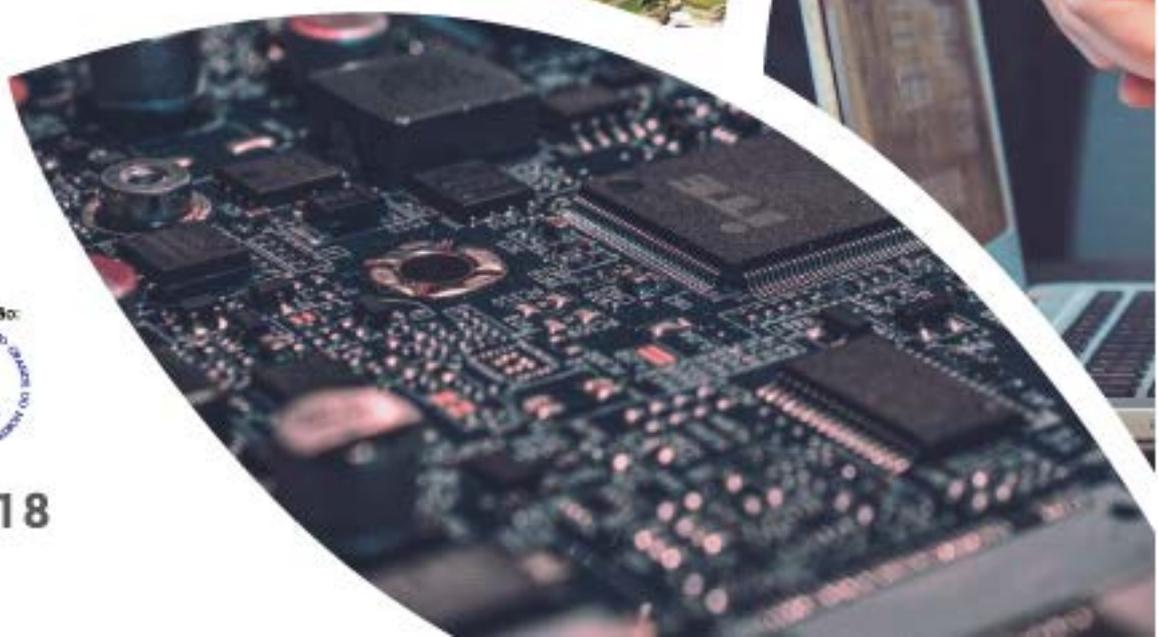


# anais 2018

XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO  
10º SBCUP – SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO UBIQUA E PERSASIVA  
CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018  
#COMPUTAÇÃOESUSTENTABILIDADE



NATAL, 2018

# cnais 2018

XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO  
CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018  
#COMPUTAÇÃOESUSTENTABILIDADE



## **Coordenador Geral**

Francisco Dantas de Medeiros Neto (UERN)

## **Comissão Organizadora**

Bartira Paraguaçu Falcão Dantas Rocha (UERN)

Camila Araújo Sena (UERN)

Everton Ranielly de Sousa Cavalcante (UFRN)

Felipe Torres Leite (UFERSA)

Ilana Albuquerque (UERN)

Isaac de Lima Oliveira Filho (UERN)

Priscila Nogueira Krüger (UERN)

## **Realização**

Sociedade Brasileira de Computação

## **Organização**

Universidade do Estado do Rio Grande do Norte

# **CSBC 2018**

## **XXXVIII Congresso da**

### **Sociedade Brasileira de Computação**

#### **Apresentação**

Estes anais registram os trabalhos apresentados durante o XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018), realizado em Natal-RN, de 22 a 26 de julho 2018. O evento teve como tema central a Computação e Sustentabilidade, pois se compreende que o avanço da computação e as questões ambientais devem caminhar lado-a-lado, tendo em vista que as técnicas computacionais necessitam ser usadas para possibilitar o desenvolvimento sustentável, e, desse modo, equilibrar as necessidades ambientais, econômicas e sociais.

Organizar o maior evento acadêmico de Computação da América Latina foi um privilégio e um desafio. Foi enriquecedor promover e incentivar a troca de experiências entre estudantes, professores, profissionais, pesquisadores e entusiastas da área de Computação e Informática de todo o Brasil. Ao mesmo foi desafiador termos que lidar, principalmente, com às dificuldades impostas pelo momento de crise que o nosso Brasil vem enfrentando. Uma crise que afeta diretamente nossas pesquisas e, conseqüentemente, o desenvolvimento e inovação do nosso amado Brasil.

Por meio de seus 25 eventos, o CSBC 2018 apresentou mais de 300 trabalhos, várias palestras e mesas-redondas. O Congresso ainda abrigou diversas reuniões, que incluem a reunião do Fórum de Pós-Graduação, a reunião do CNPq/CAPES, a reunião dos Secretários Regionais SBC, a reunião das Comissões Especiais e a reunião do Fórum IFIP/SBC.

O sucesso do CSBC 2018 só foi possível devido à dedicação e entusiasmo de muitas pessoas. Gostaríamos de agradecer aos coordenadores dos 25 eventos e aos autores pelo envio de seus trabalhos. Além disso, gostaríamos de expressar nossa gratidão ao Comitê Organizador, por sua grande ajuda em dar forma ao evento; e, em especial, à equipe da Sociedade Brasileira de Computação (SBC), por todo apoio.

