'HOMMEAUX; SEABORNE, 2008).

3. Comunicador Intercelular - responsável pela comunicação com as células vizinhas. A comunicação entre as células é realizada utilizando tecnologias P2P entre os EXEHDABase de cada célula. A comunicação cliente/servidor ocorre apenas entre os diretórios localizados nos EXEHDABase. Os CC e CR acessam apenas o diretório da célula local. O Comunicador Intercelular é responsável por repassar a pesquisa para as células vizinhas.

4. Classificador de Recursos - tem por finalidade realizar a classificação dos recursos mais adequados à solicitação do cliente. Este módulo contém os seguintes serviços:

    - Classificador MCDA - responsável por classificar todos os recursos descobertos através do algoritmo MCDA proposto.
    - Pré-Classificador *Machine Learning* - por meio deste serviço, os novos recursos recém descobertos são pré-classificados, visando reduzir o custo computacional gerado pelo algoritmo MCDA.
    - Classificador Fuzzy - tem por objetivo resolver as incertezas na especificação e processamento de preferências do cliente.

A Figura 13 ilustra o fluxo de execução dos componentes do EXEHDA-RR e a atuação do módulo Classificador de Recursos e seus serviços.

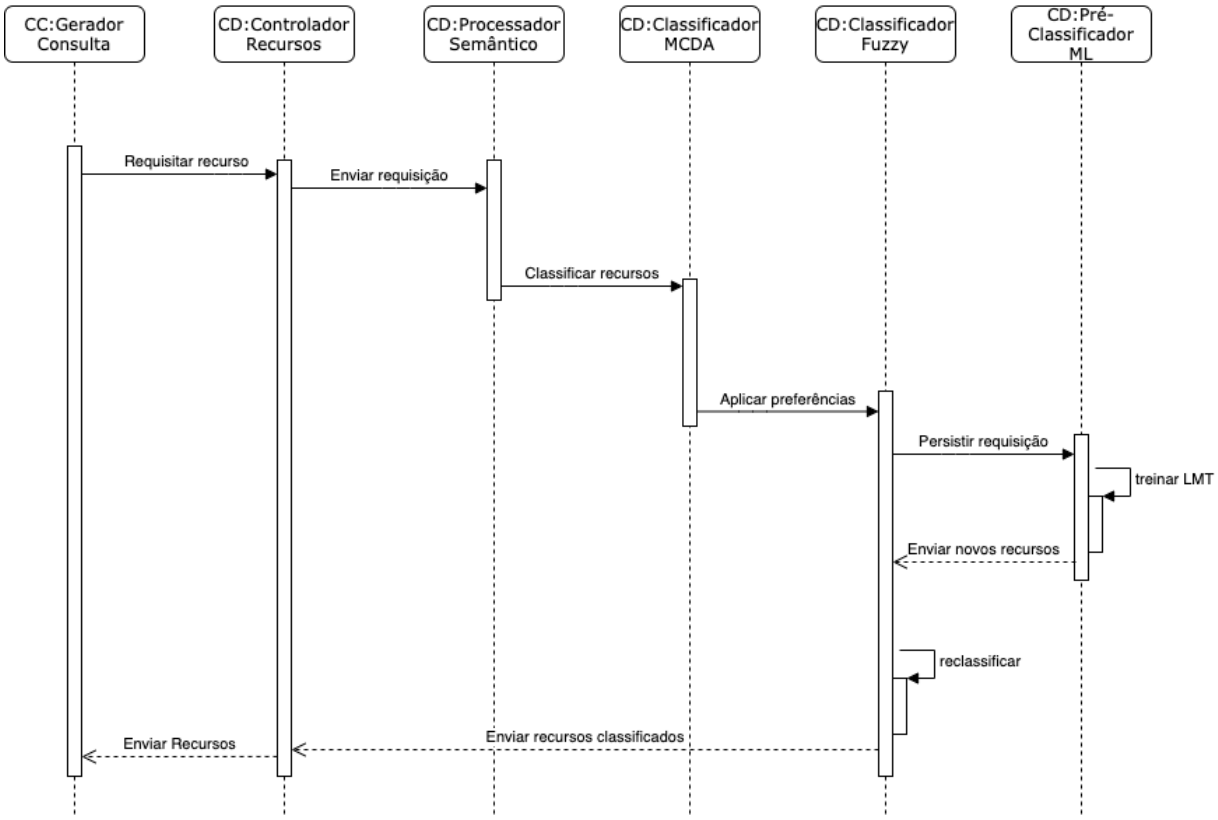

Figura 13 – Fluxo de Execução do Módulo Classificador de Recursos

Considerando que o Módulo Classificador de Recursos do EXEHDA-RR agrega as principais contribuições desta Tese, segue, na Seção 4.2 uma discussão circunstanciada sobre esse módulo. Nesta seção estão sistematizadas as suas funcionalidades, os serviços da arquitetura que são envolvidos e a sua sinergia de atuação no atendimento das diferentes demandas.

4.2 Módulo Classificador de Recursos

O módulo “Classificador de Recursos” é responsável pela classificação e seleção de recursos do *middleware* EXEHDA. Este módulo contém os seguintes serviços: Classificador MCDA, Pré-Classificador ML e Classificador Fuzzy.

4.2.1 Serviço Classificador MCDA

O Classificador MCDA tem por objetivo classificar os recursos descobertos, comparando e avaliando os atributos de QoS. Os recursos são classificados com valores de “1” a “4", sendo os melhores recursos são classificados com valor “1” e os piores, com valor “4". Essa reclassificação do algoritmo MCDA tem por finalidade facilitar o treinamento do serviço Pré-Classificador ML, que pré-classifica novos recursos descobertos por meio do *Machine Learning*, evitando, com isso, a necessidade constante de comparar todos os recursos descobertos e armazenados no Componente Diretório.

O algoritmo MCDA utilizado neste trabalho é uma adaptação dos algoritmos SAW e *Web Service Relevancy Function* (WsRF) (AL-MASRI; MAHMOUD, 2007), com a primeira etapa de normalização de dados proposta por (LIU; NGU; ZENG, 2004).

Para a normalização da matriz, são definidos dois vetores. No primeiro, N = $\{n_1,n_2,...,n_m\}$, o valor de n_j poderá ser 0 ou 1. Será 1 quando o aumento de $q_{i,j}$ beneficiar a requisição do cliente e 0 quando o aumento de $q_{i,j}$ não beneficiar essa requisição. O segundo vetor, C = $\{c_1, c_2,...,c_m\}$, contém constantes com o máximo valor normalizado para cada atributo.

Para o cálculo da avaliação dos recursos através do algoritmo MCDA, devem ser realizadas as seguintes etapas:

1. Normalizar a matriz $Q = (q_{ij})_{n\times m}$ de acordo com a Eq.(6), se o critério tiver que ser maximizado, ou Eq.(7), se o critério tiver que ser minimizado. Nessas equações, $\frac{1}{n}\sum_{i=1}^{n} q_{i,j}$ é a média dos atributos de qualidade j na matriz Q.

$$v_{i,j} = \begin{cases} \frac{q_{i,j}}{\frac{1}{n}\sum_{i=1}^{n} q_{i,j}} & if \quad \frac{1}{n}\sum_{i=1}^{n} q_{i,j} \neq 0 \\ & and \quad \frac{q_{i,j}}{\frac{1}{n}\sum_{i=1}^{n} q_{i,j}} < c_j \\ & and \quad n_j = 1 \\ c_j & if \quad \frac{1}{n}\sum_{i=1}^{n} q_{i,j} = 0 \\ & and \quad n_j = 1 \\ & or \quad \frac{q_{i,j}}{\frac{1}{n}\sum_{i=1}^{n} q_{i,j}} \geq c_j \end{cases} \tag{6}$$

$$v_{i,j} = \begin{cases} \frac{\frac{1}{n}\sum_{i=1}^{n} q_{i,j}}{q_{i,j}} & if \quad q_{i,j} \neq 0 \\ & and \quad n_j = 0 \\ & and \quad \frac{\frac{1}{n}\sum_{i=1}^{n} q_{i,j}}{q_{i,j}} < c_j \\ c_j & if \quad q_{i,j} = 0 \\ & and \quad n_j = 0 \\ & or \quad \frac{\frac{1}{n}\sum_{i=1}^{n} q_{i,j}}{q_{i,j}} \geq c_j \end{cases} \tag{7}$$

2. Calcular o vetor pontuação de cada opção disponível. Cada pontuação pode ser calculada usando-se a Eq.(8), com o operador $\max(v_j)$ representando o maior valor de atributo normalizado na coluna j. Portanto, é necessário definir uma matriz que represente a contribuição de pesos para cada recurso, onde $w = \{w1, w2, w3, ..., wj\}$. Cada peso nessa matriz representa o grau de importância ou fator de peso associado a uma propriedade QoS específica. Os valores desses pesos variam de 0 a 1. A Eq.(9) soma todos os atributos de qualidade para o recurso R_i, onde N representa o número de atributos.

$$h_{i,j} = w_j \left[\frac{v_{i,j}}{max(v_j)} \right] \quad (8) \qquad R_i = \sum_{j=1}^{N} h_{i,j} \quad (9) \qquad MCDA(A_i) = \left[\frac{100 * R_i}{max(R)} \right] \quad (10)$$

3. O resultado final do algoritmo MCDA, Eq.(10), é reclassificado, como mostra a Figura 14. As folhas da árvore são representadas por uma caixa contendo a classificação (1 a 4).

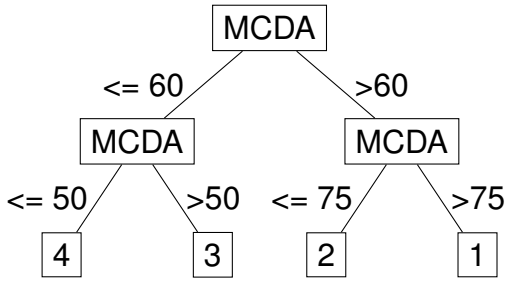

Figura 14 – Reclassificação da Saída Produzida pelo Algoritmo MCDA

O EXEHDA-RR foi modelado para tratar a incerteza na especificação das preferências dos clientes na classificação de recursos IoT. O valor de cada atributo de QoS, bem como a especificação dos recursos disponíveis, ficam armazenados no Componente Diretório de cada célula do ambiente computacional. Os atributos são avaliados e classificados, inicialmente, pelo algoritmo MCDA.

A aplicação do algoritmo MCDA é demonstrada na classificação de 10 recursos, com cinco atributos de qualidade do *dataset* QWS (AL-MASRI; MAHMOUD, 2007).

A tabela 6 apresenta a relação de atributos de qualidade (QoS) selecionados do *dataset* QWS e uma breve descrição sobre eles.

Tabela 6 – Classificador MCDA - Atributos Selecionados do Dataset QWS

Atributo	**Descrição**	**Unid**
RT - Response Time	Tempo do envio de uma requisição e receber sua resposta	ms
AV - Availability	Número de invocações corretas/total invocações	%
TH - Throughput	Total de invocações por um determinado período	%
RE - Reliability	Taxa de erro no total de mensagens	%
LA - Latency	Tempo que o servidor leva para processar uma requisição	ms
ID - Identificador	Identificador Único em substituição do URI	-

Na Tabela 7 a seguir, cada linha representa um recurso e cada coluna, um atributo de qualidade. Os valores dos atributos Response Time e Latency serão considerados melhores se forem baixos e os valores dos atributos Availability, Throughput e Reliability serão considerados melhores se forem altos.

Tabela 7 – Classificador MCDA - Dados Normalizados

	Original					Normalizado				
ID	**RT**	**AV**	**TH**	**RE**	**LA**	**RT**	**AV**	**TH**	**RE**	**LA**
01	302,75	89	7,1	73	187,75	1,70	1,07	0,97	1,06	0,21
02	482	85	16	73	1	1,07	1,02	2,19	1,06	38,56
03	3321,4	89	1,4	73	2,6	0,15	1,07	0,19	1,06	14,83
04	126,17	98	12	67	22,77	4,07	1,18	1,64	0,98	1,69
05	107	87	1,9	73	58,33	4,80	1,04	0,26	1,06	0,66
06	107,57	80	1,7	67	18,21	4,78	0,96	0,23	0,98	2,12
07	255	98	1,3	67	40,8	2,02	1,18	0,18	0,98	0,95
08	136,71	76	2,8	60	11,57	3,76	0,91	0,38	0,87	3,33
09	102,62	91	15,3	67	0,93	5,01	1,09	2,10	0,98	41,46
10	200	40	13,5	67	41,66	2,57	0,48	1,85	0,98	0,93

A primeira etapa da classificação é a normalização dos dados. Para tanto, consideramos os vetores N={0, 1, 1, 1, 0}, C={6, 2, 3, 2, 50} e w={1, 1, 1, 1, 1}. Os atributos Resp.Time e Latency qualificam o recurso com valores baixos, Eq.(4), e Availability, Throughput e Reability qualificam com valores altos, Eq.(3). Todos os atributos são normalizados com valor máximo definido em “C”. No vetor “w”, será informado se o atributo é de interesse ou não, de acordo com a preferência do cliente. Por exemplo, caso o cliente deseje informar as preferências para os atributos Availability, Reliability e Latency, será definido o vetor w={0,1,0,1,1}. A sequência dos atributos é a seguinte w={Response Time, Availability, Throughput, Reliability e Latency}. Se o cliente especificar a preferência, será informado o valor “1” para o referido atributo de qualidade, caso contrário, “0”. O resultado dessa etapa é apresentado na Tabela 7.

A Tabela 8 apresenta os valores dos atributos após aplicar a Eq.(5), ou seja, dividir o valor normalizado da Tabela 7 pelo maior valor normalizado de cada coluna. Após, é aplicada a Eq.(6), que irá somar todos os valores de atributos em cada linha. Na sequência, é aplicada a Eq.(7), que irá qualificar o recurso com um valor que vai de 0 a 100, sendo o valor 100 para o melhor recurso do *dataset*. A classificação de 1 a 4 é atribuída através da regra ilustrada na Figura 14.

O algoritmo MCDA apresentado é utilizado para classificar e selecionar os melhores recursos e gerar a classificação de referência para o Pré-Classificador.

4.2.2 Serviço Pré-Classificador Machine Learning

Na infraestrutura computacional provida pela IoT a entrada e saída de recursos é muito dinâmica e altamente escalável. Consequentemente, uma pré-classificação de novos recursos quando estes entram no ambiente computacional, permite reduzir o custo computacional, evitando de compará-los ao conjunto de recursos existentes por meio do classificador MCDA.

Tabela 8 – Classificador MCDA - Dados Classificados

ID	RT	AV	TH	RE	LA	MCDA	Classificação
01	0,34	0,91	0,44	1,00	0,00	56,13	3
02	0,21	0,87	1,00	1,00	0,93	83,50	1
03	0,03	0,91	0,09	1,00	0,36	49,64	4
04	0,81	1,00	0,75	0,92	0,04	73,33	2
05	0,96	0,89	0,12	1,00	0,02	62,08	2
06	0,95	0,82	0,11	0,92	0,05	59,25	3
07	0,40	1,00	0,08	0,92	0,02	50,48	3
08	0,75	0,78	0,18	0,82	0,08	54,21	3
09	1,00	0,93	0,96	0,92	1,00	100	1
10	0,51	0,41	0,84	0,92	0,02	56,33	3

O Pré-Classificador é ativado quando o cliente deseja manter uma persistência de sua requisição por recursos. O cliente pode especificar em minutos o tempo em que este serviço irá monitorar a entrada de novos recursos, classificando-os e enviando os melhores, classificados como “1” ao Classificador Fuzzy.

Algoritmos de *Machine Learning* supervisionado podem ser utilizados para a geração automática de regras, com o objetivo de classificar os serviços, considerando-se os requisitos não funcionais. Para tanto, é realizada uma análise da acurácia dos principais algoritmos de árvore de decisão. Os algoritmos empregados são utilizados para geração do subconjunto de um *dataset* utilizado para treinamento de um algoritmo de aprendizado, no caso, árvore de decisão.

Algoritmos de árvore de decisão são utilizados para aprender a classificação dos recursos de forma equivalente à da Função de Revelância de Web Service (WsRF). Essa função proposta por (AL-MASRI; MAHMOUD, 2007) tem por finalidade classificar recursos, avaliando os atributos de QoS. A função baseia-se no algoritmo MCDA SAW, com adaptações para preferências do cliente.

Como exemplo de treinamento de algoritmos de árvore de decisão, nesta Tese foi utilizado o *dataset* QWS disponibilizado por (AL-MASRI; MAHMOUD, 2007), com 364 instâncias e 9 atributos de qualidade.

Como parte dos trabalhos desta pesquisa, foram avaliados os algoritmos de árvore de decisão J48, LMT, RandomForest, SimpleCart, DecisionStump, RandomTree, HoeffdingTree e REPTree, na ferramenta WEKA (WITTEN; FRANK; HALL, 2011). O objetivo da avaliação é verificar a acurácia dos algoritmos na classificação de serviços considerando os valores de atributos de QoS.

A classificação dos serviços foi realizada através do aprendizado das regras necessárias para classificar o atributo “*Service Classification*” definido no *dataset* QWS (AL-MASRI; MAHMOUD, 2007).

Utilizou-se o *Machine Learning* na predição de regras, sem a necessidade de realizar uma pré-avaliação dos serviços através de funções ou cálculos. A classificação

dos melhores serviços foi realizada pelos algoritmos de árvore de decisão.

Conforme os resultados apresentados na Tabela 9, nota-se que o classificador LMT obteve uma acurácia de 85,99%.

Tabela 9 – Pré-Classificador ML - Comparação da Acurácia dos Algoritmos

Classificador	Instâncias Corretas	Instâncias Incorretas	Acurácia
LMT	313	51	85,99%
RandomForest	294	70	80,77%
J48	258	106	70,88%
RandomTree	248	116	68,13%
SimpleCart	248	116	68,13%
REPTree	239	125	65,66%
HoeffdingTree	198	166	54,40%
DecisionStump	160	204	43,95%

Os testes dos classificadores foram realizados sem alterar os padrões do WEKA, com exceção do J48. A acurácia do J48, com as configurações padrões, foi de 69,50%. Após ativar a opção "*ReduceErrorPruning*", a acurácia subiu para 70,88%.

Tabela 10 – Pré-Classificador ML - Matriz de Confusão do Algoritmo LMT

a	b	c	d	classificação
37	4	0	0	a = 1
3	86	11	0	b = 2
0	11	100	9	c = 3
0	0	13	90	d = 4

Tabela 11 – Pré-Classificador ML - Matriz de Confusão do Algoritmo J48

a	b	c	d	classificação
31	9	1	0	a = 1
15	66	18	1	b = 2
1	24	79	16	c = 3
0	2	19	82	d = 4

Pode-se observar na Matriz de Confusão do algoritmo LMT, conforme Tabela 10, que os erros ocorridos na predição foram pequenos e aceitáveis na perspectiva da qualidade do serviço na escala de 1 a 4, considerando o valor 1 mais bem qualificado. Na primeira linha, onde todas as instâncias deveriam ser classificadas com valor 1, quatro foram classificados com valor 2. O erro é pequeno, pois nenhuma instância classificada como 1 no *dataset* original foi classificada em 3 ou 4. Na segunda linha, todas as instâncias deveriam ser classificadas com valor 2, mas três foram classificadas com valor 1 e 11 classificadas com valor 3. Na terceira linha, as instâncias deveriam ser classificadas com valor 3, mas 11 foram classificadas com valor 2 e nove, classificadas com valor 4. Na última linha, as instâncias deveriam ser classificadas com valor 4, mas 13 instâncias foram classificadas com valor 3.

A Matriz de Confusão do algoritmo J48, conforme Tabela 11, apresenta uma maior imprecisão, se comparada à matriz de confusão do algoritmo LMT. Na primeira linha, onde todas as instâncias deveriam ser classificadas com valor 1, nove foram classificadas com valor 2 e uma, classificada com valor 3. Na segunda linha, todas as

instâncias deveriam ser classificadas com valor 2, mas 15 foram classificadas com valor 1, 18 classificadas com valor 3 e somente 1 com valor 4. Na terceira linha, as instâncias deveriam ser classificadas com valor 3, mas uma foi classificada com valor 1, 24 com valor 2, 79 com valor 3 e 16 com valor 4. Na última linha, as instâncias deveriam ser classificadas com valor 4, mas duas foram classificadas com valor 2, 19 com valor 3 e 82 com valor 4.

Após a avaliação e escolha do algoritmo para a pré-classificação dos recursos, foi realizada uma análise sobre a quantidade mínima de recursos para o treinamento inicial do algoritmo e os intervalos para novos treinamentos. O Pré-Classificador é iniciado somente com uma quantidade mínima de recursos no Componente Diretório (CD), necessária para o treinamento inicial do algoritmo LMT.

Com o intuito de avaliar a quantidade mínima de recursos para treinamento e o intervalo entre treinamentos do algoritmo de *Machine Learning*, foram realizadas quatro avaliações da pré-classificação de recursos: (E1) grupos de 50 recursos, totalizando 20 grupos; (E2) grupos de 100 recursos, totalizando 10 grupos; (E3) grupos de 200 recursos, totalizando 5 grupos; e (E4) grupos de 500 recursos, totalizando 2 grupos. Em cada avaliação, foram classificados 1000 recursos com valores de 1 a 4. Foi escolhido o algoritmo LMT por ter obtido a melhor acurácia dentre os algoritmos de árvore de decisão analisados.

Em cada avaliação, o grupo de recursos classificados através do algoritmo MCDA foi retirado do conjunto de dados de treinamento e usado no conjunto de dados de teste do algoritmo de *Machine Learning*.

As avaliações foram realizadas com os seguintes repositórios iniciais para treinamento do algoritmo LMT: (a) - repositório inicial com 100 recursos; (b) - repositório inicial com 200 recursos; e (c) - repositório inicial com 300 recursos.

A Tabela 12 apresenta as classificações incorretas dos recursos e o percentual de acertos. A cada grupo avaliado, o algoritmo MCDA reclassifica os recursos e treina o algoritmo de *Machine Learning*.

Ao analisar a acurácia obtida pela pré-classificação do algoritmo de *Machine Learning* LMT, pode-se observar que E1c obteve a acurácia de 84,2%. O repositório inicial estava com 300 recursos, utilizados para treinamento, sendo que, a cada 50 recursos, foi realizado um novo treinamento.

As avaliações E2b e E2c também mostraram-se satisfatórias, com acurácia de 78,4% e 80,5%. Essas avaliações foram realizadas a cada 100 recursos, portanto, os processos de normalização de dados e cálculos do algoritmo MCDA só é realizado a cada 100 recursos.

A avaliação 4 mostrou-se imprópria para a pré-classificação de recursos, com acurácia abaixo de 54% (E4c). O intervalo de 500 recursos para recálculo do algoritmo MCDA e treinamento do *Machine Learning* mostrou-se muito grande, acarretando a

Tabela 12 – Pré-Classificador ML - Avaliação da Precisão do Algoritmo LMT

Qtd	E1a	E1b	E1c	E2a	E2b	E2c	E3a	E3b	E3c	E4a	E4b	E4c
50	14	7	8									
100	13	8	10	26	21	21						
150	8	12	8									
200	11	10	11	20	24	16	41	45	42			
250	11	10	11									
300	14	10	11	22	17	18						
350	7	6	10									
400	11	8	10	13	14	15	35	34	28			
450	5	6	6									
500	6	6	5	14	12	13				109	106	108
550	7	6	14									
600	12	14	8	16	14	27	63	61	77			
650	20	21	6									
700	17	19	9	35	38	16						
750	17	16	5									
800	16	17	6	34	31	12	73	76	24			
850	11	14	11									
900	11	9	19	22	23	45						
950	14	10	7									
1000	7	11	5	26	22	12	46	50	87	213	193	356
TOTAL	232	223	180	228	216	195	258	266	258	322	299	464
Acurácia%	76,8	77,7	82	77,2	78,4	80,5	74,2	73,4	74,2	67,8	70,1	53,6

perda de acurácia.

O Pré-Classificador do EXEHDA-RR utiliza o algoritmo de *Machine Learning* LMT para a pré-classificação de recursos. Para a ativação do serviço, é necessária uma quantidade mínima de 100 recursos, enviados pelo Classificador MCDA, para treinamento do algoritmo. Após o treinamento inicial, a cada 50 novos recursos é realizado novo treinamento.

4.2.3 Serviço Classificador Fuzzy

No processo de classificação de recursos que visa satisfazer a solicitação do usuário, a avaliação dos atributos de QoS é uma etapa desafiadora. A definição do grau de importância dos atributos de QoS pelo cliente é uma atividade que depende da experiência e do conhecimento de cada indivíduo. O tratamento das incertezas ocasionadas na especificação e processamento das preferências do cliente é realizado com o emprego de Lógica Fuzzy Tipo-2 Intervalar, sendo essas preferências traduzidas em regras para o motor de inferência fuzzy, alterando a classificação final dos recursos.

O cliente poderá definir suas preferências para cada um dos atributos de QoS. A relevância para cada um deles pode ser Low, Medium ou High. Conforme a preferência selecionada, serão ativadas regras adicionais na base regras fuzzy.

No EXEHDA-RR, os atributos de QoS dos recursos são normalizados e submetidos como entrada no sistema Fuzzy Tipo-2, fuzzificados por meio de funções de pertinência trapezoidais e triangulares. A utilização de funções de pertinência Fuzzy Tipo-2 na entrada de controladores fuzzy pode resultar na diminuição do número de regras, quando comparados com um sistema fuzzy convencional. O Classificador Fuzzy consiste em seis etapas, as quais são descritas na sequência.

I. Normalização

Os recursos classificados pelo Classificador MCDA ou pelo Pré-Classificador como melhores, ou seja, "1", são normalizados antes de serem processados pelo Classificador Fuzzy. No processo de normalização, o melhor valor de atributo de QoS é normalizado em 100 e isso é feito para cada um dos atributos. Aqueles que já possuem sua escala de valores (0 - 100) não são normalizados, como é o caso do atributo AV - Availability. A normalização é realizada por meio da Eq.(11), se o critério tiver que ser maximizado, ou Eq.(12), se tiver que ser minimizado.

$$h_{i,j} = \left[\frac{v_{i,j} * 100}{max(v_j)}\right] \qquad (11) \qquad\qquad h_{i,j} = \left[\frac{min(v_j) * 100}{v_{i,j}}\right] \qquad (12)$$

II. Fuzzificação

Processo que transforma uma entrada crisp em um valor fuzzy $\tilde{A}$, indicando o grau de pertinência do elemento $x \in U$ no conjunto fuzzy A, frequentemente dado na forma triangular (A $= (a, b, c)$) ou trapezoidal ($\tilde{A} = (a, b, c, d)$), cuja correspondente expressão analítica é dada pela Eq.(13) ou Eq.(14), a seguir:

$$\mu_A(x) = \begin{cases} 0, & se\ x \leq a; \\ \frac{x-a}{b-a}, & se\ a < x \leq b; \\ \frac{c-x}{c-b}, & se\ b \leq x < c; \\ 0, & se\ x \geq c. \end{cases} \qquad (13) \qquad\qquad \mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & se\ a \leq x < b; \\ 1, & se\ b \leq x \leq c; \\ \frac{d-x}{d-c}, & se\ c < x \leq d; \\ 0, & \text{caso contrário.} \end{cases} \qquad (14)$$

Com o objetivo de avaliar o tamanho adequado da mancha de incerteza (FOU), realizou-se a classificação de 13 recursos, sem especificar as preferências do cliente. Os recursos foram classificados com Fuzzy Tipo-1 e Fuzzy Tipo-2 com o tamanho da FOU definida para cada Termo Linguístico da seguinte forma: Low, c=0,3; Medium, c=0,4; e High, c=0,2. Novamente os recursos foram classificados com Fuzzy Tipo-2, com o dobro da FOU: Low, c=0,6; Medium, c=0,8; e High, c=0,4. As classificações foram comparadas ao algoritmo MCDA. Esse experimento foi realizado com o intuito de definir o tamanho da FOU para cada função de pertinência.

Na Tabela 13, são apresentadas as variáveis linguísticas RT (Response Time), AV (Availability) e TH (Throughput). Na primeira coluna estão os termos linguísticos. O Termo High foi modelado com função de pertinência triangular e os Termos Linguísticos Medium e Low, modelados com função de pertinência trapezoidal. Já a variável linguística de saída Weight foi modelada empregando funções de pertinência trapezoidal. Os termos linguísticos e seus valores são apresentados na Tabela 14.

Tabela 13 – Classificador Fuzzy - Variáveis Ling. Tipo-1: RT, AV e TH

Termo Linguístico	FP
H - High	(62,100,100)
M - Medium	(37,52,62,77)
L - Low	(0,0,32,52)

Tabela 14 – Classificador Fuzzy - Variável Linguística Tipo-1: Weight

Termo Linguístico	FP
HighWeight	(52,82,100,100)
MediumWeight	(12,42,57,87)
LowWeight	(0,0,22,47)

Para cada atributo de QoS é adicionada uma entrada no sistema Fuzzy Tipo-2 Intervalar, modelado com uma única saída que representa o peso de importância para o recurso avaliado. Os atributos de QoS são identificados por variáveis linguísticas.

As funções de pertinência Tipo-1 utilizadas estão graficamente representadas na Figura 15. O termo linguístico High (62,100,100) é representado por números fuzzy triangulares. Já os termos linguísticos Medium (37,52,62,77) e Low (0,0,32,52) por números fuzzy trapezoidais.

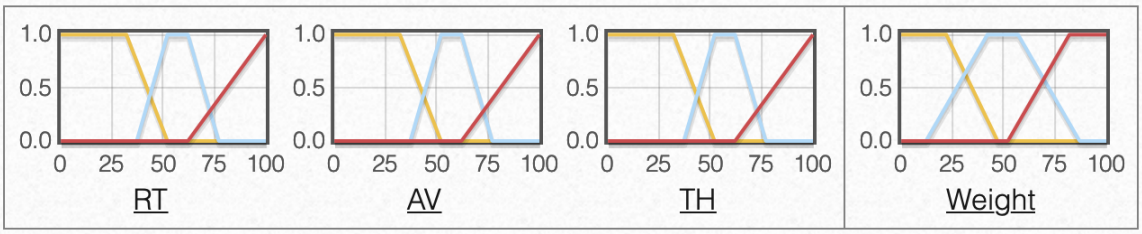

Figura 15 – Classificador Fuzzy - Funções de Pertinência Fuzzy Tipo-1

E, para o universo $\chi = \{u_1, \ldots, u_{100}\}$, tem-se os valores fuzzy $x_i = \mu_A(u_i)$, $\forall i \in \mathbb{N}_{100}$, dos conjuntos fuzzy associados aos termos linguísticos High, Medium e Low dos atributos de entrada RT, AV e TH definidos pelas funções de pertinência dadas pelas equações a seguir:

$$\mu_{HIGH}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 62; \\ \frac{u-62}{38}, & se\ 62 \leq u \leq 100. \end{cases}$$

$$\mu_{MEDIUM}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 37; \\ \frac{u-37}{15}, & se\ 37 \leq u \leq 52; \\ 1, & se\ 52 \leq u \leq 62; \\ \frac{77-u}{15}, & se\ 62 \leq u \leq 77; \\ 0, & se\ 77 \leq u \leq 100. \end{cases}$$

$$\mu_{LOW}(u) = \begin{cases} 1, & se\ 1 \leq u \leq 32; \\ \frac{52-u}{20}, & se\ 32 \leq u \leq 52; \\ 0, & se\ 52 \leq u \leq 100. \end{cases}$$

Na modelagem fuzzy para a variável de saída Weight, também foram considerados três termos linguísticos: low, medium e High. Assim, para o universo $\chi = \{u_1, \ldots, u_{100}\}$, têm-se os valores fuzzy $x_i = \mu_A(u_i)$, $\forall i \in \mathbb{N}_{100}$, do conjuntos fuzzy associado Weight, sendo estes definidos pelas funções de pertinência dadas pelas equações que seguem:

$$\mu_{HIGH}(u) = \begin{cases} 0, & se\, 0 \leq u \leq 52; \\ \frac{u-52}{30}, & se\ 52 \leq u \leq 82; \\ 1, & se\ 82 \leq u \leq 100. \end{cases} \qquad \mu_{MEDIUM}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 12; \\ \frac{u-42}{30}, & se\ 12 \leq u \leq 42; \\ 1, & se\ 42 \leq u \leq 57; \\ \frac{57-u}{30}, & se\ 52 \leq u \leq 87; \\ 0, & se\ 87 \leq u \leq 100. \end{cases}$$

$$\mu_{LOW}(u) = \begin{cases} 1, & se\ 1 \leq u \leq 22; \\ \frac{22-u}{25}, & se\ 22 \leq u \leq 47; \\ 0, & se\ 47 \leq u \leq 100. \end{cases}$$

O tamanho da FOU foi definido utilizando o parâmetro *c* ∈ [0,1] definido em (ALADI; WAGNER; GARIBALDI, 2014). Sendo que *c* = 0 representa um conjunto Fuzzy Tipo-1 com a função de pertinência original, enquanto *c* = 1 resulta em uma FOU muito larga. As funções de pertinência Fuzzy Tipo-2 foram especificadas tendo por base o parâmetro de incerteza “c”. A partir das funções de pertinência tipo-1 foram utilizadas as equações Eq.15 e Eq.16 para definir os limites inferiores e superiores.

$$\overline{\mu}_{\tilde{A}}(x) = min\left[\mu_A(x) + \frac{c}{2}, 1.0\right] \tag{15}$$

$$\underline{\mu}_{\tilde{A}}(x) = min\left[max\left[\mu_A(x) - \frac{c}{2}, 0\right], 1.0 - c\right] \tag{16}$$

As funções de pertinência tipo-2 com mancha de incerteza menor estão representadas na Figura 16.

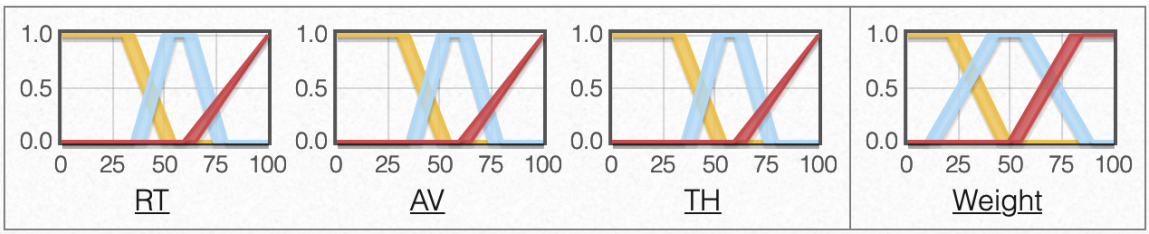

Figura 16 – Classificador Fuzzy - Funções de Pertinência Fuzzy Tipo-2 Small

As funções de pertinência tipo-2 com mancha de incerteza maior estão representadas na Figura 17.

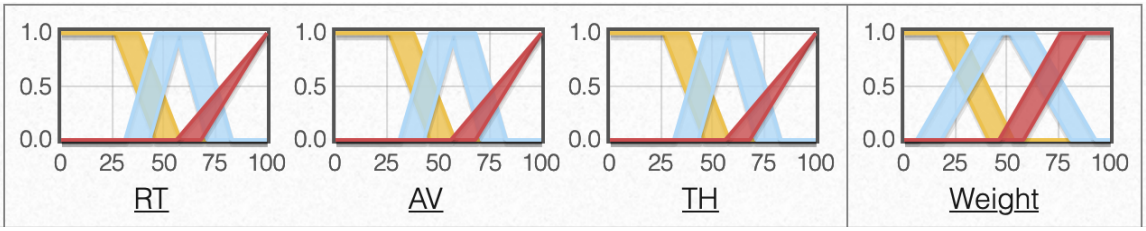

Figura 17 – Classificador Fuzzy - Funções de Pertinência Fuzzy Tipo-2 Large

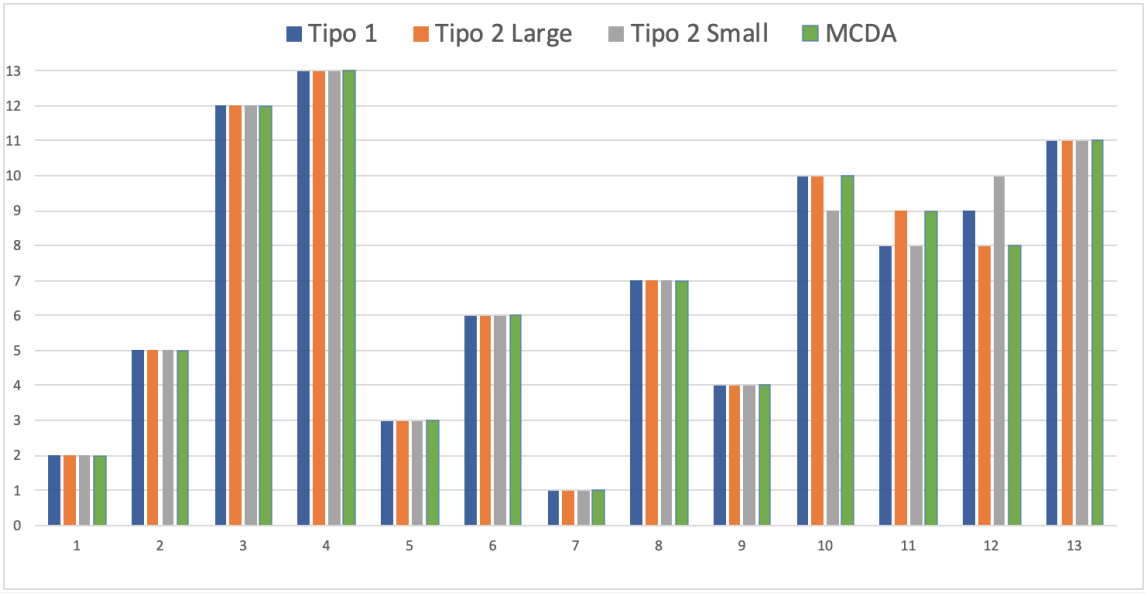

Figura 18 – Comparação dos Resultados das Funções de Pertinência vs MCDA

Pode-se observar na Figura 18 que a classificação com funções de pertinência tipo-2 maiores (Large) obteve 100% de acurácia comparada à classificação com algoritmo MCDA. Desta forma, as funções de pertinência do EXEHDA-RR foram definidas com o tamanho da FOU de acordo com as funções Tipo-2 Large.

Na Tabela 15 são apresentadas as variáveis linguísticas RT, AV e TH. Na primeira coluna estão os termos linguísticos. O Termo High foi modelado com função de pertinência triangular e os Termos Linguísticos Medium e Low foram modelados com função de pertinência trapezoidal. Os limites inferiores e superiores são apresentados nas colunas FP Superior e FP Inferior.