Por fim, reconhecemos a importância do apoio financeiro da CAPES, do CNPq, do CGI.br, do Governo do Estado do Rio Grande do Norte, da Prefeitura Municipal do Natal, da Prefeitura Municipal de Parnamirim, da CABO Telecom, da ESIG Software e Consultoria, da DynaVideo e do SENAI.

Natal (RN), 26 de julho de 2018.

**Chico Dantas (UERN)**  
Coordenador Geral do CSBC 2018

**Anais do CSBC 2018**

**10º SBCUP – SIMPÓSIO BRASILEIRO DE  
COMPUTAÇÃO UBÍQUA E PERVERSIVA**

## **Steering Committee**

Adenauer Yamin (UCPEL/UFPEL)  
André Aquino (UFAL)  
Cristiano Costa (UNISINOS)  
Fabio Costa (UFG)  
Hyggo Almeida (UFCEG)  
Iara Augustin (UFES)

## **Coordenador Geral do evento**

Rodrigo da Rosa Righi (UNISINOS)

## **Coordenadores do Comitê de Programa**

Ana Marilza Pernas Fleischmann (UFPEL)  
Frederico Araújo da Silva Lopes (UFRN)

## **Coordenador Local do evento**

Felipe de Oliveira (UERN)

## **Comitê de Programa**

Adenauer Yamin (UCPel and UFPel)  
Andre Aquino (UFAL)  
Angelo Perkusich (UFCEG)  
Antonio Alfredo Ferreira Loureiro (UFMG)  
Atslands Rocha (UFC)  
Augusto Neto (UFRN)  
Bruno Silvestre (UFG)  
Carlos Ferraz (UFPE)  
Carlos Kamienski (UFABC)  
Carlos Mauricio Figueiredo (UFAM)  
Claudio Geyer (UFRGS)  
Cristiano Costa (Unisinos)  
Cristiano Silva (UFSJ)  
Daniel Fernandes Macedo (UFMG)  
Daniel Guidoni (UFSJ)  
Eduardo Cerqueira (UFPR)  
Eduardo Nakamura (UFAM)  
Fabio Costa (UFG)  
Felipe Domingos da Cunha (PUC Minas)  
Flavia Delicato (UFRJ)  
Francisco José Silva (UFMA)  
Giovani Rubert Librelotto (UFES)  
Heitor Ramos (UFAL)  
Humberto Marques (PUC Minas)  
Hyggo Almeida (UFCEG)

Iwens Sene Jr (UFG)  
Jó Ueyama (USP)  
João Carlos Lima (UFSM)  
João Ladislau Lopes (IFSul)  
Joel Rodrigues (Inatel)  
Jorge Barbosa (Unisinos)  
Jorge Sa Silva (UC)  
Leandro de Sales (UFAL)  
Leandro Silva (UFAL)  
Leandro Villas (Unicamp)  
Luiz Fernando Bittencourt (Unicamp)  
Luiz Henrique Correia (UFLA)  
Mario Dantas (UFJF)  
Michele Nogueira (UFPR)  
Patrícia Tedesco (UFPE)  
Patricia Dockhorn Costa (UFES)  
Rodrigo de Souza (IFSul)  
Rodrigo Righi (Unisinos)  
Roseclea Medina (UFSM)  
Thais Vasconcelos Batista (UFRN)  
Thiago Henrique Silva (UTFPR)  
Vaninha Vieira (UFBA)

## SUMÁRIO

|  |    |
|--|----|
| <b>A cardiac arrhythmia monitoring platform based on feature selection and classification methods</b>  | 9  |
| Anderson P. N. Silva (UFRN), Gibeon S. Aquino-Júnior (UFRN), Joao C. Xavier-Júnior (UFRN), Cephas A. S. Barreto (UFRN)   |    |
| <b>ASAPe: uma Arquitetura para Sensoriamento Preditivo em Computação Vestível</b>  | 19 |
| Michelle G. Cacaís (UFC), Leonardo O. Moreira (UFC), Wesley L. Caldas (UFC), Danielo G. Gomes (UFC)  |    |
| <b>Avaliação de técnicas de inteligência computacional para identificação de atividades de vida diária</b>   | 29 |
| Wylken S. Machado (UFAL), Pedro H. Barros (UFAL), Eliana S. Almeida (UFAL), Andre L. L. Aquino (UFAL)  |    |
| <b>Avaliando o Tardigrade: um framework para o desenvolvimento de jogos de cartas ubíquos</b>  | 39 |
| Willian Almeida Rodrigues (UFC), Cecília Costa (UFC), Nécio de Lima Veras (UFC), Vitor Cortez (UFC), Ricardo Andrade (UFC), Fernando Trinta (UFC), Glaudiney M. Mendonça Junior (UFC), Windson Viana (UFC) |    |
| <b>Caracterização de topologia de Redes Veiculares baseada em Teoria da Informação</b>   | 49 |
| Christopher G. S. Freitas (UFAL), Heitor S. Ramos (UFAL), Raquel S. Cabral (UFAL), Osvaldo A. Rosso (UFAL), André L. L. Aquino (UFAL)  |    |
| <b>Construção do Estado da Arte em Estratégias Híbridas para Raciocínio de Contexto: uma Abordagem explorando Revisão Sistemática da Literatura</b>  | 59 |
| Roger S. Machado (UFPe), Ricardo B. Almeida (UFPe), João Ladislau Lopes (IFSul), Ana Marilza Pernas (UFPe), Adenauer C. Yamin (UFPe)   |    |
| <b>DETAB: Modelo Colaborativo De Detecção E Alerta De Buracos Em Vias Pavimentadas</b>   | 69 |
| Thiago Roberto Lima Lopes (UNISINOS), Rodrigo da Rosa Righi (UNISINOS), Lucas Pfeiffer Salomão Dias (UNISINOS), Cristiano André da Costa (UNISINOS)  |    |
| <b>EXEHDA-RR: Uma proposta para tratar incertezas e otimizar o processo de classificação de recursos na IoT</b>  | 79 |
| Renato Dilli (UFPe), Amanda Argou (UFPe), Ana Pernas (UFPe), Renata Reiser (UFPe), Adenauer Yamin (UFPe)   |    |
| <b>EXEHDA-SO: Uma Abordagem Ontológica para Ciência de Situação Aplicada a Segurança da Informação</b>   | 89 |
| Diórgenes Y. L. da Rosa (UFPe), Roger S. Machado (UFPe), Ricardo B. Almeida (UFPe), Adenauer C. Yamin (UFPe), Ana Marilza Pernas (UFPe)  |    |

|   |     |
|---|-----|
| <b>Framework de Suporte ao Desenvolvimento e Evolução de Aplicações Auto-Adaptativas em IoT</b>                                     | 99  |
| Belmondo R. A. Junior <sup>1</sup> (UFC), Tales P. Nogueira (UFC), Marcio E. F. Maia (UFC), Rossana M. C. Andrade (UFC)             |     |
| <b>IMAM – Uma ferramenta para monitoramento inteligente de sistemas e dispositivos em infraestruturas críticas de IoT</b>           | 109 |
| David Coelho dos Santos (UFRN), João Carlos Xavier Júnior (UFRN), Gibeon Soares de Aquino Junior (UFRN)                             |     |
| <b>Integrando as Plataformas FIWARE e KNoT para o Desenvolvimento de Aplicações de Internet das Coisas</b>                          | 119 |
| César Perdigão Batista (UFRN), Pedro Victor Silva (UFRN), Thais Batista (UFRN), Everton Cavalcante (UFRN)                           |     |
| <b>Modelo Baseado em Rede de Sensores Sem Fio Para Predição de Disfunções em Cultivos Agrícolas</b>                                 | 129 |
| Uélison Jean L Santos (UNISINOS), Rodrigo da Rosa Righi (UNISINOS), Cristiano André da Costa (UNISINOS)                             |     |
| <b>Predição de dados de sensoriamento visando eficiência energética de redes de sensores sem fio</b>                                | 139 |
| Charles M. P. Almeida (UFAL), Geymerson S. Ramos (UFAL), Andre L. L. Aquino (UFAL)  |     |
| <b>Seleção de Nó Raiz Ótimo para Sincronização Eficiente de Relógios em Redes de Sensores Sem Fio</b>                               | 149 |
| Tiago C. de S. Xavier (COPPE-UFRJ), Claudio L. Amorim (COPPE-UFRJ), Flavia C. Delicato (COPPE-UFRJ)                                 |     |
| <b>SININHO: um serviço de mapeamento espaço-temporal para melhoria da tomada de decisão na gestão em saúde</b>                      | 159 |
| Lindemberg Nascimento (IFCE), Renato Alves (IFCE), Daniel Victor Saraiva (IFCE), Reinaldo Braga (IFCE), Carina Oliveira (IFCE)      |     |
| <b>Uma Abordagem não Intrusiva para Reconhecimento de Atividades em Casas Inteligentes Explorando Processamento Semântico</b>       | 169 |
| Eduardo Abreu (UFPe), Gerson Cavalheiro (UFPe), Ana Pernas (UFPe), Adenauer Yamin (UFPe), João Lopes (IFSul), Cláudio Geyer (UFRGS) |     |
| <b>Uma Máquina de Estados para Especificação de Códigos de Simulação para Redes de Sensores sem Fio Urbanas</b>                     | 179 |
| Marcos Carrero (UFPR, FAE), Katriny Zamproni (UFPR), Martin A. Musicante (UFRN), Aldri Santos (UFPR), Carmem Hara (UFPR)            |     |

# A cardiac arrhythmia monitoring platform based on feature selection and classification methods

Anderson P. N. Silva<sup>1</sup>, Gibeon S. Aquino-Júnior<sup>1</sup>,  
João C. Xavier-Júnior<sup>2</sup>, Cephax A. S. Barreto<sup>2</sup>

<sup>1</sup>Informatics and Applied Mathematics Department  
Federal University of Rio Grande do Norte  
Natal, RN, Brazil

<sup>2</sup>Digital Metrolopolis Institute  
Federal University of Rio Grande do Norte  
Natal, RN, Brazil

{anderson, gibeon}@dimap.ufrn.br, jcxavier@imd.ufrn.br, cephasax@gmail.com

**Abstract.** *Heart arrhythmia, also known as irregular heartbeat, affects millions of people around the world. One of the ways to detect this cardiac dysrhythmia is by performing an electrocardiogram (ECG) exam which records the electrical activity of the heart. However, this type of exam is always interpreted by a doctor. In order to provide an alternative in heart arrhythmia diagnosis, this paper aims at developing a platform based on Internet of Things infrastructure capable of automatically monitoring and identifying cardiac arrhythmia based on feature selection and classification methods.*

## 1. Introduction

Considering estimations made by the United Nations (UN) that the Earth will have just over 9 billion inhabitants by 2050, concerns with diseases have become an increasingly alarming topic worldwide. According to the Brazilian Institute of Supplementary Health Studies, in a study carried out with the assistance of the Medical School of the Federal University of Minas Gerais, approximately 829 Brazilians die daily in public and private hospitals due to heart malfunctioning.