Tabela 15 – Classificador Fuzzy - Variáveis Linguísticas Tipo-2: RT, AV e TH

Termo Linguístico	FP Superior	FP Inferior
H - High	(56,100,100)	(68,100,100)
M - Medium	(31,46,68,83)	(43,58,56,71)
L - Low	(0,0,38,58)	(0,0,26,46)

Para os conjuntos triangulares fuzzy High_Superior ($\overline{H}$) e High_Inferior ($\underline{H}$), temos:

$$\mu_{\overline{H}}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 56; \\ \frac{u-56}{44}, & se\ 56 \leq u \leq 100. \end{cases} \qquad \mu_{\underline{H}}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 68; \\ \frac{u-68}{32}, & se\ 68 \leq u \leq 100. \end{cases}$$

Para os conjuntos trapezoidais fuzzy Medium_Superior ($\overline{M}$) e Medium_Inferior ($\underline{M}$), tem-se:

$$\mu_{\overline{M}}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 31; \\ \frac{u-31}{15}, & se\ 31 \leq u \leq 46; \\ 1, & se\ 46 \leq u \leq 68; \\ \frac{83-u}{15}, & se\ 68 \leq u \leq 83; \\ 0, & se\ 83 \leq u \leq 100. \end{cases} \qquad \mu_{\underline{M}}(u) = \begin{cases} 0, & se\ 0 \leq u \leq 43; \\ \frac{u-43}{15}, & se\ 43 \leq u \leq 58; \\ 1, & se\ 58 \leq u \leq 59; \\ \frac{71-u}{12}, & se\ 59 \leq u \leq 71; \\ 0, & se\ 71 \leq u \leq 100. \end{cases}$$

Para os conjuntos trapezoidais fuzzy Low_Superior ($\overline{L}$) e Low_Inferior ($\underline{L}$), tem-se:

$$\mu_{\overline{L}}(u) = \begin{cases} 1, & se\ 0 \leq u \leq 38; \\ \frac{58-u}{20}, & se\ 38 \leq u \leq 58; \\ 0, & se\ 58 \leq u \leq 100. \end{cases} \qquad \mu_{\underline{L}}(u) = \begin{cases} 1, & se\ 0 \leq u \leq 26; \\ \frac{46-u}{20}, & se\ 26 \leq u \leq 46; \\ 0, & se\ 46 \leq u \leq 100. \end{cases}$$

A variável linguística de saída Weight foi modelada empregando-se funções de pertinência trapezoidal. Os termos linguísticos e seus valores estão apresentados na Tabela 16.

Tabela 16 – Classificador Fuzzy - Variável Linguística Tipo-2: Weight

Termo Linguístico	FP Superior	FP Inferior
H - High	(46,76,100,100)	(58,88,100,100)
M - Medium	(6,36,63,93)	(18,48,51,81)
L - Low	(0,0,28,53)	(0,0,16,41)

III. Base de Regras do Classificador Fuzzy

A Base de Regras descreve como as variáveis linguísticas são processadas em termos linguísticos na Máquina de Inferência. No controlador desenvolvido por Mamdani (MAMDANI; ASSILIAN, 1975) foi adotado um método para o processo de decisão baseado em regras do tipo “If X Then Y”, nas quais tanto o antecedente quanto o consequente são valores de variáveis linguísticas, expressos por meio de conjuntos fuzzy.

Os principais operadores Fuzzy utilizados são a união fuzzy e a intersecção fuzzy. O método Mamdani agrega as regras através do operador OU (união), modelado pelo operador máximo e, em cada regra, o operador lógico E é modelado pelo operador mínimo (intersecção).

No sistema Fuzzy Tipo-2 as regras seguem as mesmas do sistema Fuzzy Tipo-1, porém, a diferença está na natureza das funções de pertinência (KARNIK; MENDEL, 2001).

O EXEHDA-RR processa as preferências do cliente por meio de regras fuzzy. A Tabela 17 contém as 27 regras utilizadas pela máquina de inferência fuzzy.

Tabela 17 – Classificador Fuzzy - Base de Regras Mamdani

Regra	Antecedente	Consequente
01	LowRT and LowAV and LowTH	LowWeight
02	LowRT and LowAV and MediumTH	LowWeight
03	LowRT and LowAV and HighTH	LowWeight
04	LowRT and MediumAV and LowTH	LowWeight
05	LowRT and MediumAV and MediumTH	LowWeight
06	LowRT and MediumAV and HighTH	LowWeight
07	LowRT and HighAV and LowTH	LowWeight
08	LowRT and HighAV and MediumTH	LowWeight
09	LowRT and HighAV and HighTH	LowWeight
10	MediumRT and LowAV and LowTH	LowWeight
11	MediumRT and LowAV and MediumTH	LowWeight
12	MediumRT and LowAV and HighTH	LowWeight
13	MediumRT and MediumAV and LowTH	LowWeight
14	MediumRT and MediumAV and MediumTH	MediumWeight
15	MediumRT and MediumAV and HighTH	MediumWeight
16	MediumRT and HighAV and LowTH	LowWeight
17	MediumRT and HighAV and MediumTH	MediumWeight
18	MediumRT and HighAV and HighTH	MediumWeight
19	HighRT and LowAV and LowTH	LowWeight
20	HighRT and LowAV and MediumTH	LowWeight
21	HighRT and LowAV and HighTH	LowWeight
22	HighRT and MediumAV and LowTH	LowWeight
23	HighRT and MediumAV and MediumTH	MediumWeight
24	HighRT and MediumAV and HighTH	MediumWeight
25	HighRT and HighAV and LowTH	LowWeight
26	HighRT and HighAV and MediumTH	MediumWeight
27	HighRT and HighAV and HighTH	HighWeight

A preferência do cliente é informada para três atributos de QoS, sendo que o sistema possibilita a utilização de mais atributos, mas a base de regras deve ser aumentada para contemplar todas as condições. Os atributos de QoS são representados por variáveis linguísticas na entrada do sistema fuzzy, com três termos linguísticos cada, resultando em 27 regras. Tais regras são sempre processadas, mesmo que o cliente não deseje especificar as preferências, ou seja, todos os atributos de qualidade são importantes. Os atributos RT - Response Time, AV - Availability e TH - Throughput terão o mesmo peso de importância na classificação final.

A lógica utilizada na construção das regras é que o menor valor dos termos linguísticos entre as variáveis linguísticas de entrada é utilizado na variável de saída. Nessa modelagem, por exemplo, a regra 08 pode ser desmembrada da seguinte forma: If RT = Low and AV = High and TH = Medium then Weight = Low. A variável de entrada RT possui o menor termo linguístico (Low), este termo é o mesmo utilizado na variável de saída (Weight).

A Tabela 18 contém regras adicionais para aplicar a preferência do cliente, onde é especificada da seguinte forma: RT - Low, AV - High e TH - Medium. Ao conjunto de 27 regras-base, foram adicionadas duas novas regras:

- (i) regra 28, que verifica se o atributo AV do recurso é High e se o atributo TH é Medium e, caso seja, é atribuído o valor Medium à variável de saída Weight;
- (ii) regra 29, que verifica se o atributo AV é High e se o atributo TH é High e, caso seja, é atribuído o valor High à variável de saída Weight.

Essas regras têm por objetivo levar a pertinência fuzzy das varáveis de entrada à variável de saída, garantindo, assim, a preferência definida pelo cliente.

Tabela 18 – Classificador Fuzzy - Preferência de Cliente LHM

Regra	Antecedente	Consequente
28	HighAV and MediumTH	MediumWeight
29	HighAV and HighTH	HighWeight

A Tabela 19 contém as regras adicionais para aplicar a preferência do cliente, especificada da seguinte forma: RT - Low, AV - High e TH - Low. Ao conjunto de 27 regras-base, foi adicionada uma nova regra: regra 28, que verifica se o valor do atributo AV do recurso é High e, caso seja, é atribuído o valor High à variável de saída Weight.

Tabela 19 – Classificador Fuzzy - Preferência de Cliente LHL

Regra	Antecedente	Consequente
28	HighAV	HighWeight
29	MediumRT	MediumWeight
30	MediumTH	MediumWeight

A Tabela 20 contém as regras adicionais para aplicar a preferência do cliente, onde é especificada da seguinte forma: RT - Medium, AV - High e TH - Medium. Ao conjunto de 27 regras-base, foram adicionadas duas novas regras:

- (i) regra 28, que verifica se o valor do atributo RT do recurso é Medium e, caso seja, é atribuído o valor Medium à variável de saída Weight;

- (ii) regra 29, que verifica se o valor do atributo AV é High e, caso seja, é atribuído o valor High à variável de saída Weight.

Tabela 20 – Classificador Fuzzy - Preferência de Cliente MHM

Regra	Antecedente	Consequente
28	MediumRT and MediumTH	MediumWeight
29	HighAV	HighWeight

A Tabela 21 contém as regras adicionais para aplicar a preferência do cliente, especificada da seguinte forma: RT - High, AV - Low e TH - Low. Ao conjunto de 27 regras-base, foram adicionadas as novas regras:

- (i) regra 28, que verifica se o valor do atributo RT do recurso é High e, caso seja, é atribuído o valor High à variável de saída Weight;
- (ii) regra 29, que verifica se o valor do atributo AV do recurso é Medium e, caso seja, é atribuído o valor Medium à variável de saída Weight;
- (iii) regra 30, que verifica se o valor do atributo TH do recurso é Medium e, caso seja, é atribuído o valor Medium à variável de saída Weight.

Tabela 21 – Classificador Fuzzy - Preferência de Cliente HLL

Regra	Antecedente	Consequente
28	HighRT	HighWeight
29	MediumAV	MediumWeight
30	MediumTH	MediumWeight

A Tabela 22 contém a regra adicional para aplicar a preferência do cliente, especificada da seguinte forma: RT - High, AV - High e TH - Low. Ao conjunto de 27 regras-base, foi adicionada uma nova regra: a regra 28, que verifica se o atributo RT do recurso é High e o atributo AV é High e, caso a condição seja verdadeira, é atribuída a menor pertinência entre as duas variáveis de entrada à variável de saída Weight.

Tabela 22 – Classificador Fuzzy - Preferência de Cliente HHL

Regra	Antecedente	Consequente
28	HighRT and HighAV	HighWeight

A Tabela 23 contém as regras adicionais para aplicar a preferência do cliente, especificada da seguinte forma: RT - Medium, AV - Low e TH - High. Ao conjunto de 27 regras-base, foram adicionadas duas novas regras:

- (i) regra 28, que verifica se o atributo RT do recurso é Medium e o atributo TH é High e, caso a condição seja verdadeira, é atribuída a menor pertinência entre as duas variáveis de entrada à variável de saída Weight;
- (ii) regra 29, que verifica se o atributo TH é High e, caso seja, é atribuído High à variável de saída Weight.

Tabela 23 – Classificador Fuzzy - Preferência de Cliente MLH

Regra	Antecedente	Consequente
28	MediumRT and HighTH	MediumWeight
29	HighTH	HighWeight

A Tabela 24 contém as regras adicionais para aplicar a preferência do cliente, especificada da seguinte forma: RT - High, AV - Low e TH - Medium. Ao conjunto de 27 regras-base, foram adicionadas duas novas regras:

- (i) regra 28, verifica se o atributo RT do recurso é High, caso seja é atribuído High à variável de saída Weight;
- (ii) regra 29 verifica se o atributo TH é Medium, caso seja é atribuído Medium à variável de saída Weight.

Tabela 24 – Classificador Fuzzy - Preferência de Cliente HLM

Regra	Antecedente	Consequente
28	HighRT	HighWeight
29	MediumTH	MediumWeight

A Tabela 25 contém as regras adicionais para aplicar a preferência do cliente, especificada da seguinte forma: RT - High, AV - Medium e TH - Low. Ao conjunto de 27 regras-base, foram adicionadas duas novas regras:

- regra 28, verifica se o atributo RT do recurso é High, caso seja é atribuído High à variável de saída Weight;
- regra 29 verifica se o atributo TH é Medium, caso seja é atribuído Medium à variável de saída Weight.

Novas regras podem ser adicionadas ao EXEHDA-RR através de perfis. As regras definidas fazem parte do perfil "*default*".

IV. Motor de Inferência Fuzzy

Tabela 25 – Classificador Fuzzy - Preferência de Cliente HML

Regra	Antecedente	Consequente
28	HighRT and MediumAV	MediumWeight
29	HighRT and HighAV	HighWeight

Fornece a saída de cada entrada, de acordo com a base de regras construídas por operadores de composição e conectivos LF. A máquina de inferência fuzzy processa as regras do tipo Mamdani, com operador lógico “and”. Dessa forma, na variável linguística de saída “Weight” será atribuída a menor pertinência entre as variáveis linguísticas de entrada (RT, AV e TH).

Com o intuito de exemplificar o processamento das regras, utilizou-se o *framework* JuzzyOnline[1], inserindo-se as 27 regras Mamdani e as funções de pertinência para as variáveis linguísticas RT, AV, TH e Weight (Figuras 19 e 20).

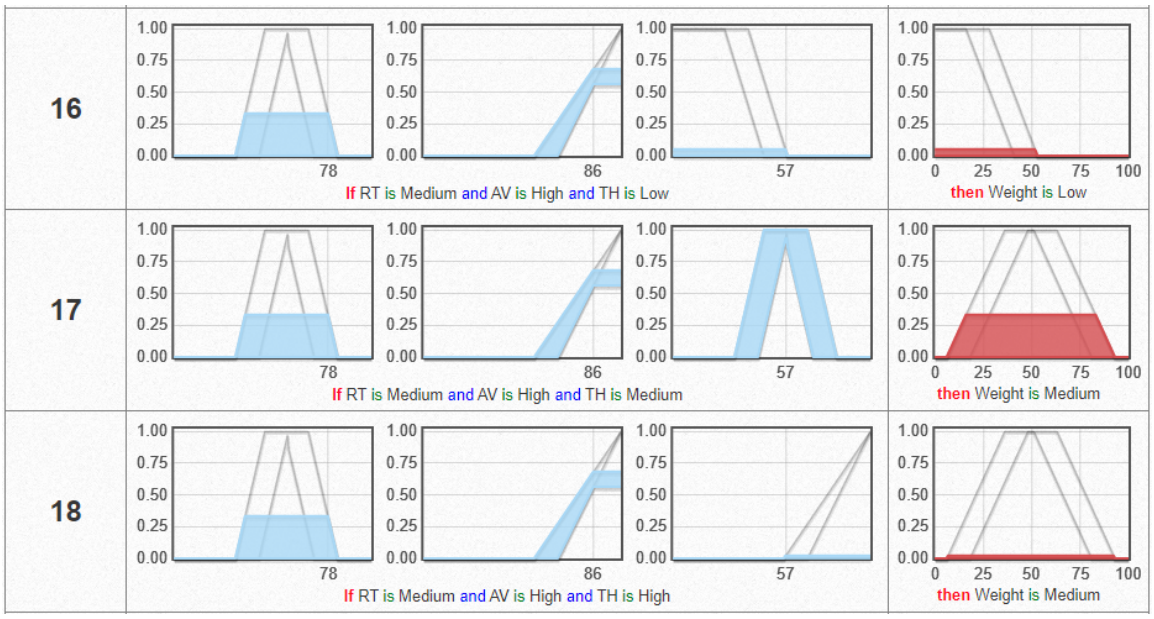

Figura 19 – Inferência Fuzzy - Processamento das Regras 16 a 18

Foram processados os seguintes atributos de QoS, já normalizados, de um recurso qualquer: RT (78), AV(86) e TH(57). Não foram consideradas as preferências do cliente neste exemplo, ou seja, a mesma preferência para cada um dos atributos (HHH, MMM ou LLL). Assim, somente as 27 regras principais foram consideradas. Os atributos deste recurso foram processados por seis regras. A agregação das regras é apresentada na Figura 21.

V. Redutor para Fuzzy Tipo-1 e Defuzzificação

[1] http://ritweb.cloudapp.net:8080/JuzzyOnline2

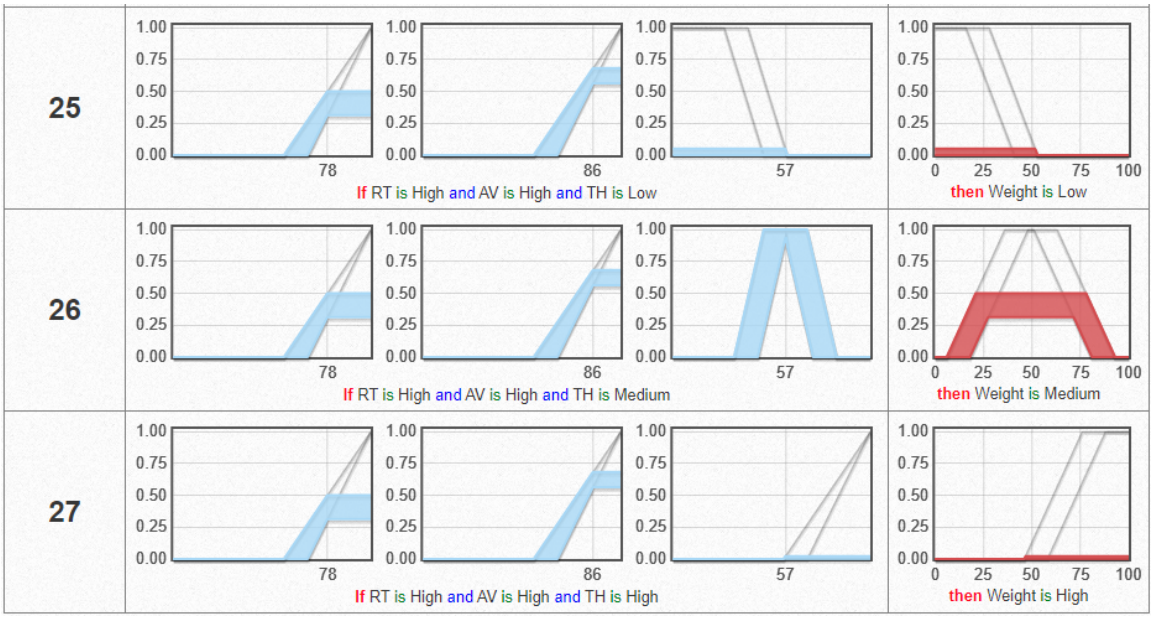

Figura 20 – Inferência Fuzzy - Processamento das Regras 25 a 27

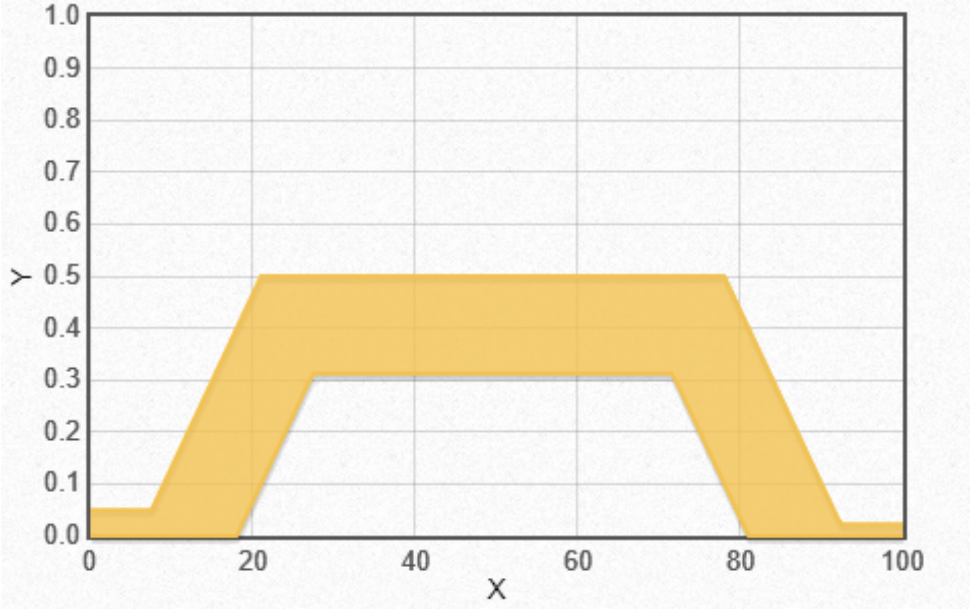

Figura 21 – Inferência Fuzzy - Resultado da Agregação de Regras

Considerando a melhor escolha (de granulosidade), para transformação de um conjunto Fuzzy Tipo-2 em um conjunto Fuzzy Tipo-1. O bloco redutor de tipo tem como objetivo utilizar o algoritmo de Karnik-Mendel (KM) que determina o mínimo (C_L) e o máximo (C_R) dos centroides de conjuntos Fuzzy Tipo-1.

$$C_L = \min_{L \in \mathbb{N}} \mathbf{c}(T(L)); C_R = \max_{R \in \mathbb{N}} \mathbf{c}(T(R)),$$

quando $x_1 < x_2 < \cdots < x_N$ and $L, R \in \mathbb{N}$ são pontos de troca de $\overline{\mu}_{\tilde{A}}$ para $\underline{\mu}_{\tilde{A}}$, e

analogamente, de $\underline{\mu}_{\tilde{A}}$ para $\overline{\mu}_{\tilde{A}}$, respectivamente:

$$\mathbf{c}(T(L)) = \frac{\sum_{i=1}^{L} x_i \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{L} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)}; \quad (17)$$

$$\mathbf{c}(T(R)) = \frac{\sum_{i=1}^{R} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} x_i \overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} \overline{\mu}_{\tilde{A}}(x_i)}. \quad (18)$$

e o máximo (y_R) dos centroides de conjuntos fuzzy do tipo-1.

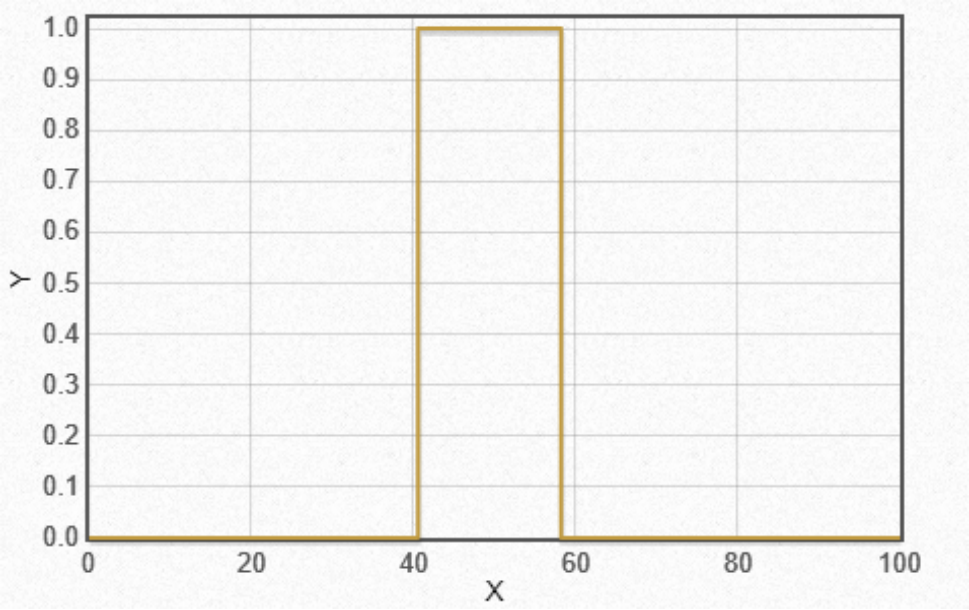

Figura 22 – Inferência Fuzzy - Intervalo Resultante

Observou-se que ao defuzzificar os intervalos para obter um valor crisp com o propósito de ordenar para gerar uma classificação final, em alguns casos o valor defuzzificado de intervalos diferentes permaneceu o mesmo, dificultando o processo de classificação. Neste trabalho não está sendo realizada a defuzzificação, como a média entre dois valores, onde o primeiro calcula o ponto final esquerdo do intervalo (C_L) e o último calcula o ponto final direito (C_R). Optou-se por preservar os intervalos e ordená-los por meio de Ordens Admissíveis.

VI. Ordens Admissíveis

Concluindo a etapa de defuzzificação, como muitos elementos em uma lista de intervalos gerados na abordagem de conjuntos Fuzzy Tipo-2 Intervalar podem não ser comparáveis pela ordem parcial usual, uma ordem total de intervalos é considerada com base nas ordens admissíveis, conforme proposto em (XU; YAGER, 2006).

Assim, nesta proposta, o desenvolvimento do módulo Classificador de Recursos não se restringiu ao uso de operadores de agregação (como média aritmética) execu-

tados sobre os limites superior e inferior do intervalo no procedimento final da etapa de defuzzificação.

Essa nova estratégia de preservação de dados de intervalo nos permite aplicar métricas (por exemplo, diâmetro), estendendo a análise dos resultados do intervalo, considerando não apenas a incerteza associada aos dados de entrada, mas, também, a imprecisão relacionada aos cálculos do processo de classificação. Essa proposta modela as incertezas relacionadas a possível indecisão das preferências do cliente e à imprecisão dos cálculos computacionais. Além disso, continuamos capazes de garantir a comparação de todos os dados do intervalo de saída.

Os recursos são classificados em ordem decrescente de intervalos através da ordem admissível proposta por (XU; YAGER, 2006).

4.3 Considerações Sobre o Capítulo

Diferentemente dos Trabalhos Relacionados, que não fazem foco na natureza dinâmica da IoT, e o decorrente impacto que isso causa, considerando a típica escalabilidade elevada de recursos, o EXEHDA-RR emprega uma técnica de pré-classificação para tratar esta situação.

Nesse sentido, o EXEHDA-RR explora a classificação inicial dos recursos com a proposição de uma abordagem baseada em MCDA. Caso seja de interesse do cliente também monitorar dinamicamente a entrada de novos recursos, foi concebido um Pré-Classificador *Machine Learning*, ativado por demanda. A exploração sinérgica destas duas funcionalidades do EXEHDA-RR viabiliza uma operação com menores custos computacionais.

Por sua vez, o tratamento da incerteza no EXEHDA-RR é realizado por um Classificador Fuzzy. Para isso, o cliente informa suas preferências especificando a relevância dos atributos de QoS dos diferentes recursos IoT empregando variáveis linguísticas. No EXEHDA-RR também é facultado ao especialista poder especificar novas funções de pertinência, para que sejam mais adequadas ao perfil operacional da sua aplicação.

No Capítulo seguinte, são apresentados os três cenários concebidos para avaliar as funcionalidades dos serviços Classificador MCDA, Pré-Classificador *Machine Learning* e Classificador Fuzzy, que integram o Módulo Classificador de Recursos, o qual constitui a contribuição central da Proposta EXEHDA-RR.

5 EXEHDA-RR: CENÁRIOS DE USO

Tendo por base os diversos métodos de avaliação discutidos na literatura da área de ciência da computação, nesta Tese optou-se pelo emprego de uma avaliação baseada em cenários. Destacam-se como principais razões para esta escolha os seguintes critérios: (i) ser considerada uma alternativa madura para ser empregada em ambientes como os da IoT (PATIDAR; SUMAN, 2015); (ii) ser explorada para avaliar a utilização de uma arquitetura de software em relação às demandas de um conjunto de aplicações (SHANMUGAPRIYA; SURESH, 2012); e (iii) ser utilizada como método de avaliação pelos trabalhos relacionados à arquitetura do EXEHDA-RR.

Para avaliar o Classificador de Recursos do EXEHDA-RR foram definidos três cenários de uso que exploram diferentes aspectos da arquitetura de software proposta.

O primeiro cenário tem por finalidade comparar a classificação obtida pelo Classificador Fuzzy, com aquela obtida pelo algoritmo MCDA WsRF. Neste cenário, é facultado ao especialista realizar o ajuste da função de pertinência de um dos atributos de QoS. Oportuno registrar que o algoritmo MCDA WSRF é entendido pela literatura da área como um algoritmo de referência, em termos de classificação de recursos.

Os outros dois cenários exploram solicitações de classificação por recursos, com diferentes preferências definidas pelo cliente, especificadas com o emprego de variáveis linguísticas, minimizando a incerteza e, também, alterando o peso dos atributos de QoS dos recursos, no momento da classificação.

Os recursos IoT foram emulados com a inserção das informações do *dataset* QWS versão 2.0, disponibilizado por (AL-MASRI; MAHMOUD, 2007), com 2.505 recursos. Este *dataset* contém 13 atributos, dos quais foram selecionados três deles para serem inseridos no Componente Diretório. Nestes dois cenários, foram realizadas oito requisições pelo recurso denominado “AnalysisWSAppLabImplService”.

O Classificador MCDA e o Pré-Classificador ML foram desenvolvidos na linguagem Python. Para prototipação dos algoritmos de árvores de decisão, foi utilizado o *Massive Online Analysis* (MOA), um *framework* para mineração de dados. As bibliotecas do MOA e do WEKA foram desenvolvidas em Java e, portanto, foi utilizado o Jpype para que o Python pudesse manipular as bibliotecas em Java. O Classificador Fuzzy

foi desenvolvido em linguagem Java, com o emprego do *framework* Juzzy, proposto por (WAGNER, 2013).

5.1 Cenário 1 - Classificação de Recursos EXEHDA-RR vs WsRF

Este cenário avalia as funcionalidades do Classificador Fuzzy quanto ao tratamento da incerteza na especificação das preferências do cliente, por meio de Variáveis Linguísticas e, também, a possibilidade do especialista adequar as funções de pertinência para que os Termos Linguísticos Low, Medium e High se aproximem da necessidade do cliente. Dessa forma, realizou-se uma avaliação, classificando-se 10 recursos (Tabela 26), com a preferência especificada pelo cliente da seguinte forma: RT - High, AV - Medium e TH - Low.

Tabela 26 – Cenário 1 - Recursos Considerados para Classificação

	Original			Normalizado		
ID	**RT**	**AV**	**TH**	**RT**	**AV**	**TH**
01	220,88	93	11,1	99,34	93	90,24
02	240,03	95	7,6	91,42	95	61,79
03	260,84	80	5,2	84,12	80	42,28
04	254,97	70	4,9	86,06	70	39,84
05	219,43	72	12,3	100	72	100
06	244,11	85	7,2	89,89	85	58,54
07	223,18	50	10,9	98,32	50	88,62
08	251,31	86	6,8	87,31	86	55,28
09	248,21	85	9,7	88,40	85	78,86
10	281,13	65	6,4	78,05	65	52,03

A Tabela 27 apresenta o resultado da classificação dos 10 recursos empregando o algoritmo MCDA WsRF. A preferência RT - High, AV - Medium e TH - Low foi adaptada ao algoritmo MCDA WsRF, com os seguintes pesos de atributo equivalentes: RT - 1, AV - 0,6 e TH - 0,3.

Tabela 27 – Cenário 1 - Classificação da Preferência HML com WsRF

ID	**RT**	**AV**	**TH**	**RT**	**AV**	**TH**	**WsRF**	**Classif.**
01	1,11	1,19	1,35	0,99	0,59	0,27	1,85	1
02	1,02	1,22	0,93	0,91	0,60	0,19	1,70	3
03	0,94	1,02	0,63	0,84	0,51	0,13	1,47	8
04	0,96	0,90	0,60	0,86	0,44	0,12	1,42	9
05	1,11	0,92	1,50	1,00	0,45	0,30	1,75	2
06	1,00	1,09	0,88	0,90	0,54	0,18	1,61	5
07	1,10	0,64	1,33	0,98	0,32	0,27	1,56	7
08	0,97	1,10	0,83	0,87	0,54	0,17	1,58	6
09	0,98	1,09	1,18	0,88	0,54	0,24	1,66	4
10	0,87	0,83	0,78	0,78	0,41	0,16	1,35	10

Nesse cenário, foram utilizadas funções de pertinência para os atributos de QoS RT e TH representadas graficamente na Figura 23.

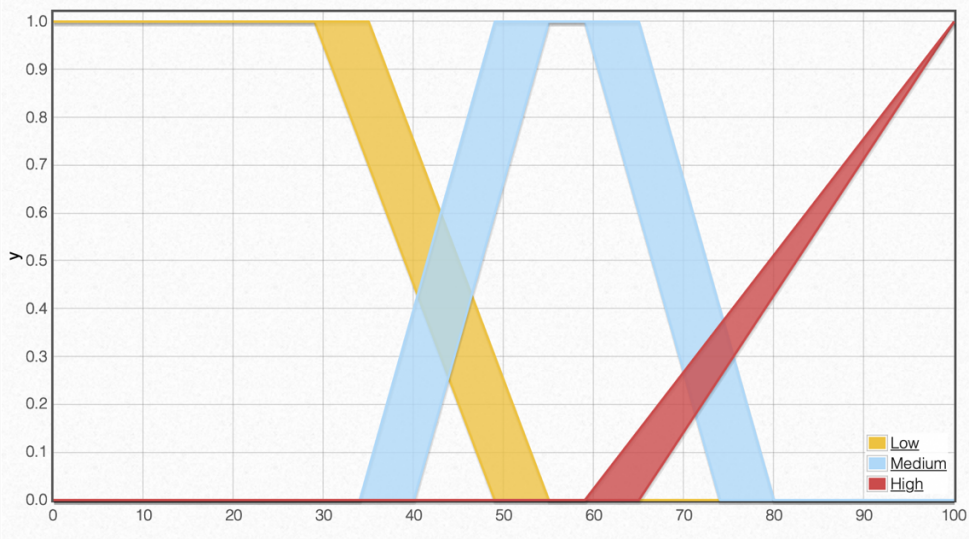

Figura 23 – Cenário 1 - Funções de Pertinência para RT e TH

A função de pertinência AV foi remodelada para melhor atender à determinada aplicação, conforme Figura 24. Neste cenário, foi criado o perfil “cenário_1”, em que o especialista modelou a função de pertinência para o atributo de QoS AV, adequando o termo linguístico Medium com pertinência máxima em 80 e o termo High com função de pertinência iniciando em 80. Esse procedimento, adaptado a um perfil operacional, mantém as preferências especificadas pelo cliente, refletindo na classificação final.

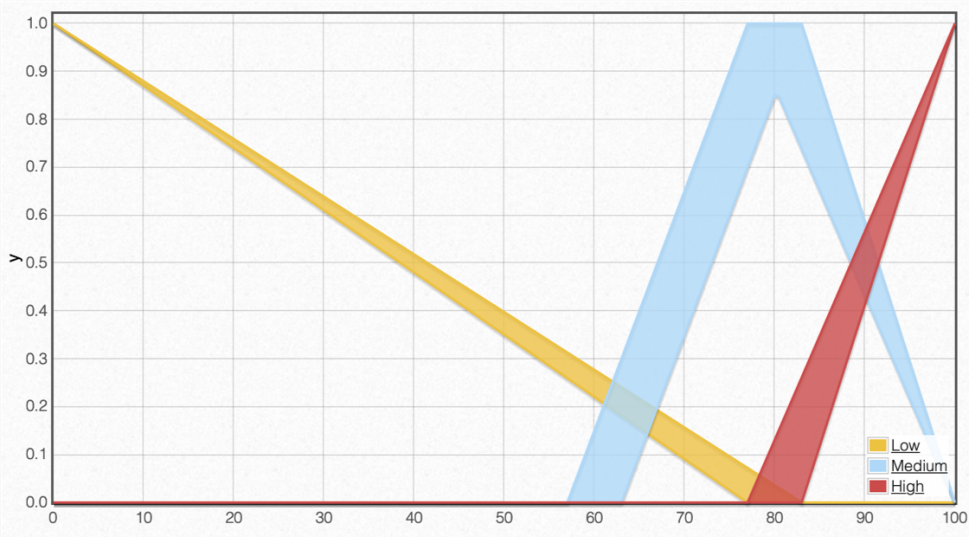

Figura 24 – Cenário 1 - Funções de Pertinência para AV Definidas pelo Especialista

A Figura 25 apresenta a interface para especificação de preferências do EXEHDA-RR. O tratamento da incerteza proveniente dos termos linguísticos Low, Medium e High está relacionada à correta especificação dos conjuntos fuzzy pelos especialistas do domínio.

Nessa interface, o cliente pode especificar os termos linguísticos High, Medium ou Low para cada um dos atributos de QoS, identificados pelas variáveis linguísticas RT,

Figura 25 – Cenário 1 - Seleção da Preferência HML

AV e TH. O Perfil seleciona o conjunto de funções de pertinência especificados pelo especialista, sendo que existe um perfil padrão com as funções de pertinência especificadas na Seção 4.2.3, chamado *default*. A preferência especificada pelo cliente foi HML (RT - High, AV - Medium e TH - Low).

A interface de cadastro de perfil do EXEHDA-RR (Figura 26) é manipulada pelo usuário que possui permissão de administrador. Um perfil representa um conjunto de funções de pertinência e regras do tipo Mandani.

Figura 26 – Cenário 1 - Cadastro de Perfil do Processamento Fuzzy

Após o cadastro do perfil, ele fica disponível para ser utilizado pelo cliente. A Figura

25 apresenta a seleção da preferência HML pelo cliente e o perfil “Cenário_1”.

Por sua vez, a Figura 27 apresenta o resultado da classificação para a preferência de cliente HML, ordenado por ID do recurso. Os valores dos atributos RT, AV e TH estão normalizados. Centróide Inferior e Superior representa o intervalo resultante da defuzzificação. A última coluna apresenta a classificação, sendo o valor 1 para o melhor recurso.

ID	Response Time(RT)	Availability (AV)	Throughput (TH)	Centróide Inferior	Centróide Superior	Classificação
1	99,34	93,00	90,24	61,36	68,21	1
2	91,42	95,00	61,79	59,01	63,72	2
3	84,12	80,00	42,28	36,70	47,45	7
4	86,06	70,00	39,84	33,11	46,36	9
5	100,00	72,00	100,00	43,01	52,80	6
6	89,89	85,00	58,54	48,56	57,87	5
7	98,32	50,00	88,62	19,13	23,20	10
8	87,31	86,00	55,28	49,95	58,80	3
9	88,40	85,00	78,86	48,70	57,82	4
10	78,05	65,00	52,03	31,88	51,81	8

Figura 27 – Cenário 1 - Classificação da Preferência HML

A classificação dos dez recursos empregados neste cenário está caracterizada na Figura 28, sendo que o eixo horizontal representa o ID do recurso e o eixo vertical, a classificação obtida com o algoritmo WsRF e com o Classificador Fuzzy do EXEHDA-RR com a Função de Pertinência AV definida pelo especialista.

O emprego da Lógica Fuzzy Tipo-2 possibilita o tratamento de incertezas que a lógica tradicional não permite. A definição das preferências do cliente através de variáveis linguísticas, associada às funções de pertinência definidas por especialistas do domínio permite uma classificação de recursos mais apurada e condizente ao ambiente computacional em questão. É possível observar, na Figura 28, que a maior diferença na classificação ocorreu com o recurso ID 05. Este recurso possui excelente RT (219,43), atende a preferência RT - High, mas o valor de AV (72) não atende a preferência AV - Medium, de acordo com a função de pertinência especificada pelo especialista. Isso justifica a grande diferença na classificação do recurso ID 05: com o algoritmo MCDA WsRF, o recurso recebeu a classificação 2; com o Classificador Fuzzy, recebeu a classificação 6.