To reduce the number of deaths caused by diseases, health professionals have invested resources in tools which can perform diagnoses, whether preventive or emergency ones. This anticipation in the discovery of some diseases such as heart dysfunctions can determine medical treatments, and even prevent patients' sudden deaths. On top of that, special type of algorithms have been used consistently in order to provide immediate benefits to disciplines with reproducible or standardized processes.

Machine learning techniques have been used since the beginning in the healthcare field to identify disease patterns [Ilayaraja and Meyyappan 2013]. Information Technology companies have already begun to develop Machine Learning applications that can remotely track the employees' health or monitor the health of older people. Moreover, many studies have also been focused on monitoring different diseases, such as: high blood pressure, diabetes, and others.

Differently from other works, this work has the main purpose of developing a cardiac arrhythmia monitoring platform based on Internet of Things (IoT) infrastruc-

ture, which enables monitoring, identifying and notifying health professionals, patients and family members in real-time based on feature selection techniques and classification methods. Moreover, two well-known public arrhythmia datasets were used for training and testing the platform, and also three different feature selection methods were used to better understand the particularities (more relevant attributes) of such heart malfunctioning. On top of that, was performed empirical analyses with both base classifiers (Decision tree, Naive Bayes, Multilayer Perceptron, k-Nearest Neighbor and Support Vector Machine), and ensembles of classifiers (AdaBoost and Random Forest). In addition, the results of the experiments indicate that the use of ensembles of classifiers, and feature selection method improves system response time, and helps to better understand the most relevant attributes in the context of Heart arrhythmia.

## 2. Background

According to [Mitchell 1997], machine learning is a subarea of Artificial Intelligence responsible for the development of models (hypotheses) generated from data, and that automatically improve with the experience. In this way, machine learning aims at constructing models that can be learned according to samples and past experiences.

As many pattern recognition techniques were originally not designed to cope with large amounts of irrelevant features, combining them with FS techniques has become a necessity in many applications [Guyon and Elisseeff 2003, Liu and Motoda 2012]. The objectives of feature selection are manifold, the most important ones being: (a) to avoid overfitting and improve model performance, i.e. prediction performance in the case of supervised classification and better cluster detection in the case of clustering, (b) to provide faster and more cost-effective models and (c) to gain a deeper insight into the underlying processes that generated the data.

Classification is well known machine learning task. Data classification is the process of creating a prediction model from a learning algorithm. The aim of this model is to predict the value of the class attribute of testing instances. In this paper, we use the following classification methods: k-NN, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Decision tree, Naive Bayes, Random Forest and AdaBoost.

An ensemble of classifiers can be defined as a collection of classifiers responsible for producing the final output of the system when aggregated to a combination method [Kuncheva 2004]. The ensembles lead to greater generalization capacity than when working separately. Several studies have shown that combining the results of different classifiers outperform base classifiers. In the of ensemble of classifiers, three main aspects must be considered [Kuncheva 2004], which are:

- The structure of the ensemble system: it defines how the components are organized, and how they will interact [Kuncheva 2004].
- The components of the ensemble system: it is necessary to define which components will be used to compose the system [Kuncheva 2004]:
- The combination method: this method is used to combine results of the output of the classifiers [Kuncheva 2004]. There is a vast number of combination-based methods reported in the literature.

### 3. Related Works

As the main purpose of this work is to develop a cardiac arrhythmia monitoring platform based on feature selection techniques and classification methods, we will present the works more related to this main goal.

In [Soman and Bobbie 2005], the authors used three machine learning methods that were applied in the 5 tasks of arrhythmia classification, and the most accurate learning methods were evaluated. In [Tsipouras et al. 2005], the authors propose a knowledge-based method for classification of arrhythmic beats and detection of arrhythmic episodes using only the RR interval signal extracted from ECG recordings.

In the work [Oresko et al. 2010] is explored the use of a hierarchical model for the classification of cardiac arrhythmia. In addition, the authors investigate the performance of machine learning techniques for the four-beat classification of cardiac arrhythmia. In this article [Jenny et al. 2014] the authors propose a computer aided ECG diagnostic system for premature ventricular contraction (PVC). The proposed system uses Independent Component Analysis (ICA) to extract resources for cluster k-means and the Fuzzy C-Means classifier (FCM).

In the work [Ahmed and Arafat 2014] is presented the development of a platform based on smartphones for detection of cardiovascular diseases using wearable device, which are able to perform in real time the acquisition of ECG data, extraction of characteristics, and classification of arrhythmias. The author used to classify the machine learning algorithm MLP.

In the paper [Desai et al. 2015], a machine learning approach for computer-aided detection of the five classes of ECG arrhythmia beats is described using Discrete Wavelet Transform (DWT). The paper [Pławiak 2018] presents a methodology that allows the efficient classification of cardiac disorders (17 classes) based on ECG signal analysis and a neural evolutionary system.) To improve the characteristics of the ECG signal, the spectral power density was estimated (using the Welch method and a discrete Fourier transform).

## 4. The Proposed Monitoring Platform

As mentioned, the main aim of this paper is to develop a platform based on Internet of Things infrastructure capable of automatically monitoring and identifying cardiac arrhythmia. We discuss the relevant details in the next subsection.