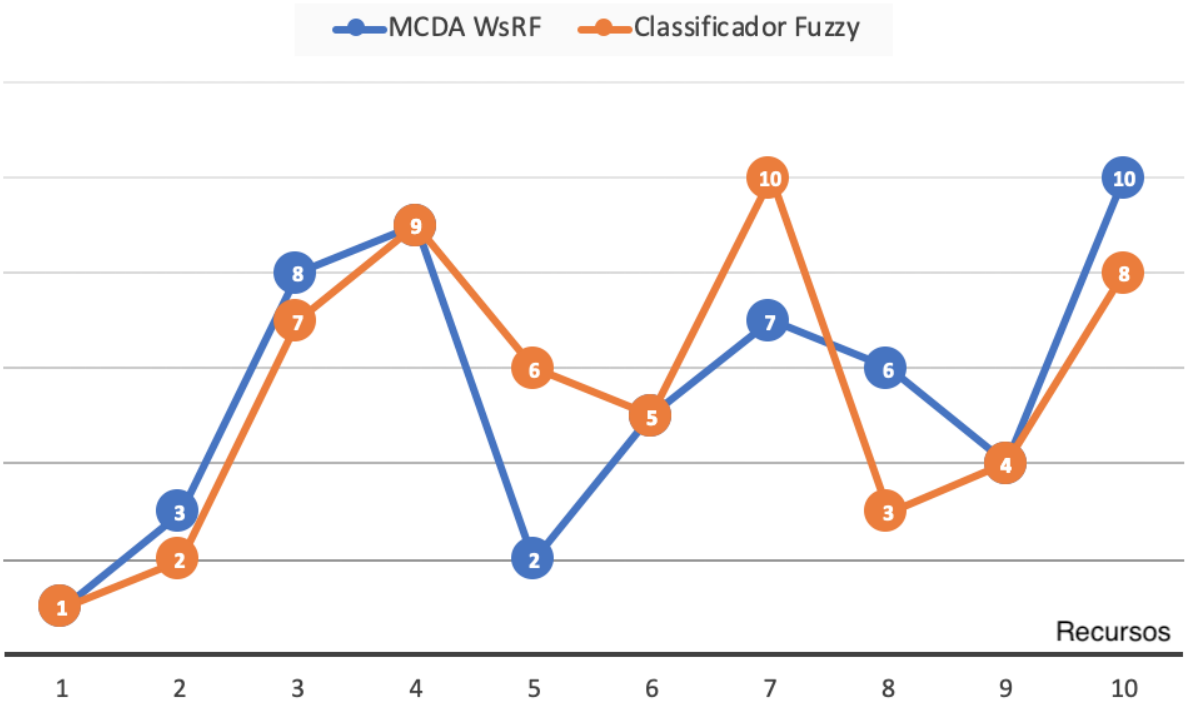

Figura 28 – Cenário 1 - Comparação dos Classificadores Fuzzy e WsRF

5.2 Cenário 2 - Classificação sem Persistência na Requisição do Cliente

Este cenário contempla o esforço de classificação de recursos a partir de diferentes preferências dos clientes. Nesse sentido, o Componente Cliente envia oito requisições pelo recurso “AnalysisWSAppLabImplService”, sendo que em cada requisição é aplicada uma preferência diferente para os atributos de QoS RT, AV e TH.

O Componente Diretório processa a requisição do cliente e o resultado da query em busca do recurso “AnalysisWSAppLabImplService” retorna 102 recursos que podem atender a requisição do cliente. Esses recursos são classificados pelo Classificador MCDA, que analisa os atributos de qualidade. Os melhores recursos são classificados como “1” e os piores, como “4”. O Classificador MCDA retorna 13 recursos (Tabela 28) como sendo os melhores, que são enviados ao Classificador Fuzzy para aplicar as preferências do cliente.

No Classificador Fuzzy os atributos são normalizados, conforme procedimento apresentado na Seção 4.2.3, antes de serem processados pelo *framework* Juzzy.

A Tabela 28 apresenta os 13 recursos resultantes do classificador MCDA, com os valores originais e normalizados dos atributos de QoS. Os URIs dos recursos foram substituídos por IDs para facilitar a visualização dos dados nas tabelas.

A Tabela 29 contém a classificação final dos 13 recursos, considerando as seguintes preferências do cliente: HHH (RT - High, AV - High e TH - High), LHM (RT - Low, AV - High e TH - Medium), LHL (RT - Low, AV - High e TH - Low) e MHM (RT - Me-

Tabela 28 – Cenário 2 - Recursos Selecionados pelo Classificador MCDA

	Original			Normalizado		
ID	**RT**	**AV**	**TH**	**RT**	**AV**	**TH**
01	221,48	90	10,9	99,77	90	98,2
02	239,22	85	7,7	92,37	85	69,37
03	261,09	85	5,4	84,63	85	48,65
04	256,39	85	5,1	86,18	85	45,95
05	220,96	87	11,1	100	87	100
06	243,91	85	7,1	90,59	85	63,96
07	223,83	92	10,8	98,72	92	97,3
08	250,74	86	7	88,12	86	63,06
09	246,43	85	9,7	89,66	85	87,39
10	277,35	86	6,3	79,67	86	56,76
11	280,39	86	6,4	78,8	86	57,66
12	275,87	85	6,5	80,1	85	58,56
13	267,43	86	5,7	82,62	86	51,35

dium, AV - High e TH - Medium). Nessa Tabela, cada preferência está organizada em três colunas: o ID do recurso (ID), o intervalo resultante da defuzzificação (Defuzz.) e o ranqueamento obtido (R). O valor “1” refere-se ao melhor recurso e o “13”, ao pior recurso.

Tabela 29 – Cenário 2 - Classificação com Preferências - HHH, LMH, LHL e MHM

	HHH		LMH		LHL		MHM	
ID	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**
01	[77,75 - 84,24]	2	[79,33 - 85,09]	2	[77,75 - 84,24]	2	[77,75 - 84,24]	2
02	[36,27 - 71,01]	5	[36,27 - 71,01]	5	[53,34 - 76,26]	5	[54,96 - 76,26]	5
03	[35,02 - 57,77]	12	[35,02 - 57,77]	12	[48,47 - 67,39]	13	[49,56 - 67,36]	13
04	[28,08 - 61,51]	13	[28,08 - 61,51]	13	[45,92 - 72,18]	8	[46,71 - 72,18]	12
05	[77,02 - 83,94]	3	[79,60 - 85,27]	1	[77,02 - 83,94]	3	[77,02 - 83,94]	3
06	[42,25 - 58,84]	6	[42,25 - 58,84]	6	[53,59 - 65,91]	7	[55,44 - 65,86]	9
07	[78,19 - 84,45]	1	[79,18 - 85,01]	3	[78,19 - 84,45]	1	[78,19 - 84,45]	1
08	[42,66 - 58,01]	7	[42,66 - 58,08]	7	[54,15 - 65,48]	6	[55,82 - 65,44]	10
09	[76,48 - 83,77]	4	[77,12 - 83,98]	4	[76,48 - 83,77]	4	[76,48 - 83,77]	4
10	[41,03 - 57,52]	10	[41,03 - 57,52]	10	[53,90 - 63,23]	10	[56,64 - 67,92]	7
11	[41,22 - 58,06]	9	[41,22 - 58,06]	9	[54,19 - 63,14]	9	[57,29 - 68,48]	6
12	[41,80 - 57,87]	8	[41,80 - 57,87]	8	[53,81 - 63,18]	11	[56,51 - 67,32]	8
13	[38,34 - 56,61]	11	[38,34 - 56,61]	11	[51,16 - 65,21]	12	[52,72 - 66,37]	11

A Figura 29 apresenta a interface de consulta por recursos do EXEHDA-RR. Essa interface possui apenas um campo de consulta simplificado para a descrição do recurso desejado pelo cliente, suficiente para demonstrar a classificação de recursos do EXEHDA-RR.

Nessa interface, o cliente também deve informar se deseja ou não a persistência da consulta pelo recurso desejado. Nesse cenário, o cliente não deseja persistência. Após, o cliente deve informar as preferências para os atributos de QoS RT, AV

Figura 29 – Cenário 2 - Busca por Recursos

e TH. A Figura 30 apresenta a seleção das preferências HLM com o perfil "*default*". Após a seleção das preferências e o perfil desejado o cliente, este clica em “Aplicar preferências” e a consulta é processada uma única vez.

Figura 30 – Cenário 2 - Especificação da Preferência HLM

O resultado da classificação é apresentado na Figura 31. Os recursos estão ordenados por ID e a classificação obtida é apresentada na última coluna.

A Tabela 30 contém a classificação final dos 13 recursos, considerando mais quatro preferências do cliente: HLL (RT - High, AV - Low e TH - Low); HHL (RT - High, AV - High e TH - Low); MLH (RT - Medium, AV - Low e TH - High); e HLM (RT - High, AV - Low e TH - Medium).

Figura 31 – Cenário 2 - Resultado da Classificação com Preferência HLM

Tabela 30 – Cenário 2 - Classificação com Preferências - HLL, HHL, MLH e HLM

	HLL		HHL		MLH		HLM	
ID	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**
01	[79,56 - 85,24]	2	[77,75 - 84,24]	2	[79,33 - 85,09]	2	[79,56 - 85,24]	2
02	[56,69 - 77,91]	5	[54,96 - 76,26]	5	[36,27 - 71,04]	5	[56,69 - 77,91]	5
03	[48,26 - 67,28]	9	[49,33 - 67,24]	12	[35,02 - 57,77]	12	[48,26 - 67,28]	9
04	[46,62 - 72,53]	8	[46,71 - 72,18]	8	[28,08 - 61,51]	13	[46,62 - 72,53]	8
05	[79,60 - 85,27]	1	[77,02 - 83,94]	3	[79,60 - 85,27]	1	[79,60 - 85,27]	1
06	[55,99 - 67,51]	6	[55,44 - 65,86]	6	[42,25 - 58,84]	6	[55,99 - 67,51]	6
07	[79,41 - 85,14]	3	[78,19 - 84,45]	1	[79,18 - 85,01]	3	[79,41 - 85,14]	3
08	[55,08 - 66,09]	7	[55,82 - 65,44]	7	[42,66 - 58,08]	7	[55,08 - 66,09]	7
09	[77,67 - 84,20]	4	[76,48 - 83,77]	4	[77,12 - 83,98]	4	[77,67 - 84,20]	4
10	[50,55 - 61,45]	12	[52,35 - 65,67]	11	[41,03 - 57,52]	10	[50,55 - 61,45]	12
11	[50,33 - 61,12]	13	[52,21 - 65,87]	10	[41,22 - 58,06]	9	[50,33 - 61,12]	13
12	[51,20 - 61,79]	11	[53,21 - 65,56]	9	[41,80 - 57,87]	8	[51,20 - 61,79]	11
13	[49,38 - 64,21]	10	[50,68 - 65,24]	13	[38,34 - 56,61]	11	[49,38 - 64,21]	10

A Tabela 31 compara a classificação final de 13 recursos selecionados pelo Classificador MCDA. A classificação com emprego de Lógica Fuzzy Tipo-2 é demonstrada com as seguintes preferências: (HHH) RT - High, AV - High, TH - High; (LHM) RT - Low, AV - High, TH - Medium; (LHL) RT - Low, AV - High e TH - Low; (MHM) RT - Medium, AV - High, TH - Medium; (HLL) RT - High, AV - Low, TH - Low; (HHL) RT - High, AV - High, TH - Low; (MLH) RT - Medium, AV - Low, TH - High; e (HLM) RT - High, AV - Low, TH - Medium.

A Figura 32 apresenta as classificações obtidas por cada recurso, de acordo com a preferência estabelecida pelo cliente. Os recursos estão representados pelos IDs de 1 a 14 no eixo horizontal da figura, e a classificação obtida pelo recurso é apresentada

Tabela 31 – Cenário 2 - Classificação Considerando todas Preferências de Cliente

ID	RT	AV	TH	HHH	LMH	LHL	MHM	HLL	HHL	MLH	HLM
01	221,48	90	10,9	2	2	2	2	2	2	2	2
02	239,22	85	7,7	5	5	5	5	5	5	5	5
03	261,09	85	5,4	12	12	13	13	9	12	12	9
04	256,39	85	5,1	13	13	8	12	8	8	13	8
05	220,96	87	11,1	3	1	3	3	1	3	1	1
06	243,91	85	7,1	6	6	7	9	6	6	6	6
07	223,83	92	10,8	1	3	1	1	3	1	3	3
08	250,74	86	7	7	7	6	10	7	7	7	7
09	246,43	85	9,7	4	4	4	4	4	4	4	4
10	277,35	86	6,3	10	10	10	7	12	11	10	12
11	280,39	86	6,4	9	9	9	6	13	10	9	13
12	275,87	85	6,5	8	8	11	8	11	9	8	11
13	267,43	86	5,7	11	11	12	11	10	13	11	10

no eixo vertical.

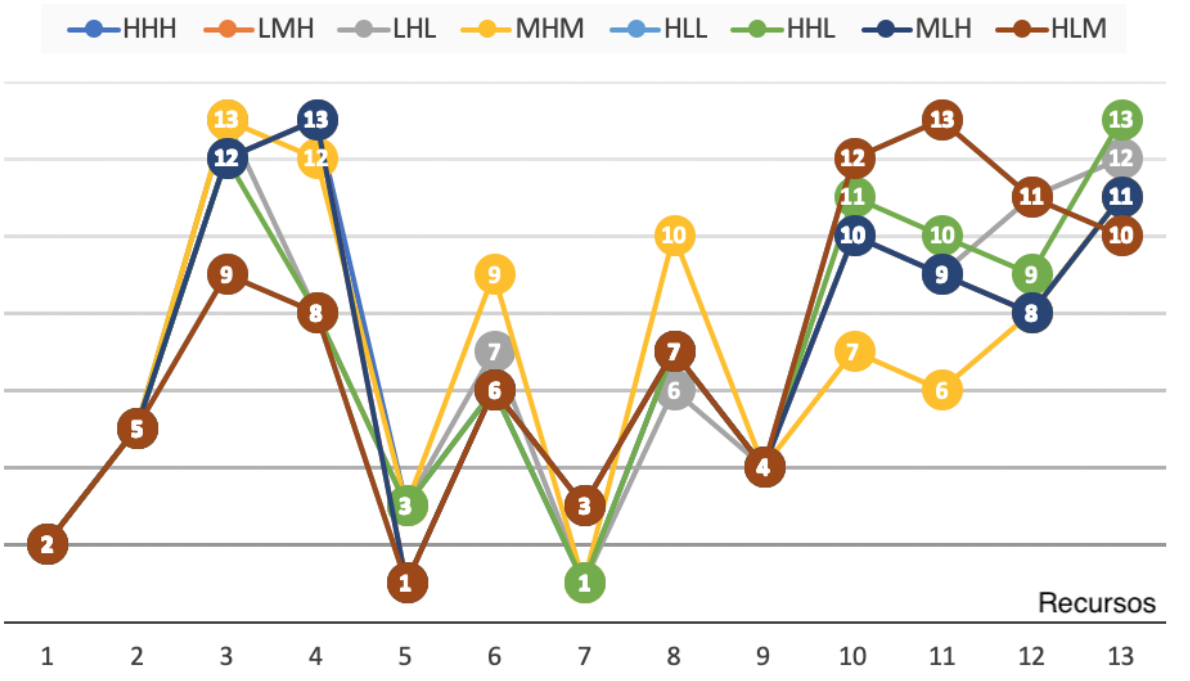

Figura 32 – Cenário 2 - Ranking do Classificador Fuzzy Considerando Preferências de Cliente

Pode-se observar que as preferências definidas com o atributo Availability com H (High), ou seja, alta importância na preferência do cliente classificaram como melhor recurso “1” o recurso de ID 07 (preferências HHH, LHM, LHL, MHM e HHL).

5.3 Cenário 3 - Classificação Considerando Persistência na Requisição do Cliente

Este cenário avalia também o procedimento de classificação a partir de preferências definidas pelo cliente, entretanto, considera, ainda, o emprego de persistência na requisição por recursos, informada no momento da requisição, juntamente com as preferências pelos atributos de QoS.

O serviço Pré-Classificador ML é acionado e o algoritmo de *Machine Learning* LMT é treinado com os dados resultantes do Classificador MCDA. A entrada de novos recursos no ambiente é monitorada pelo período de tempo informado pelo cliente e os recursos são classificados pelo algoritmo LMT.

Nesse cenário, foi considerada a presença de um novo recurso com ID 14, do tipo “AnalysisWSAppLabImplService”, com os seguintes atributos de QoS: RT 290, AV 95 e TH 9. O algoritmo LMT, com o treinamento já realizado, classifica o ID 14 como “1” e, desse modo, é enviado ao Classificador Fuzzy para ser aplicada a preferência do cliente; Assim, a classificação final é refeita.

A Figura 33 apresenta a interface de Consulta de Recursos. Nesse cenário, o recurso consultado e as preferências aplicadas são os mesmos do Cenário 2, mas com persistência de 120 minutos na consulta. Durante esse período de tempo, que é estipulado pelo cliente, o serviço Pré-Classificador ML monitora a entrada de novos recursos, classificando-os. Os recursos classificados como melhores são enviados ao serviço Classificador Fuzzy.

Figura 33 – Cenário 3 - Busca por Recursos

A classificação após a entrada de um novo recurso é apresentada na Figura 34.

Nessa classificação, apresentada na Figura 34, foram consideradas as seguintes preferências do cliente: HHH (RT - High, AV - High e TH - High), LHM (RT - Low, AV

Figura 34 – Cenário 3 - Resultado da Classificação com Preferência HLM

- High e TH - Medium), LHL (RT - Low, AV - High e TH - Low) e MHM (RT - Medium, AV - High e TH - Medium). A Tabela 32 apresenta o ID do recurso (ID), o intervalo resultante da defuzzificação (Defuz.) e o ranqueamento obtido (R). O valor “1” refere-se ao melhor recurso e o “14”, ao pior recurso.

Tabela 32 – Cenário 3 - Classificação com Preferências - HHH, LMH, LHL e MHM

	HHH		LMH		LHL		MHM	
ID	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**
01	[77,75 - 84,24]	2	[79,33 - 85,09]	2	[77,75 - 84,24]	2	[77,75 - 84,24]	2
02	[36,27 - 71,01]	5	[36,27 - 71,01]	6	[53,34 - 76,26]	6	[54,96 - 76,26]	6
03	[35,02 - 57,77]	13	[35,02 - 57,77]	13	[48,47 - 67,39]	14	[49,56 - 67,36]	14
04	[28,08 - 61,51]	14	[28,08 - 61,51]	14	[45,92 - 72,18]	9	[46,71 - 72,18]	13
05	[77,02 - 83,94]	3	[79,60 - 85,27]	1	[77,02 - 83,94]	3	[77,02 - 83,94]	3
06	[42,25 - 58,84]	7	[42,25 - 58,84]	7	[53,59 - 65,91]	8	[55,44 - 65,86]	10
07	[78,19 - 84,45]	1	[79,18 - 85,01]	3	[78,19 - 84,45]	1	[78,19 - 84,45]	1
08	[42,66 - 58,08]	8	[42,66 - 58,08]	8	[54,15 - 65,48]	7	[55,82 - 65,44]	11
09	[76,48 - 83,77]	4	[77,12 - 83,98]	4	[76,48 - 83,77]	4	[76,48 - 83,77]	4
10	[41,03 - 57,52]	11	[41,03 - 57,52]	11	[53,90 - 63,23]	11	[56,64 - 67,92]	8
11	[41,22 - 58,06]	10	[41,22 - 58,06]	10	[54,19 - 63,14]	10	[57,29 - 68,48]	7
12	[41,80 - 57,87]	9	[41,80 - 57,87]	9	[53,81 - 63,18]	12	[56,51 - 67,32]	9
13	[38,34 - 56,61]	12	[38,34 - 56,61]	12	[51,16 - 65,21]	13	[52,72 - 66,37]	12
14	[49,63 - 83,91]	5	[54,87 - 83,66]	5	[62,38 - 84,87]	5	[62,38 - 84,87]	5

A Tabela 33 contém a classificação final dos 14 recursos, considerando mais quatro preferências do cliente: HLL (RT - High, AV - Low e TH - Low), HHL (RT - High, AV - High e TH - Low), MLH (RT - Medium, AV - Low e TH - High) e HLM (RT - High, AV - Low e TH - Medium).

Por sua vez, a Tabela 34 apresenta as classificações obtidas com a entrada do

Tabela 33 – Cenário 3 - Classificação com Preferências - HLL, HHL, MLH e HLM

	HLL		HHL		MLH		HLM	
ID	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**	**Defuzz.**	**R**
01	[79,56 - 85,24]	2	[77,75 - 84,24]	2	[79,33 - 85,09]	2	[79,56 - 85,24]	2
02	[56,69 - 77,91]	5	[54,96 - 76,26]	6	[36,27 - 71,04]	6	[56,69 - 77,91]	5
03	[48,26 - 67,28]	10	[49,33 - 67,24]	13	[35,02 - 57,77]	13	[48,26 - 67,28]	10
04	[46,62 - 72,53]	9	[46,71 - 72,18]	9	[28,08 - 61,51]	14	[46,62 - 72,53]	9
05	[79,60 - 85,27]	1	[77,02 - 83,94]	3	[79,60 - 85,27]	1	[79,60 - 85,27]	1
06	[55,99 - 67,51]	7	[55,44 - 65,86]	7	[42,25 - 58,84]	7	[55,99 - 67,51]	7
07	[79,41 - 85,14]	3	[78,19 - 84,45]	1	[79,18 - 85,01]	3	[79,41 - 85,14]	3
08	[55,08 - 66,09]	8	[55,82 - 65,44]	8	[42,66 - 58,08]	8	[55,08 - 66,09]	8
09	[77,67 - 84,20]	4	[76,48 - 83,77]	4	[77,12 - 83,98]	4	[77,67 - 84,20]	4
10	[50,55 - 61,45]	13	[52,35 - 65,67]	12	[41,03 - 57,52]	11	[50,55 - 61,45]	13
11	[50,33 - 61,12]	14	[52,21 - 65,87]	11	[41,22 - 58,06]	10	[50,33 - 61,12]	14
12	[51,20 - 61,79]	12	[53,21 - 65,56]	10	[41,80 - 57,87]	9	[51,20 - 61,79]	12
13	[49,38 - 64,21]	11	[50,68 - 65,24]	14	[38,34 - 56,61]	12	[49,38 - 64,21]	11
14	[49,63 - 83,91]	6	[49,63 - 83,91]	5	[54,87 - 83,66]	5	[49,63 - 83,91]	6

novo recurso de ID 14, aplicando-se as oito diferentes preferências do cliente.

Tabela 34 – Cenário 3 - Classificação Considerando todas Preferências de Cliente

ID	**RT**	**AV**	**TH**	**HHH**	**LMH**	**LHL**	**MHM**	**HLL**	**HHL**	**MLH**	**HLM**
01	221,48	90	10,9	2	2	2	2	2	2	2	2
02	239,22	85	7,7	6	6	6	6	5	6	6	5
03	261,09	85	5,4	13	13	14	14	10	13	13	10
04	256,39	85	5,1	14	14	9	13	9	9	14	9
05	220,96	87	11,1	3	1	3	3	1	3	1	1
06	243,91	85	7,1	7	7	8	10	7	7	7	7
07	223,83	92	10,8	1	3	1	1	3	1	3	3
08	250,74	86	7	8	8	7	11	8	8	8	8
09	246,43	85	9,7	4	4	4	4	4	4	4	4
10	277,35	86	6,3	11	11	11	8	13	12	11	13
11	280,39	86	6,4	10	10	10	7	14	11	10	14
12	275,87	85	6,5	9	9	12	9	12	10	9	12
13	267,43	86	5,7	12	12	13	12	11	14	12	11
14	290,00	95	9	5	5	5	5	6	5	5	6

A classificação dos diferentes recursos que consideram a entrada de um novo recurso no ambiente computacional, está sintetizada na Figura 35. Os recursos estão representados pelos IDs de 1 a 14 no eixo horizontal da figura, sendo a classificação obtida pelo recurso apresentada no eixo vertical.

O novo recurso que foi pré-classificado e identificado pelo ID 14 foi classificado como “5” na maioria das preferências do cliente. Esse recurso possui um RT (290,00) com valor alto, representando baixo desempenho. Para preferências HLL e HLM, que exigem um RT High, o recurso ficou classificado em “6”. A preferência HHL classificou o recurso em “5” porque o atributo AV possuir um valor High, melhorando a classificação do recurso.

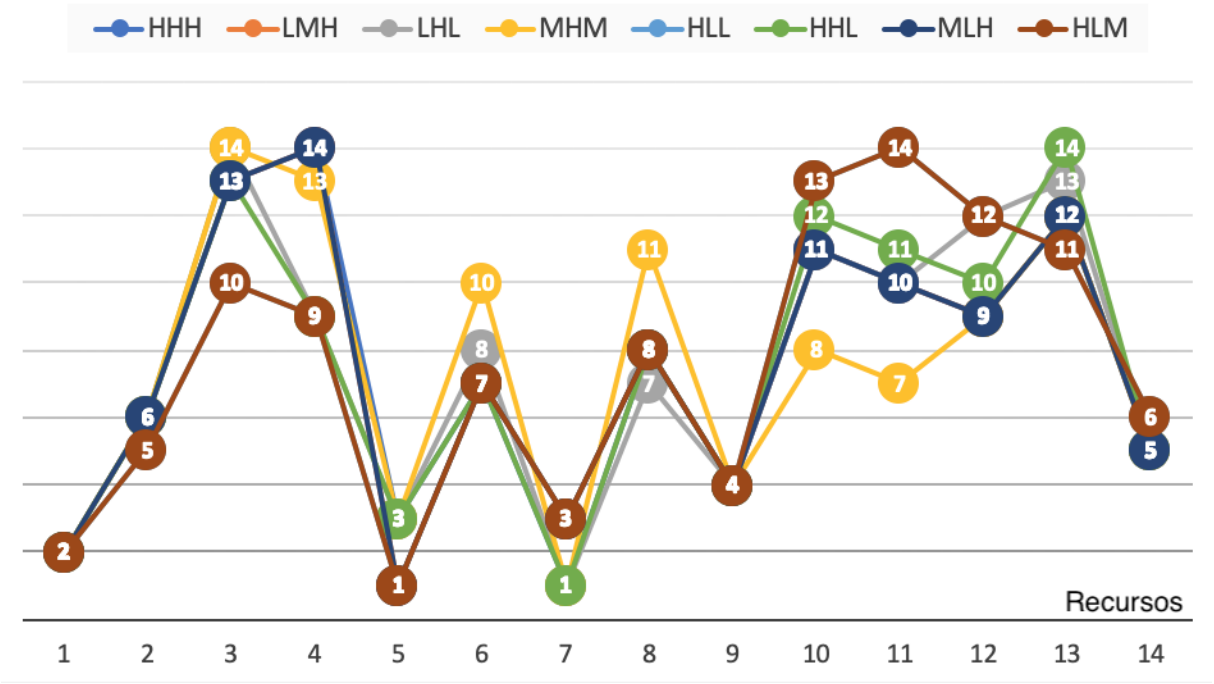

Figura 35 – Cenário 3 - Ranking do Classificador Fuzzy Considerando Preferências do Cliente (Com a Inclusão de Novo Recurso)

5.4 Considerações Sobre o Capítulo

Este Capítulo apresentou a avaliação das funcionalidades do EXEHDA-RR por meio de três cenários de uso.

O primeiro cenário comparou a classificação de um conjunto de recursos pelo algoritmo MCDA WsRF, com aquela obtida pelo Classificador Fuzzy do EXEHDA-RR. Foi possível constatar que o emprego de perfis com funções de pertinência adequadas ao contexto da aplicação do cliente qualifica o resultado da classificação, resultando na indicação de recursos mais adequados às requisições dos clientes.

O segundo cenário avaliou a requisição do cliente por um determinado recurso, aplicando oito diferentes preferências. Nesse cenário, é demonstrada a atuação do serviço Classificador MCDA e do serviço Classificador Fuzzy.

Por sua vez, no terceiro cenário foi realizada uma avaliação que considerou a persistência na requisição do cliente. Dessa forma, o serviço Pré-classificador *Machine Learning* foi ativado, sempre que necessário, para monitorar a entrada de novos recursos, sendo possível observar que ocorreu uma minimização dos esforços de comparação extensiva.

O próximo e último Capítulo contempla as considerações finais deste trabalho, sendo discutidas as principais conclusões atingidas, relacionadas as publicações realizadas durante o doutorado, bem como são destacados alguns potenciais trabalhos futuros relacionados à Tese.

6 CONSIDERAÇÕES FINAIS

Este Capítulo sintetiza as principais conclusões obtidas durante a concepção do EXEHDA-RR, relaciona as publicações realizadas durante os esforços de estudo e pesquisa da Tese, bem como apresenta perspectivas de continuidade dos trabalhos na área, indicando frentes de trabalhos futuros.

6.1 Principais Conclusões

A revisão de literatura aponta para uma elevada convergência entre academia e indústria quando da consolidação de tecnologias e padrões para uso na IoT. Desta convergência decorreu um crescimento elevado, nos últimos anos, do número de equipamentos que provém serviços na IoT, muitos deles com perfis aproximados.

Foi possível constatar, como decorrência dessa situação, que o processo de classificação de recursos na Internet das Coisas pode tornar-se complexo, pois, além de ser necessário manipular uma grande quantidade de recursos, poderá haver incertezas, por parte do cliente, quanto aos parâmetros de QoS, tendo como critérios os requisitos da sua aplicação. Outrossim, tais recursos, na perspectiva operacional da IoT, podem ter um comportamento dinâmico quanto a sua disponibilidade ao longo tempo.

Considerando cenário como o apresentado, decorre a questão de pesquisa considerada durante o desenvolvimento desta Tese, qual seja: Como classificar recursos descobertos na infraestrutura computacional provida pela IoT?

Nesse sentido, concebeu-se uma arquitetura de software para classificação de recursos, denominada EXEHDA-RR. Para tanto, são minimizadas as incertezas do cliente quanto à especificação dos intervalos para cada atributo de QoS dos recursos por meio da Lógica Fuzzy Tipo-2 Intervalar.

A estratégia concebida de contar com o conhecimento de um especialista na definição dos conjuntos fuzzy, os quais oportunizam uma otimização da definição de preferências por parte dos clientes, mostrou-se oportuna. Para tanto, optou-se para o EXEHDA-RR pelo emprego de técnicas denominadas como subjetivas, nas quais um especialista do domínio da aplicação informa as funções de pertinência, caracteri-

zando os intervalos para os termos low, medium e high.

Tendo por base promover uma sinergia com trabalhos desenvolvidos ou em andamento no PPGC/UFPEL, a concepção do EXEHDA-RR teve por premissa a integração da sua arquitetura, à plataforma de software do *middleware* EXEHDA. Desse modo, a fundamentação teórica apresentada no Capítulo 2, discutiu conceitos sobre IoT, Descoberta de Recursos, Qualidade de Serviço e Abordagens para Classificação de Recursos, sendo caracterizada a abrangência da Tese considerando os esforços atuais do grupo de pesquisa.

Como um dos requisitos para a concepção do EXEHDA-RR foi realizada uma Revisão Sistemática da Literatura sobre o processo de classificação de recursos, para a qual foram definidas questões de pesquisa com o objetivo de identificar nos trabalhos atuais, técnicas, metodologias e desafios de pesquisa. A RSL trouxe subsídios oportunos para o entendimento do estado da arte no tema da Tese, seja pela revisão detalhada dos artigos selecionados, seja, ainda, por todo o trabalho associado ao tratamento dos diferentes artigos localizados nos indexadores, que contribuiu para uma visão horizontal da área de pesquisa.

Observando as tendências de pesquisa na área, identificadas quando do desenvolvimento da RSL, foi então proposta a arquitetura de software do EXEHDA-RR, cujos componentes, bem como sua dinâmica operacional, foram caracterizados no Capítulo 4. É importante ressaltar que, dentre as premissas de concepção, consideraram-se as características típicas da IoT no tocante a seu modo de operação, e que, apesar dos cuidados para prover suporte ao *middleware* EXEHDA, entende-se que o EXEHDA-RR tem potencial para ser integrado a outros *middlewares*, desde que estes explorem o conceito de célula de execução.

A decisão sobre o método para avaliação do EXEHDA-RR levou em conta aqueles observados nos trabalhos relacionados, bem como a tendência analisada quando da revisão da literatura sobre o tratamento da incerteza na classificação de recursos. Desse modo, foram contemplados três cenários de uso, que exploram diferentes funcionalidades da arquitetura concebida.

No primeiro cenário de uso, foi explorado o serviço Classificador Fuzzy do EXEHDA-RR, quando um especialista define as funções de pertinência que representam os termos linguísticos low, medium e high, referentes ao atributo de QoS Availability. A partir dessa definição pelo especialista, criou-se um perfil contendo um novo conjunto de funções de pertinência. O resultado do Classificador Fuzzy utilizando este perfil gerado foi comparado com o algoritmo MCDA WsRF.

Empregando-se as mesmas preferências do cliente em ambos classificadores, foi possível observar que os resultados do classificador fuzzy se mostraram-se mais próximos da situação real. Isso acontece porque o algoritmo MCDA realiza seus cálculos baseado em médias, diferentemente do Classificador Fuzzy que trabalha com interva-

los definidos pelo especialista.

No segundo cenário de uso, foi avaliada a requisição do cliente por um determinado recurso, aplicando-se oito diferentes preferências do cliente. A requisição do cliente retornou 102 recursos aptos, sendo que o serviço Classificador MCDA reduziu para 13 recursos, e o Classificador Fuzzy aplicou as preferências do cliente.

Por fim, um terceiro cenário de uso teve como propósito demonstrar o funcionamento do serviço Pré-Classificador *Machine Learning*. Neste cenário, foi realizada a mesma requisição do cenário 2, mas com a persistência na requisição por 120 minutos. Caracterizou-se a entrada de um novo recurso no ambiente durante este período de tempo informado. Tal recurso foi pré-classificado e enviado ao Classificador Fuzzy que refez a classificação final considerando o novo recurso. Foi possível observar que a abordagem de pré-classificação proposta mostrou-se oportuna no tratamento da entrada de novos recursos no ambiente computacional.

A Tabela 35 relaciona o EXEHDA-RR com os trabalhos selecionados pela Revisão Sistemática da Literatura e discutidos na Seção 3.2. As Questões de Qualidade, especificadas no Capítulo 3, foram utilizadas como critérios para uma comparação resumida. A presença do critério foi identificada com o símbolo "+" e a sua ausência com o símbolo "–": (i) QQ1 - utiliza algoritmos MCDA; (ii) QQ2 - trata de alguma forma a incerteza; (iii) QQ3 - emprega técnicas para predição e/ou pré-classificação de recursos; (iv) QQ4 - emprega tecnologias de Web Semântica; (v) QQ5 - apresenta resultados da classificação; e (vi) QQ6 - considera as preferências do cliente.

Tabela 35 – Comparação com os Trabalhos Relacionados

Trabalhos	**QQ1**	**QQ2**	**QQ3**	**QQ4**	**QQ5**	**QQ6**
GOHAR; PUROHIT, 2016	+	+	–	+	+	+
KUMAR; MISHRA; KUMAR, 2017	+	+	–	–	+	+
WANG et al., 2015	+	–	–	–	+	+
SALAH; SAADI, 2016	+	+	+	–	+	+
LIU; XIAO; LIN, 2013	+	+	–	–	+	+
PATINIOTAKIS; VERGINADIS; MENTZAS, 2015	+	+	–	+	+	+
SUCHITHRA; RAMAKRISHNAN, 2016	+	+	–	–	+	+
EXEHDA-RR	+	+	+	+	+	+

Considerando as características do cenário da IoT e seu impacto nos esforços de concepção do EXEHDA-RR, bem como os respectivos desdobramentos na sua arquitetura de software, entende-se como principais contribuições desta Tese os aspectos indicados a seguir:

- Considerar as preferências dos clientes - o Cliente pode especificar suas preferências na requisição por recursos informando os requisitos funcionais, des-

crevendo o tipo de recurso com suas características e também os requisitos não funcionais, tais como disponibilidade, tempo de resposta, etc. A inclusão dos requisitos não funcionais quando da requisição por recursos, promove uma classificação dos mesmos com melhor potencial de atender as preferências do cliente.

- Prover tratamento à incerteza na especificação das preferências do cliente - o emprego de variáveis linguísticas para representar os atributos de QoS e termos linguísticos para associar a relevância do atributo para o cliente reduz possíveis incertezas provenientes do desconhecimento de intervalos e escalas de medidas. Por meio de funções de pertinência os especialistas definem os intervalos adequados para os termos baixo (low), médio (medium) e alto (high). A incerteza pode ocorrer na informação incorreta ou imprecisa definida nas preferências do cliente, bem como atributos de qualidade faltantes ou incompletos (com emprego de Lógica Fuzzy Tipo-2).

- Dar suporte à operação em regime de elevada escalabilidade - foi desenvolvido um Pré-Classificador de recursos com o algoritmo de árvore de decisão LMT na classificação de novos recursos detectados no ambiente, evitando-se que seja necessário recalcular todos os atributos de qualidade, pré-classificando-se novos recursos sem a necessidade de compará-los com os existentes no repositório.

- Facultar integração com *middlewares* para IoT - a arquitetura de software do EXEHDA-RR considera atributos de QoS como requisitos não funcionais. Dessa forma, esta proposta para descoberta e classificação de recursos poderá ser incorporada a outros *middlewares* que operem em ambientes de elevada escalabilidade e distribuição.

Uma visão geral dos diferentes esforços de estudo e pesquisa desenvolvidos durante o doutorado, inclusive contemplando uma visão cronológica, está disponível no wiki http://ubiq.inf.ufpel.edu.br/dilli/.

6.2 Publicações Realizadas

Os resultados da pesquisa desenvolvida até o momento foram publicados e estão organizados em publicações como primeiro autor e coautor.

6.2.1 Publicações como Primeiro Autor

- DILLI, R.; ARGOU, A.; REISER, R.; YAMIN, A. Uncertainty Handling with Type-2 Fuzzy Logic in IoT Resource Classification. In: **Applied Soft Computing Journal**, 2020 (em revisão).

- DILLI, R.; ARGOU, A.; PILLA, M.; PERNAS, A.; REISER, R.; YAMIN, A. Fuzzy Logic and MCDA in IoT Resources Classification. In: **The 33rd ACM/SIGAPP Symposium On Applied Computing**. Pau, France: SAC, 2018.

- DILLI, R.; ARGOU, A.; PERNAS, A. M.; REISER, R.; YAMIN, A. C. EXEHDA-RR: Uma proposta para tratar incertezas e otimizar o processo de classificação de recursos na IoT. In: **X SBCUP (CSBC 2018 - Simpósio Brasileiro de Computação Ubíqua e Pervasiva)**. Natal - RN: CSBC, 2018.

- DILLI, R.; ARGOU, A.; REISER, R.; YAMIN, A. C. IoT Resources Ranking: Decision Making under Uncertainty Combining Machine Learning and Fuzzy Logic. In: **NAFIPS'2018 (37th Annual Conference of the North American Fuzzy Information Processing Society)**. Fortaleza - CE: NAFIPS, 2018.

- DILLI, R.; ARGOU, A.; REISER, R.; YAMIN, A. C. Autonomic Ranking of Resources in IoT Exploring Fuzzy Logic and Machine Learning. In: **CLEI 2018 (XLIV Latin American Computing Conference)**. São Paulo - SP: CLEI, 2018.

- DILLI, R.; FILHO, H. K.; DE AZEVEDO, M. M.;PERNAS, A. M.; YAMIN, A. C. Uma Abordagem Dinâmica para Descoberta de Recursos na IoT Explorando Processamento Semântico. In: **IX SBCUP (CSBC 2017 - Simpósio Brasileiro de Computação Ubíqua e Pervasiva)**. São Paulo - SP: CSBC, 2017.

- DILLI, R.; FILHO, H.; PERNAS, A.; YAMIN, A. EXEHDA-RR: Machine Learning and MCDA with Semantic Web in IoT Resources Classification. In: **WebMedia 2017 - Symposium on Multimedia and the Web**. Gramado - RS: WebMedia, 2017.