### 4.1. Platform Requirement

In order to build a platform that allows data collection, storage, patient monitoring, prediction and event notification, it was necessary to define a set of functional and non-functional requirements so that the proposed platform can fulfill its purpose.

#### 4.1.1. Non-Functional Requirements

During the planning and analysis of the proposed platform, it was necessary to identify the non-functional requirements necessary for the best functioning of the platform. From this analysis, we discuss some details below:

- In something of great importance as health, especially if it is of organs of greater importance as the heart, the diagnostic time is of fundamental importance, therefore, it can modify completely the mode of intervention. In this sense, the faster the diagnosis, the lower the risk of complications. So one of the requirements of the platform is to perform the analysis of the information in **real-time**. For example, if a significant variation in cardiac electrical behavior occurs in an Intensive Care Unit patient, the proposed solution should activate an alarm within a few seconds.
- **Data privacy** is an essential factor in a system, and especially when the system is in the health area where there is a lot of intimate and sensitive information because of the discriminatory potential they hold. Therefore, a system of permissions will be created on the platform in which you need the permission of the data type to have access to it and to avoid using it in undue ways.

#### 4.1.2. Functional Requirement

The functional requirements are discuss below:

- **Collect Data:** the platform must be capable of continuous data collection from the monitored patient. All collections will be recorded as they occur, over a certain period of time.
- **Predict information:** the platform can predict information according to the data being collected and stored by the system. The platform will have the ability to learn criteria that can support future decisions, such as discovering new knowledge, finding unknown patterns in the data.
- **Send Alerts:** the platform must have an alert system that will be responsible for notifying end users about the health of the hospitalized patient. From the moment the platform judges that the information detected is critical the system will issue the alert for the service.

#### 4.2. System Abstraction

Figure 1 system abstraction of the intelligent Cardiac Arrhythmia Monitoring Platform. The system will be designed to ensure support for the various data sources found that use the *Health Level 7 (HL7)* protocol in the most varied contexts. It is noted that the general architecture can be divided into three layers: monitoring, *middleware* and services.

The first layer of monitoring is composed of two elements, being: the **sensor** that are the devices used for the monitoring of events in the body, the ECG is the main sensor in the use of this work. The **gateway** will be able to receive the data directly from the sensors and pass it on to the layer above, which is the **middleware**.

The second layer of the architecture is **middleware**, which is responsible for receiving the data from the monitoring layer, processing it, and making it available to the services layer. This layer is composed of one following element: the **intelligence module**. The intelligence module receives the data from the gateway and performs the processing of these using machine learning algorithms for predicting information in real-time flow. After detecting some kind of abnormality in the patient's health, the module will provide notifications information to the service layer.

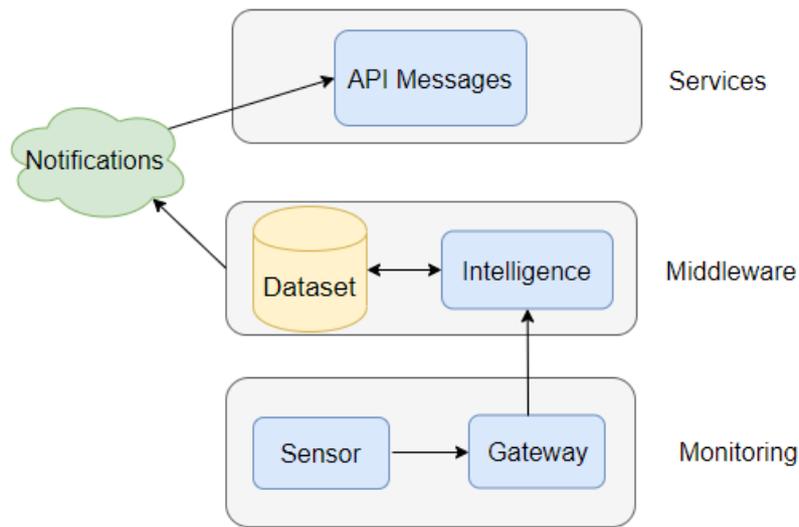


Figure 1. System Abstraction

The third layer of services is responsible for the **services** and programming of interfaces Applications (API). This layer is composed of the **API messages** that will have Representational State Transfer (REST) technology that is responsible for providing information for edge devices such as smart clocks, smartphones and computers.

### 4.3. The functioning of the Platform

The proposed platform consists of 6 steps such as describe the Figure2

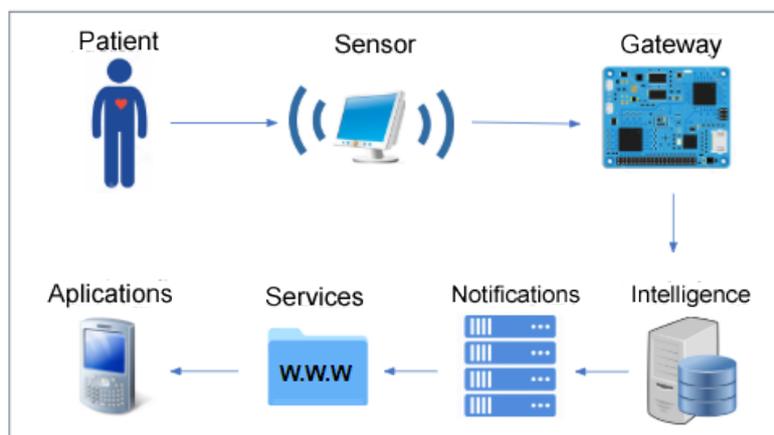


Figure 2. Functioning of the Platform

## 5. Methodology

In this section, important details will be described such as: datasets and experimental configuration.

## 5.1. Datasets

Two well-known datasets were used which are widely used in the literature being MIT-BIH<sup>1</sup> and UCI<sup>2</sup> Dataset.

- MIT-BIH Arrhythmia dataset: the data consist of 48 records, 30 min in length, extracted from 24 hours of ECG acquisition. The signs were acquired from 47 patients between 1975 and 1979 at the Boston Beth Israel Hospital Arrhythmia Laboratory, aged 23-89 years, of whom 22 were women and 25 men. The heart rate was marked and classified manually by specialists in 15 classes on the type of arrhythmia.
- UCI Arrhythmia dataset: this dataset contains 279 attributes, 206 of which are linear valued and the rest are nominal. The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups. Class 01 refers to 'normal' ECG classes 02 to 15 refers to different classes of arrhythmia and class 16 refers to the rest of unclassified ones.

## 5.2. Implementational Issues

In order to evaluate the performance of the classification methods, we created five versions of the datasets, randomizing the original instances. In this way, each classification method were applied to two original datasets and their five versions, performing 12 datasets in total. For simplicity reason, allow us to call this as the first scenario.

From the first scenario (12 datasets), we applied different feature selection techniques (attribute evaluator and search method), such as: CfsSubsetEval with GreedyStepwise, WrapperSubsetEval (J48) with GreedyStepwise, and WrapperSubsetEval (Naive Bayes) with GreedyStepwise. In this way, we obtained three more scenarios.

In this paper, in order to obtain a better estimation of the accuracy rates, we applied 10-fold cross validation, and 4 different percentage splits (90%, 75%, 66% and 50%) for all classification methods.

The Friedman statistical test was applied to evaluate the performance of all classifiers in four different scenarios. It is important to emphasize that the Friedman test is applied directly on the accuracy values of all classifiers. If any significant difference is detected, a posthoc test will be applied.

## 6. Results

In order to validate our proposed platform, we conducted an empirical analysis which will be described in details in the next subsections.

### 6.1. Performance Analysis

Table 1 presents the average accuracy and standard deviation for all seven classifiers. Note that AdaBoost/J48 classifier obtained the best accuracy results in both datasets.

As it can be noticed in Table 2, AdaBoost/J48 classifier was significantly better than the other classifiers.

<sup>1</sup><https://www.physionet.org/physiobank/database/mitdb/>

<sup>2</sup><https://archive.ics.uci.edu/ml/datasets/arrhythmia>

**Table 1. Scenario 1 - Original Datasets**

| Datasets / Classifier | MIT arrhythmia    | UCI arrhythmia    | Average / Standard Deviation |
|-----------------------|-------------------|-------------------|------------------------------|
| Ada/J48               | <b>93,08 1,49</b> | <b>91,53 1,15</b> | <b>89,55 1,56</b>            |
| J48                   | 70,65 4,86        | 64,47 4,11        | 52,88 3,52                   |
| kNN                   | 81,86 3,17        | 78,00 2,63        | 71,22 2,54                   |
| MLP                   | 81,41 2,47        | 68,01 4,20        | 74,71 3,33                   |
| NB                    | 75,79 2,44        | 72,70 4,47        | 74,25 3,46                   |
| RF                    | 91,73 1,79        | 71,12 4,70        | 81,43 3,25                   |
| SVM                   | 75,24 2,14        | 63,04 4,73        | 69,14 3,43                   |

**Table 2. Scenario 1 - Original Datasets**

Nemenyi p-values, with no further adjustment

|            | Ada/J48 | J48     | kNN     | MLP     | NB      | RF      |
|------------|---------|---------|---------|---------|---------|---------|
| <b>J48</b> | 0,00000 |         |         |         |         |         |
| <b>kNN</b> | 0,00000 | 0,00008 |         |         |         |         |
| <b>MLP</b> | 0,00000 | 0,87580 | 0,01376 |         |         |         |
| <b>NB</b>  | 0,00000 | 0,00000 | 0,38330 | 0,00000 |         |         |
| <b>RF</b>  | 0,00480 | 0,62260 | 0,00000 | 0,05167 | 0,00000 |         |
| <b>SVM</b> | 0,00000 | 0,83746 | 0,01825 | 1,00000 | 0,00000 | 0,04037 |

Table 3 presents the accuracy results for all classifiers in scenario 2. Note that the Random Forest classifier obtained the best overall results. However, it outperformed the AdaBoost/J48 only in the UCI arrhythmia dataset.

**Table 3. Scenario 2 - Feature Selection (CfsSubsetEval with GreedyStepwise)**

| Datasets / Classifier | MIT arrhythmia    | UCI arrhythmia    | Average / Standard Deviation |
|-----------------------|-------------------|-------------------|------------------------------|
| Ada/J48               | <b>89,05 1,96</b> | 72,02 4,85        | 80,54 3,40                   |
| J48                   | 87,75 1,26        | 69,29 4,29        | 78,52 2,77                   |
| kNN                   | 88,74 2,11        | 63,25 3,70        | 76,00 2,90                   |
| MLP                   | 71,32 2,96        | 68,09 3,39        | 69,71 3,18                   |
| NB                    | 61,19 2,44        | 69,50 4,47        | 65,35 3,45                   |
| RF                    | 88,39 1,89        | <b>75,13 3,62</b> | <b>81,76 2,75</b>            |
| SVM                   | 61,40 3,52        | 69,21 4,65        | 65,31 4,08                   |

It is important to emphasize that Random Forest classifier was significantly better than the other classifiers. However, it was not the case when compared to AdaBoost/J48 (see Table 4).