6.2.2 Publicações como Coautor

- ARGOU, A.; DILLI, R.; REISER, R.; YAMIN, A. Exploring Type-2 Fuzzy Logic with Dynamic Rules in IoT Resources Classification. In: **FUZZ-IEEE 2019 - International Conference on Fuzzy Systems**. New Orleans - USA: FUZZ-IEEE, 2019.

- GRUENDEMANN, F.; BURTET, J.; DILLI, R.; PERNAS, A.; YAMIN, A. Avaliação do Emprego de Algoritmos de Árvores de Decisão para a Classificação de Recursos na IoT. In: **ERAD 2018 - XVIII - Escola Regional de Alto Desempenho**. Porto Alegre - RS: ERAD, 2018.

- JOAO, L.; KAISER, H.; PILLA, M.; DILLI, R.; PERNAS, A.; TABIM, V.; YAMIN, A. Avaliando o Uso de Fog-Computing no Servidor de Borda do Middleware EXEHDA. In: **ERAD 2018 - XVIII - Escola Regional de Alto Desempenho**. Porto Alegre - RS: ERAD, 2018.

- BURTET, J.; KAISER, H.; DILLI, R.; GRUENDEMANN, F.; PERNAS, A.; YAMIN, A. Ranqueamento de Recursos na Internet das Coisas Explorando Algoritmos MCDA. In: **ERAD 2018 - XVIII - Escola Regional de Alto Desempenho**. Porto Alegre - RS: ERAD, 2018.

- KAISER, H.; DILLI, R.; PERNAS, A.; YAMIN, A. Uma Proposta Multicelular Hierárquica para a Localização de Recursos na IoT. In: **ERAD 2018 - XVIII - Escola Regional de Alto Desempenho**. Porto Alegre - RS: ERAD, 2018.

- ARGOU, A.; DILLI, R.; REISER, R.; YAMIN, A. Lógica Fuzzy na Definição da Importância dos Atributos de QoS na Seleção de Recursos da IoT. In: **WEIT 2017 - IV Workshop-Escola de Informática Teórica**. Santa Maria - RS: WEIT, 2017.

- DE AZEVEDO, M. M.; DILLI, R.; DAVET, P. T.; YAMIN, A. C. Descoberta de Recursos no Middleware EXEHDA para o Cenário da IoT. In: **ERRC 2016 - XIV Escola Regional de Redes de Computadores**. Porto Alegre - RS: ERRC, 2016.

- DAVET, P.; DILLI, R.; XAVIER, L.; SOUZA, R.; WARKEN, N.; CARDOZO, A.; MACHADO, N.; PARREIRA, W.; YAMIN, A. Uma Abordagem Ciente de Contexto na IoT Aplicada à Pesquisa Agropecuária. In: **SBrT 2016 - XXXIV Simpósio Brasileiro de Telecomunicações**. Santarém - PA: SBrT, 2016.

6.3 Trabalhos Futuros

Dentre as diversas alternativas para continuidade da pesquisa desenvolvida nesta Tese, destacamos as frentes de trabalho a seguir:

- modelar alternativas para incluir o *feedback* do cliente no processo de classificação dos recursos utilizado pelo EXEHDA-RR;

- avaliar alternativas para geração de regras para o processamento Fuzzy tendo por base históricos dos atributos de QoS dos recursos, ficando o especialista somente responsável por especificar regras adicionais para contemplar as preferências do cliente;

- empregar medidas de consenso visando a participação de vários especialistas na definição das funções de pertinência consideradas no EXEHDA-RR, e;

- conceber novas métricas para avaliação da acurácia obtida na classificação de recursos pelo EXEHDA-RR.

REFERÊNCIAS

AGGARWAL, C. C. (Ed.). **The Internet of things: a survey from the data-centric perspective**. 383–428p.

AL-MASRI, E.; MAHMOUD, Q. H. QoS-based discovery and ranking of Web services. **Proceedings - International Conference on Computer Communications and Networks, ICCCN**, Honolulu, HI, USA, p.529–534, 2007.

ALABOOL, H. M.; MAHMOOD, A. K. Trust-Based Service Selection in Public Cloud Computing Using Fuzzy Modified VIKOR Method. **Australian Journal of Basic and Applied Sciences**, v.7, n.9, p.211–220, 2013.

ALADI, J. H.; WAGNER, C.; GARIBALDI, J. M. Type-1 or interval type-2 fuzzy logic systems - On the relationship of the amount of uncertainty and FOU size. **IEEE International Conference on Fuzzy Systems**, Beijing, p.2360–2367, 2014.

ALMULLA, M.; YAHYAOUI, H.; AL-MATORI, K. A new fuzzy hybrid technique for ranking real world Web services. **Knowledge-Based Systems**, v.77, p.1–15, 2015.

ANDREA, Z. et al. **Internet of Things for Smart Cities**. n.[29]. (vol. 1, pp. 22-32). Disponível em: <http://eprints.networks.imdea.org/740/1/06740844.pdf>.

ASHTON, K. **That 'Internet of Things' Thing - RFID Journal.pdf**. 1p. Disponível em: <http://rfidjournal.com/articles/view?4986>.

ATZORI, L.; IERA, A.; MORABITO, G. **The Internet of Things: A survey**. 2787–2805p. v.54, n.15. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1389128610001568>.

ATZORI, L.; LERA, A.; MORABITO, G. The internet of things: A survey. , p.43–50, 2018.

AZEVEDO, M. M. d. **Descoberta de Recursos para o Middleware EXEHDA na Perspectiva da Internet das Coisas**. 2017. 79p. Dissertação (Mestrado em Ciência da Computação) — Universidade Católica de Pelotas.

BARNAGHI, P.; SHETH, A. On Searching the Internet of Things : Requirements and Challenges. **IEEE Intelligent Systems**, v.31, n.6, p.71–75, 2016.

BARROS, L.; BASSANEZI, R. **Tópicos de lógica fuzzy e biomatemática**. 354p.

BHADDURGATTE, R. C.; BP, V. K. A Review: QoS Architecture and Implementations in IoT Environment. **Research & Reviews: Journal of Engineering and Technology**, p.6–12, 2015.

BOUTABA, R. et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. **Journal of Internet Services and Applications**, v.9, n.1, 2018.

BROCK, D. **The Electronic Product Code (EPC) A Naming Scheme for Physical Objects**.

CABRERA, N. **Aplicação da Extensão de Zadeh para Conjuntos Fuzzy Tipo 2 Intervalar**. 2014. 72p. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Uberlândia.

CASTAÑÓN-PUGA, M. et al. Hybrid-Intelligent Mobile Indoor Location Using Wi-Fi Signals - Location Method Using Data Mining Algorithms and Type-2 Fuzzy Logic Systems. **ICEIS 2015 - 17th International Conference on Enterprise Information Systems, Proceedings**, v.1, p.609–615, 2015.

CRISTÓBAL, J. R. S. Multi-criteria decision-making in the selection of a renewable energy project in spain: The Vikor method. **Renewable Energy**, v.36, n.2, p.498–502, 2011.

DAVET, P. T. **EXEHDA-IoT: Uma Contribuição à Arquitetura do Middleware EXEHDA Direcionada à Internet das Coisas**. 2016. 109p. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Pelotas.

DELICATO, F. C.; PIRES, P. F.; BATISTA, T. **Resource Management for Internet of Things**. (SpringerBriefs in Computer Science).

DILLI, R. et al. Uma Abordagem Dinâmica para Descoberta de Recursos na IoT Explorando Processamento Semântico. **Simpósio Brasileiro de Computação Ubíqua - CSBC**, p.908–917, 2017.

DUBOIS, D.; PRADE, H. **Fundamentals of Fuzzy Sets**. (The Handbooks of Fuzzy Sets).

EVANS, D. **A Internet das Coisas Como a próxima evolução da Internet está mudando tudo**. 5–7p.

FACELI, K.; LORENA, A. C.; GAMA, J.; DE CARVALHO, A. **Inteligência artificial: uma abordagem de aprendizado de máquina**.

FLÁVIA C., D.; PAULO F., P.; THAIS, B. **Middleware Solutions for the Internet of Things**. 78p. n.[23]. (2191-5768).

GARCÍA, J. M. **Improving Semantic Web Services Discovery and Ranking**. 2012. 184p. Tese (Doutorado em Ciência da Computação) — University of Seville.

GARCÍA, J. M.; RUIZ, D.; RUIZ-CORTÉS, A.; PAREJO, J. A. QoS-Aware Semantic Service Selection: An Optimization Problem. **2008 IEEE Congress on Services - Part I**, n.July, p.384–388, 2008.

GOHAR, P.; PUROHIT, L. Discovery and prioritization of web services based on fuzzy user preferences for QoS. **IEEE International Conference on Computer Communication and Control, IC4 2015**, 2016.

GOMES, P. et al. A QoC-Aware Discovery Service for the Internet of Things. **Ubiquitous Computing and Ambient Intelligence**, v.7656, p.344–355, 2016.

GOMIDE, F. A. C.; GUDWIN, R. R.; TANSCHEIT, R. **Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações**. n.June.

GRECO, S.; EHRGOTT, M.; FIGUEIRA, J. Multiple Criteria Decision Analysis: State of the Art Surveys. **Multiple Criteria Decision Analysis State of the Art Surveys**, New York, p.1347, 2016.

GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, v.29, n.7, p.1645–1660, 2013.

HOFEREK, O. **Ontological Reasoning with Taxonomies in RDF Database**. 2012. Tese (Doutorado em Ciência da Computação) — Department of Software Engineering - Charles University.

IHS Markit. **The Internet of Things: a movement, not a market**. 9p.

JENNIFER, Y.; BISWANATH, M.; Ghosal Dipak. **Wireless sensor network survey**. n.[20]. Disponível em: <http://cobweb.cs.uga.edu/ laks/ADCS-2013-Fall/YMG-WSN-Survey.pdf>.

KAKARONTZAS, G.; GEROGIANNIS, V. C. Handling Uncertainty During Services Ranking. **International Journal of Services Computing**, v.4, n.August, p.45–57, 2016.

KARNIK, N. N.; MENDEL, J. M. Centroid of a type-2 fuzzy set. **Information Sciences**, v.132, n.1-4, p.195–220, 2 2001.

KHUTADE, P. A.; PHALNIKAR, R. QoS Aware Web Service Selection and Ranking Framework Based on Ontology. **International Journal of Soft Computing and Engineering (IJSCE)**, v.4, n.3, p.77–81, 2014.

KOHAR, R. Optimal Web Service Selection Model using Fuzzy Extended AHP and Weighted Sum Method. **Journal of Digital Information Management**, v.16, n.4, p.157, 2018.

KUMAR, R. R.; MISHRA, S.; KUMAR, C. Prioritizing the solution of cloud service selection using integrated MCDM methods under Fuzzy environment. **Journal of Supercomputing**, v.73, n.11, p.4652–4682, 2017.

LEE, K. et al. QoS for Web Services: Requirements and Possible Approaches. **W3C Working Group Note**, 2003.

LIANG, H. et al. Adaptive computing resource allocation for mobile cloud computing. **International Journal of Distributed Sensor Networks**, v.2013, 2013.

LIMA, H. P. d. Uma Abordagem para Construção de Sistemas Fuzzy baseados em Regras Integrando Conhecimento de Especialistas e Extraído de Dados. , p.137, 2015.

LIU, F. G.; XIAO, F.; LIN, Y. D. Combining experts' opinion with consumers' preference in web service QoS selection. **Proceedings - International Conference on Machine Learning and Cybernetics**, v.4, p.1740–1746, 2013.

LIU, Y.; NGU, A. H.; ZENG, L. Z. QoS computation and policing in dynamic web service selection. **Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters**, p.66–73, 2004.

LOPES, J. B. **Uma Arquitetura para Provimento de Ciência de Situação Direcionada às Aplicações Ubíquas na Infraestrutura da Internet das Coisas**. 2016. 123p. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International Journal of Man-Machine Studies**, 1975.

MENASCÉ, D. A.; RUAN, H.; GOMAA, H. QoS management in service-oriented architectures. **Performance Evaluation**, v.64, n.7-8, p.646–663, 2007.

MENDEL, J. M. A comparison of three approaches for estimating (synthesizing) an interval type-2 fuzzy set model of a linguistic term for computing with words. **Granular Computing**, v.1, n.1, p.59–69, 2016.

MENDEL, J. M. **Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions**. 2.ed. 684p.

MIORANDI, D.; SICARI, S.; DE PELLEGRINI, F.; CHLAMTAC, I. Internet of things: Vision, applications and research challenges. **Ad Hoc Networks**, v.10, n.7, p.1497–1516, 2012.

MOHAMMADSHAHI, Y. A state-of-art survey on TQM applications using MCDM techniques. **Decision Science Letters**, v.2, p.125–134, 2013.

NUNES, L. H. et al. Multi-criteria IoT Resource Discovery: A Comparative Analysis. **Wiley InterScience**, p.1–16, 2016.

PAGANELLI, F.; PARLANTI, D. A DHT-based discovery service for the internet of things. **Journal of Computer Networks and Communications**, v.2012, 2012.

Parsifal. **Parsifal systematic literature review tool**. Disponível em: <https://parsif.al/about/>.

PATIDAR, A.; SUMAN, U. A survey on software architecture evaluation methods. **2nd International Conference on Computing for Sustainable Global Development (INDIACom)**, India, p.967–972, 2015.

PATINIOTAKIS, I.; VERGINADIS, Y.; MENTZAS, G. PuLSaR: preference-based cloud service selection for cloud service brokers. **Journal of Internet Services and Applications**, v.6, n.1, p.1–14, 2015.

PAULO F., P. et al. **Plataformas para a Internet das Coisas.** n.[26]. (Capítulo 3). Disponível em: <http://sbrc2015.ufes.br/wp-content/uploads/Ch3.pdf>.

PERERA, C. **Sensing as a Service for Internet of Things: A Roadmap**.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. **IEEE Communications Surveys and Tutorials**, v.16, n.1, p.414–454, 2014.

PLATENIUS, M. et al. A survey of fuzzy service matching approaches in the context of on-the-fly computing. **CBSE 2013 - Proceedings of the 16th ACM SIGSOFT Symposium on Component Based Software Engineering**, n.April 2017, p.143–152, 2013.

PRIYA, N. H.; CHANDRAMATHI, S. QoS Based Optimal Selection of Web Services Using Fuzzy Logic. **Journal of Emerging Technologies in Web Intelligence 6.3 (2014): 331-339**, v.6, n.3, p.331–339, 2014.

PRUD'HOMMEAUX, E.; SEABORNE, A. **SPARQL Query Language for RDF**. W3C Recommendation. Disponível em: <http://www.w3.org/TR/rdf-sparql-query/>.

RAY, P. P. A survey on Internet of Things architectures. **Journal of King Saud University-Computer and Information Sciences**, v.30, n.3, p.291–319, 2018.

RAZZAQUE, M. A.; MILOJEVIC-JEVRIC, M.; PALADE, A.; CLARKE, S. Middleware for internet of things: A survey. **IEEE Internet of Things Journal**, v.3, n.1, p.70–95, 2016.

REHMAN, Z. U.; HUSSAIN, O. K.; HUSSAIN, F. K. Iaas cloud selection using MCDM methods. **Proceedings - 9th IEEE International Conference on E-Business Engineering, ICEBE 2012**, p.246–251, 2012.

RESEARCH, B. **Global Sensors in Internet of Things (IoT) Devices Market, Analysis & Forecast: 2016 to 2022**.

SAINT-EXUPERY, A. D. Internet of Things: Strategic Research Roadmap. **Internet of Things Strategic Research Roadmap**, p.1–50, 2009.

SALAH, N. B.; SAADI, I. B. Fuzzy AHP for Learning Service Selection in Context-Aware Ubiquitous Learning Systems. **2016 Intl IEEE Conferences on Ubiquitous Intelligence & Comp., Adv. and Trusted Comp., Scalable Comp. and Communications, Cloud and Big Data Comp., Internet of People, and Smart World Congress**, p.171–179, 2016.

SHANMUGAPRIYA, P.; SURESH, R. M. Software Architecture Evaluation Methods - A Survey. **International Journal of Computer Applications**, USA, v.49, n.16, p.19–26, 2012.

SIMON, H. A. Why Should Machines Learn? **Machine Learning: An Artificial Intelligence Approach**, 1983.

SOMA, B.; MUNMUN, S.; SOUVIK, M.; SUBHAJIT, D. **Role of middleware for internet of things: A study**. n.[39]. (vol. 2, no. 3, pp. 94-105). Disponível em: <http://airccse.org/journal/ijcses/papers/0811cses07>.

SOUZA, R. S. d. **Uma arquitetura para Internet das Coisas com suporte ao processamento distribuído do contexto**. 2017. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

SUCHITHRA, M.; RAMAKRISHNAN, M. Non Functional QoS Criterion Based Web Service Ranking. **Proceedings of the International Conference on Soft Computing Systems, ICSCS.**, v.398, 2016.

TANENBAUM, A. S.; BOS, H. **Modern Operating Systems**. 4th.ed.

PRESS, C. (Ed.). **Multiple Attribute Decision Making: Methods and Applications**. 352p. (A Chapman & Hall book).

WAGNER, C. Juzzy - A Java based toolkit for Type-2 Fuzzy Logic. **Proceedings of the 2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems, T2FUZZ 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013**, Singapore, v.1, n.April 2013, p.45–52, 2013.

WANG, h.; YU, C.; WANG, L.; YU, Q. Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences. **IEEE Transactions on Services Computing**, p.1–1, 2015.

WHAIDUZZAMAN, M. et al. Cloud service selection using multicriteria decision analysis. **The Scientific World Journal**, v.2014, p.10, 2014.

WITTEN, I. H.; FRANK, E.; HALL, M. a. **Data Mining: Practical Machine Learning Tools and Techniques**. 3rd.ed. 664p.

XIAO, Y.; WATSON, M. Guidance on Conducting a Systematic Literature Review. **Journal of Planning Education and Research**, v.39, n.1, 2019.

XIN, R. S. et al. Shark: SQL and Rich Analytics at Scale. **Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data**, New York, NY, USA, p.13–24, 2013.

XU, Z.; YAGER, R. R. Some geometric aggregation operators based on intuitionistic fuzzy sets. **International Journal of General Systems**, London, v.35, n.4, p.417–433, 2006.

YAMIN, A. C. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva**. 2004. 195p. Tese (Doutorado em Ciência da Computação) — Instituto de Informática-UFRGS, Porto Alegre-RS.

ZADEH, L. A. Quantitative fuzzy semantics. **Information Sciences**, 1971.

ZADEH, L. A. The concept of a linguistic variable and its application to approximate reasoning-I. **Information Sciences**, 1975.

ANEXOS

ANEXO A MIDDLEWARE EXEHDA

O EXEHDA, citado inicialmente em (YAMIN, 2004), é um *middleware* adaptativo ao contexto e baseado em serviços, com objetivo de criação e gerenciamento de um ambiente computacional, provendo alternativas de execução neste ambiente. Estas aplicações são distribuídas, móveis e adaptativas ao contexto em que seu processamento ocorre, estando disponíveis a partir de qualquer lugar, todo o tempo (LOPES, 2016).

A.1 Arquitetura e Ambiente Computacional

O *middleware* EXEHDA tem como premissa principal a definição de uma arquitetura que capacite aplicações se comunicarem em um ambiente computacional de composição dinâmica, explorando a ciência de contexto na gerência de seus aspectos funcionais e não funcionais. As aplicações alvo do *middleware* EXEHDA são distribuídas, cientes ao contexto e abrangem a mobilidade lógica e física.

Arquitetura de Software

A arquitetura de software do *middleware* EXEHDA é apresentada na Figura 36. Os principais requisitos que o EXEHDA deve atender são: (i) gerenciar tanto aspectos funcionais como não funcionais da aplicação, de modo independente; (ii) dar suporte à adaptação dinâmica de aplicações; (iii) disponibilizar mecanismos para obter e tratar informações de contexto; (iv) empregar informações de contexto na tomada de decisões; (v) decidir as ações adaptativas de forma colaborativa com a aplicação; e (vi) disponibilizar a semântica siga-me, permitindo ao usuário iniciar as aplicações e acessar dados a partir de qualquer lugar.