Table 5 presents the accuracy results for all classifiers in scenario 3. Again the Random Forest classifier obtained the best overall results. However, it outperformed AdaBoost/J48 only in the UCI arrhythmia dataset. As it can be seen in Table 6, Random Forest classifier was significantly better than kNN, MLP and SVM classifiers.

Table 7 presents the accuracy results for all classifiers in scenario 4. Again the

**Table 4. Scenario 2 - Feature Selection (CfsSubsetEval with GreedyStepwise)**

| Nemenyi p-values, with no further adjustment |         |         |         |         |         |         |
|--|---------|---------|---------|---------|---------|---------|
|  | Ada/J48 | J48     | kNN     | MLP     | NB      | RF      |
| <b>J48</b>                                   | 0,00127 |         |         |         |         |         |
| <b>kNN</b>                                   | 0,00000 | 0,60833 |         |         |         |         |
| <b>MLP</b>                                   | 0,00000 | 0,03556 | 0,82700 |         |         |         |
| <b>NB</b>                                    | 0,00000 | 0,00057 | 0,16946 | 0,92227 |         |         |
| <b>RF</b>                                    | 0,99983 | 0,00027 | 0,00000 | 0,00000 | 0,00000 |         |
| <b>SVM</b>                                   | 0,00000 | 0,00075 | 0,19471 | 0,94054 | 1,00000 | 0,00000 |

**Table 5. Scenario 3 - Feature Selection (WrapperSubsetEval/J48 with GreedyStepwise)**

| Datasets / Classifier | MIT arrhythmia           | UCI arrhythmia           | Average / Standard Deviation |             |
|-----------------------|--------------------------|--------------------------|------------------------------|-------------|
| Ada/J48               | <b>93,03</b> <b>1,56</b> | 68,34 4,25               | 80,68                        | 2,91        |
| J48                   | 91,60 1,12               | 72,35 3,60               | 81,98                        | 2,36        |
| kNN                   | 89,01 1,81               | 61,12 3,88               | 75,07                        | 2,84        |
| MLP                   | 80,83 2,68               | 70,10 3,53               | 75,46                        | 3,11        |
| NB                    | 75,03 1,90               | 66,40 5,29               | 70,72                        | 3,60        |
| RF                    | 91,82 3,74               | <b>72,80</b> <b>3,53</b> | <b>82,31</b>                 | <b>3,63</b> |
| SVM                   | 75,59 3,92               | 60,46 5,37               | 68,02                        | 4,64        |

**Table 6. Scenario 3 - Feature Selection (WrapperSubsetEval/J48 with GreedyStepwise)**

| Nemenyi p-values, with no further adjustment |         |         |         |         |         |         |
|--|---------|---------|---------|---------|---------|---------|
|  | Ada/J48 | J48     | kNN     | MLP     | NB      | RF      |
| <b>J48</b>                                   | 0,99903 |         |         |         |         |         |
| <b>kNN</b>                                   | 0,00000 | 0,00000 |         |         |         |         |
| <b>MLP</b>                                   | 0,01376 | 0,00231 | 0,07351 |         |         |         |
| <b>NB</b>                                    | 0,00000 | 0,00000 | 0,93481 | 0,00195 |         |         |
| <b>RF</b>                                    | 0,67863 | 0,92227 | 0,00000 | 0,00001 | 0,00000 |         |
| <b>SVM</b>                                   | 0,00000 | 0,00000 | 0,01956 | 0,00000 | 0,30841 | 0,00000 |

Random Forest classifier obtained the best overall results. However, it was outperformed by AdaBoost/J48 and Naive Bayes in MIT arrhythmia and UCI arrhythmia datasets.

Again, as it can be seen in Table 8, Random Forest classifier was significantly better than kNN, MLP and SVM. classifiers. Moreover, Random Forest was not significantly better than AdaBoost/J48 in any scenario.

## 7. Conclusion

In this paper, a platform proposal for the cardiac arrhythmia monitoring platform was presented, and also an empirical analysis was conducted in order to evaluate the accuracy performance of seven classification methods (Adaboost, k-NN, Naive Bayes, Decision

**Table 7. Scenario 4 - Feature Selection (WrapperSubsetEval/Naive Bayes with GreedyStepwise)**

| Datasets / Classifier | MIT arrhythmia    | UCI arrhythmia    | Average / Standard Deviation |
|-----------------------|-------------------|-------------------|------------------------------|
| Ada/J48               | <b>93,14 1,67</b> | 66,94 3,00        | 80,04 2,33                   |
| J48                   | 91,23 1,32        | 67,29 3,82        | 79,26 2,57                   |
| kNN                   | 89,06 1,75        | 61,49 4,19        | 75,27 2,97                   |
| MLP                   | 81,41 2,47        | 68,01 4,20        | 74,71 3,33                   |
| NB                    | 75,79 2,44        | <b>72,70 4,47</b> | 74,25 3,46                   |
| RF                    | 91,73 1,79        | 71,12 4,70        | <b>81,43 3,25</b>            |
| SVM                   | 75,24 2,14        | 63,04 4,73        | 69,14 3,43                   |