Ambiente Ubíquo Disponibilizado

O ambiente ubíquo se equivale ao ambiente computacional, visto que recursos e serviços gerenciados pelo EXEHDA tem como premissa cumprir os requisitos propostos em um ambiente computacional. Sistema este, composto por todos dispositivos e

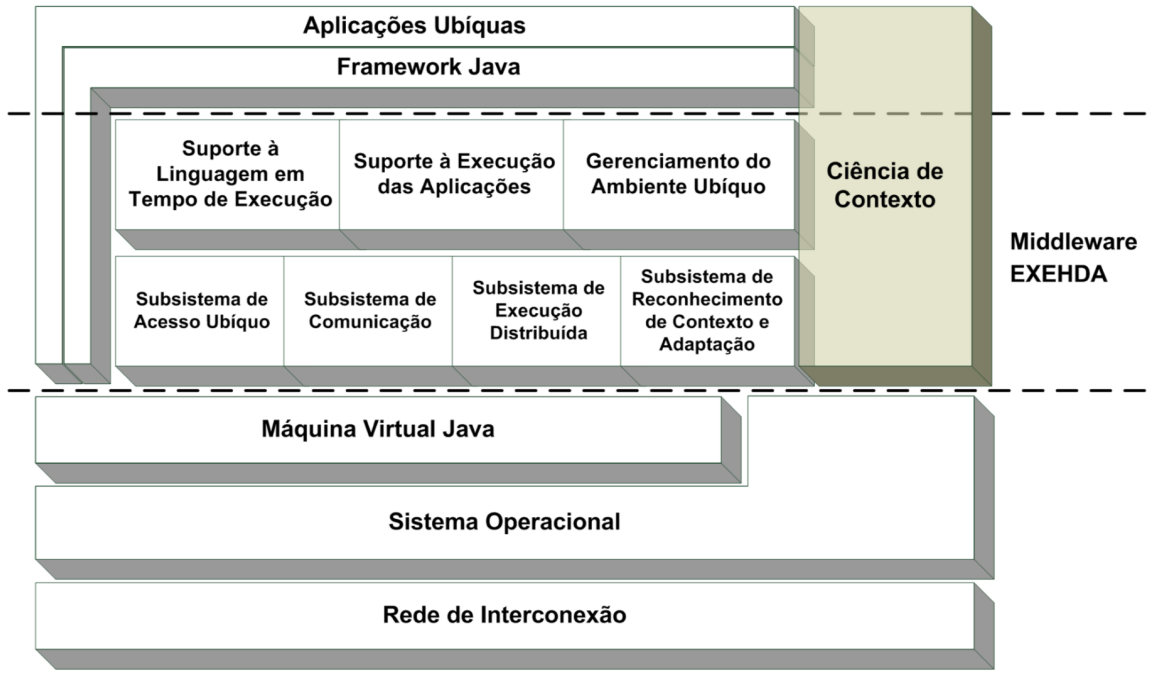

Figura 36 – Arquitetura de Software do *Middleware* EXEHDA
Fonte: LOPES, 2016

usuários relacionados à sua execução no *middleware*.

O EXEHDA propõe um ambiente computacional constituído por células de execução, nas quais os dispositivos computacionais são distribuídos (vide Figura 37). Cada célula é constituída dos seguintes componentes (DAVET, 2016):

- EXEHDAbase - elemento central da célula, sendo responsável por todos serviços básicos e constituindo referência para os demais elementos.
- EXEHDAnodo - corresponde aos dispositivos computacionais responsáveis pela execução das aplicações.
- EXEHDAnodo móvel - um sub-caso do anterior, que corresponde aos dispositivos tipicamente móveis que podem se deslocar entre as células do ambiente ubíquo, como *notebooks*, *tablets* ou *smartphones*.
- EXEHDAborda - responsável por fazer a interoperação entre os serviços do *middleware* e os diversos tipos de *Gateways*.
- EXEHDAgateway - consiste no elemento responsável por setorizar pontos de coleta e/ou atuação distribuídos, disponíveis no meio físico, realizando a interação destes com os outros componentes do *middleware*.

O Servidor de Borda é instanciado em um equipamento do tipo EXEHDAborda e se destina a gerenciar a interação com o meio físico através de *Gateways*, bem como

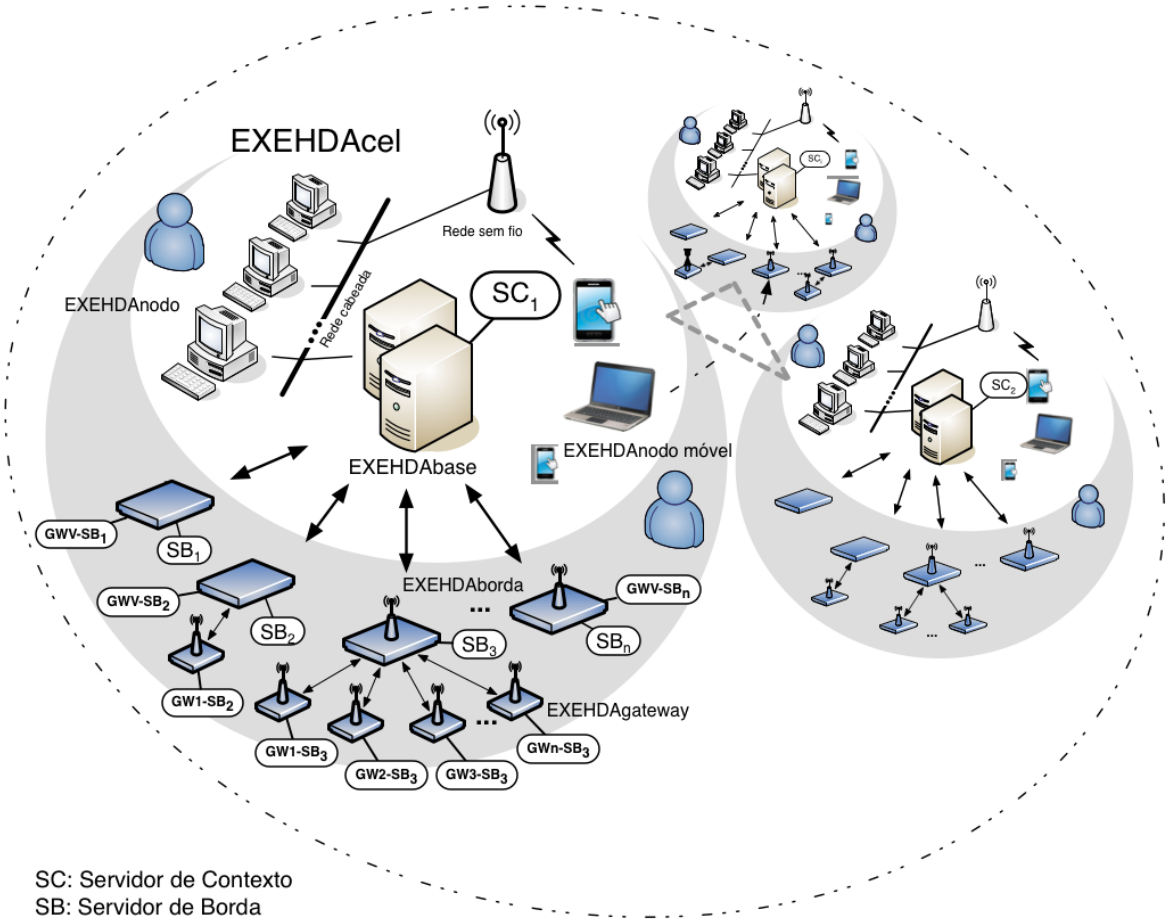

Figura 37 – Organização Celular do Ambiente Ubíquo Gerenciado pelo EXEHDA na IoT
Fonte: DAVET, 2016

a execução de regras de contingência e armazenamento temporário das informações contextuais coletadas, em caso de falha de comunicação. O Servidor de Contexto, por sua vez, é alocado no EXEHDAbase e atua no armazenamento e no processamento das informações contextuais, integrando informações históricas e aquelas provenientes de diferentes Servidores de Borda distribuídos no ambiente ubíquo.

Na proposta do EXEHDA a premissa é que os sensores e/ou atuadores sejam incorporados ao Servidor de Borda através de *Gateways*. Os *Gateways* são utilizados então, para tratar a heterogeneidade, tanto de hardware como de software, inerentes a dispositivos de sensoriamento e/ou atuação, realizando a conversão de protocolos e o gerenciamento dos dispositivos, além de fornecer a estes capacidade de comunicação via Internet.

Com o intuito de garantir a interoperabilidade com as tecnologias de mercado, e também potencializar a distribuição das iniciativas de coleta e/ou atuação, foram pre-

vistos três tipos distintos de *Gateways* (DAVET, 2016):

- Gateways Proprietários (GW-PR) - tem funcionalidades e protocolos heterogêneos variando de acordo com seus fabricantes.
- Gateways Nativos (GW-SB) - suas funcionalidades operam de maneira integrada à arquitetura do EXEHDA, sendo instanciados em equipamentos do tipo EXEHDAgateway.
- Gateway Virtual (GWV-SB) - localizado no mesmo hardware em que executa o Servidor de Borda, constituindo-se em uma virtualização do Gateway Nativo, dispensando quando possível a necessidade de um hardware específico para o *Gateway*.

A existência de dispositivos amplamente heterogêneos em um ambiente computacional, torna necessária a adoção de um núcleo com funcionalidades mínimas, estendidas através de serviços carregados sob demanda. Esta abordagem para padronização de projetos é referenciada na literatura como *microkernel*.

Os ambientes computacionais da atualidade, como os providos pela IoT, são bastante complexos, compreendendo sistemas legados, sistemas fechados provenientes de empresas e sistemas desenvolvidos por grupos de pesquisas interessados. São caracterizados por sua distribuição e heterogeneidade, características comuns em ambientes computacionais. Com base nisso, percebe-se a dificuldade no desenvolvimento de sistemas com sincronização de dados e desenvolvimento de novas funcionalidades que supram necessidades requisitadas por serviços na perspectiva da Computação Ubíqua.

A.2 Subsistemas do EXEHDA

O middleware EXEHDA é formado por quatro subsistemas (LOPES, 2016): Subsistema de Execução Distribuída, Subsistema de Comunicação, Subsistema de Acesso Ubíquo e Subsistema de Reconhecimento de Contexto e Adaptação. Os subsistemas do EXEHDA são descritos nas próximas seções.

Subsistema de Execução Distribuída

O Subsistema de Execução Distribuída é responsável pelo suporte ao processamento distribuído no EXEHDA. No intuito de promover uma execução efetivamente ubíqua, este subsistema interage com outros subsistemas do EXEHDA. Em específico, interage com o Subsistema de Reconhecimento de Contexto e Adaptação, de forma a prover comportamento distribuído e adaptativo às aplicações da Computação Ubíqua. Este subsistema é constituído pelos seguintes componentes:

- Executor - serviço que acumula as funções de disparo de aplicações, e de criação e migração de seus objetos. Na implementação destas funções é empregada a instalação de código sob demanda. Para tal, interage com os serviços CIB e BDA, que podem ser vistos no itens seguintes.

- CIB (Cell Information Base) - serviço que implementa a base de informações da célula. Sua principal funcionalidade está relacionada à manutenção da infraestrutura distribuída que forma o ambiente ubíquo.

- OXManager - a abstração OX (Objeto eXehda), provida pelo middleware às aplicações, consiste em uma instância de objeto, criada por intermédio do serviço Executor, a qual pode ser associada meta-informação em tempo de execução. No caso da abstração OX, esta meta-informação ocorre na forma de atributos, i.e., pares <nome, valor>, sendo que “nome” é uma cadeia de caracteres ASCII que descreve o atributo, podendo “valor” ser um conteúdo genérico, inclusive de natureza binária. A gerência e manutenção da meta-informação associada a um OX é atribuição do serviço OXManager. Este serviço confere às operações de consulta e atualização dos atributos do OX o necessário caráter ubíquo, permitindo que estes sejam acessados a partir de qualquer nodo do ambiente ubíquo.

- Discoverer - o serviço de descoberta de recursos é responsável pela localização de recursos especializados no ambiente ubíquo a partir de especificações abstratas dos mesmos. As especificações caracterizam o recurso a ser descoberto por meio de atributos e seus respectivos valores.

- ResourceBroker - o controle da alocação de recursos às aplicações no EXEHDA é desempenhado pelo serviço ResourceBroker, o qual atende tanto requisições originárias da própria EXEHDAcel quanto oriundas de outras células do ambiente ubíquo.

- Gateway - serviço que faz a intermediação das comunicações entre os nodos externos à célula e os recursos internos a ela. Da sua ação integrada com o ResourceBroker decorre o controle de acesso aos recursos de uma EXEHDAcel.

- StdStreams - serviço que provê o suporte ao redirecionamento dos streams padrões de entrada, saída e erro. Sua funcionalidade ocorre por aplicação, sem a necessidade de modificação no código da mesma. Para isto, define que cada aplicação em execução em um determinado EXEHDAnodo possui um componente individualizado, o qual agrupa os três streams padrões.

- Logger - provê a funcionalidade de registro de rastro de execução (logging). Esta funcionalidade pode ser empregada para registro de operações importantes e/ou

críticas realizadas, facilitando a identificação de situações de intrusão, ou de uso indevido do sistema. Dynamic Configurator - DC: serviço que tem como objetivo realizar a configuração do perfil de execução do *middleware* em um determinado EXEHDAnodo de forma automatizada.

Subsistema de Comunicação

A natureza da mobilidade do hardware e, na maioria das vezes, também a do software, não garante a interação contínua entre os componentes da aplicação distribuída. As desconexões são comuns, não somente devido à existência de alguns links sem fio, mas sobretudo como uma estratégia para economia de energia nos dispositivos móveis. O subsistema de comunicação do EXEHDA disponibiliza mecanismos que atendem estes aspectos da Computação Ubíqua. Integram este subsistema os seguintes serviços:

- Dispatcher - serviço que disponibiliza o modelo de comunicação mais elementar do EXEHDA, ou seja, troca de mensagens ponto-a-ponto com garantia de entrega e ordenamento das mensagens, o qual é especializado para operação no ambiente ubíquo. As mensagens trafegam entre instâncias do Dispatcher localizadas em nodos diferentes através de estruturas denominadas canais. Nesse sentido, quando de sua inicialização, o Dispatcher atualiza a informação do EXEHDAnodo na CIB (*Cell Information Base*), em específico o atributo contactAddress, provendo uma lista de protocolos e endereços que podem ser utilizados para alcançar aquele EXEHDAnodo.

- WORB - serviço que tem o objetivo de simplificar a construção de serviços distribuídos, permitindo que os programadores focalizem esforços no refinamento da semântica distribuída associada ao serviço em desenvolvimento, abstraindo aspectos de baixo nível relativos ao tratamento das comunicações em rede. Para tanto, oferece um modelo de comunicação baseado em invocações remotas de método, similar ao RMI, porém sem exigir a manutenção da conexão durante toda a execução da chamada remota.

- CCManager - considerando a premissa da mobilidade lógica dos componentes que constituem as aplicações ubíquas, o suporte a um mecanismo de comunicação com característica de desacoplamento temporal e espacial é particularmente oportuno, à medida que este simplifica a construção de tais aplicações. Este serviço vem atender a esta demanda, disponibilizando um mecanismo baseado na abstração espaço de tuplas, o qual prescinde da coexistência temporal de emissor e receptor. Outro aspecto oportuno desta abstração é a facilidade

com que podem ser implementados outros padrões de comunicação, além do ponto-a-ponto.

Subsistema de Acesso Ubíquo

A premissa de acesso em qualquer lugar, todo o tempo, a dados e código da Computação Ubíqua, requer um suporte do *middleware*. Os serviços que compõem este subsistema no EXEHDA são:

- BDA (Base de Dados das Aplicações) - o serviço BDA contempla métodos para a recuperação do código integral de uma aplicação ou de componentes específicos e suas dependências. Ainda, este serviço inclui métodos para a gerência das aplicações instaladas. O padrão de utilização concebido para o serviço BDA define que as requisições geradas nos EXEHDAnodos de uma determinada célula são sempre direcionadas à instância celular do serviço BDA da mesma EXEHDAcel onde foi gerada a requisição. Esta, se necessário, realiza o acesso a instâncias remotas (BDAs de outras células) para satisfazer uma dada requisição.

- AVU (Ambiente Virtual do Usuário) - a premissa siga-me reflete-se não só nas aplicações que um usuário atualmente executa, mas no seu ambiente computacional como um todo. Este engloba, além das aplicações em execução, as informações de personalização das aplicações definidas pelo usuário, o conjunto de aplicações instaladas, como também seus arquivos privados. É atribuição do serviço AVU a manutenção do acesso ubíquo a este ambiente virtual, da forma mais eficiente possível.

- SessionManager - serviço responsável pela gerência da sessão de trabalho do usuário, sendo definida pelo conjunto de aplicações correntemente em execução para aquele usuário. A informação que descreve o estado da sessão de trabalho é armazenada no AVU, estando portanto disponível de forma ubíqua.

- Gatekeeper - serviço responsável por intermediar acessos entre as entidades externas ao EXEHDA e os serviços do *middleware*, conduzindo os procedimentos de autenticação necessários.

Subsistema de Reconhecimento de Contexto e Adaptação

Este subsistema inclui serviços que tratam desde a extração da informação bruta sobre as características dinâmicas e estáticas dos recursos que compõem o ambiente ubíquo, passando pela identificação em alto nível dos elementos de contexto,

até o disparo das ações de adaptação em reação a modificações no estado de tais elementos de contexto. Integram este subsistema os seguintes serviços:

- Collector - responsável pela extração da informação bruta (diretamente dos recursos envolvidos) que, posteriormente refinada, dará origem aos elementos de contexto. Para isto, o serviço Collector aglutina a informação oriunda de vários componentes monitores e as repassa aos consumidores registrados. Um componente de monitoramento gerencia um conjunto de sensores parametrizáveis.

- Deflector - disponibiliza abstração de canais de multicast para uso na disseminação das informações monitoradas. A presença do serviço Deflector é decorrência da busca de escalabilidade para a arquitetura de monitoração do EXEHDA.

- ContextManager - responsável pelo tratamento da informação bruta produzida pela monitoração para produção de informações abstratas referentes aos elementos de contexto (valores contextualizados). É recebida como parâmetro uma descrição (XML) de como o dado referente àquele elemento de contexto deve ser produzido a partir da informação proveniente da monitoração.

- AdaptEngine - responsável pelo controle das adaptações de cunho funcional, este serviço provê facilidades para definição e gerência de comportamentos adaptativos por parte das aplicações. Deste modo, libera o programador de gerenciar os aspectos de mais baixo nível envolvidos na definição e liberação dos elementos de contexto junto ao ContextManager. Outra função do serviço AdaptEngine é prover um mecanismo de carga de código contextualizado. Desta forma, com base na informação do estado do elemento de contexto fornecido pelo serviço ContextManager, seleciona e carrega o código correspondente dentre alternativas de código definidas na política de adaptação funcional vigente. Este serviço disponibiliza ainda métodos para a ativação de ações quando determinado estado de um elemento de contexto tornar-se ativo.

- Scheduler - serviço central na gerência das adaptações de cunho não funcional no EXEHDA, isto é, que não implicam alteração de código. Nesse sentido, o Scheduler emprega a informação de monitoração, obtida junto ao serviço Collector, para orientar operações de mapeamento. Essas operações decorrem de instanciações remotas ou migrações realizadas pelo serviço Executor, ou quando de chamadas de re-escalonamento, originadas do estado atual de um recurso não satisfazer mais as necessidades de um objeto anteriormente a ele alocado.