**Table 8. Scenario 4 - Feature Selection (WrapperSubsetEval/Naive Bayes with GreedyStepwise)**

| Nemenyi p-values, with no further adjustment |         |         |         |         |         |         |
|--|---------|---------|---------|---------|---------|---------|
|  | Ada/J48 | J48     | kNN     | MLP     | NB      | RF      |
| <b>J48</b>                                   | 0,83746 |         |         |         |         |         |
| <b>kNN</b>                                   | 0,00000 | 0,00003 |         |         |         |         |
| <b>MLP</b>                                   | 0,00098 | 0,09191 | 0,34485 |         |         |         |
| <b>NB</b>                                    | 0,06942 | 0,74492 | 0,01478 | 0,88447 |         |         |
| <b>RF</b>                                    | 0,78187 | 0,08230 | 0,00000 | 0,00000 | 0,00039 |         |
| <b>SVM</b>                                   | 0,00000 | 0,00000 | 0,11988 | 0,00005 | 0,00000 | 0,00000 |

Tree, Random Forest, SVM and MLP). Moreover, two well-known cardiac arrhythmia datasets were used in this analysis. Feature selection techniques were also applied to these two datasets.

In general, some significant improvements were observed in some cases after applying feature selection techniques, besides the reduction of the dataset and consequently an improvement in system response time. Among the seven different classification methods, ensembles of classifiers, such as Random Forest and AdaBoost (J48) have outperformed all base classification methods.

The results indicate that the use of feature selection can help us to better understand the most relevant attributes in the datasets, and also to maintain a high degree of accuracy, which shows that machine learning can assist with a good degree of accuracy in the diagnosis of cardiac arrhythmia.

### 7.1. Future Works

- It is planned to carry out new case studies, and with this, it is intended to evaluate all the operating flows and requirements present in the platform.
- Create partnerships with local doctors to determine the creation of a dataset regional of cardiac arrhythmia, thus, contributing to science and well-being of peoples.
- Using committees with different numbers of members and investigate other classifiers bases.

## References

- Ahmed, R. and Arafat, S. (2014). Cardiac arrhythmia classification using hierarchical classification model. In *Computer Science and Information Technology (CSIT), 2014 6th International Conference on*, pages 203–207. IEEE.
- Desai, U., Martis, R. J., Nayak, C. G., Sarika, K., and Seshikala, G. (2015). Machine intelligent diagnosis of ecg for arrhythmia classification using dwt, ica and svm techniques. In *India Conference (INDICON), 2015 Annual IEEE*, pages 1–4. IEEE.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.
- Ilayaraja, M. and Meyyappan, T. (2013). Mining medical data to identify frequent diseases using apriori algorithm. In *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, pages 194–199. IEEE.
- Jenny, N. Z. N., Faust, O., and Yu, W. (2014). Automated classification of normal and premature ventricular contractions in electrocardiogram signals. *Journal of Medical Imaging and Health Informatics*, 4(6):886–892.
- Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.
- Liu, H. and Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media.
- Mitchell, T. M. (1997). *Machine Learning. First edition*.
- Oresko, J. J., Jin, Z., Cheng, J., Huang, S., Sun, Y., Duschl, H., and Cheng, A. C. (2010). A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):734–740.
- Pławiak, P. (2018). Novel methodology of cardiac health recognition based on ecg signals and evolutionary-neural system. *Expert Systems with Applications*, 92:334–349.
- Soman, T. and Bobbie, P. O. (2005). Classification of arrhythmia using machine learning techniques. *WSEAS Transactions on computers*, 4(6):548–552.
- Tsipouras, M. G., Fotiadis, D. I., and Sideris, D. (2005). An arrhythmia classification system based on the rr-interval signal. *Artificial intelligence in medicine*, 33(3):237–250.

# ASAPe: uma Arquitetura para Sensoriamento Preditivo em Computação Vestível

Michelle G. Cacaís<sup>1</sup>, Leonardo O. Moreira<sup>1</sup>, Wesley L. Caldas<sup>1</sup>,  
Danielo G. Gomes<sup>1</sup>

<sup>1</sup>Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

[michellecacaís, weslleylc]@alu.ufc.br, [leoomoreira, danielo]@ufc.br

**Abstract.** *Among the several concepts associated with Internet of Things (IoT), we highlight wearable computing, which involves body sensing to capture data from people who are "wearing" the sensor. This paper presents ASAPe, an architecture for predictive sensing in wearable computing. Techniques and strategies for implementing our proposal are discussed and evaluated in this proposal. The results achieved so far indicate that ASAPe can be very useful both in real-time data collection contexts as well as in massive amounts of data. The main contribution is a new paradigm for working components and prediction in Wearable Computing.*

**Resumo.** *Dentre os vários conceitos associados à Internet das Coisas (IoT), destacamos a computação vestível, a qual envolve sensoriamento corporal para captar dados das pessoas que estiverem "vestindo" o sensor. Este artigo apresenta ASAPe, uma arquitetura para sensoriamento preditivo em computação vestível. Técnicas e estratégias de implementação da nossa proposta são discutidas e avaliadas nesta proposta. Os resultados alcançados até o presente momento indicam que a ASAPe pode ser muito útil tanto em contextos de coleta de dados em tempo real como também em quantidade massiva de dados. A principal contribuição é fornecer um novo paradigma para trabalhar componentes e predição em Computação Vestível.*

## 1. Introdução

Na última década, temos visto uma significativa proliferação de objetos físicos passíveis de conexão à Internet. Com o paradigma da Internet das Coisas (IoT), os chamados objetos inteligentes oferecem informações sensoriadas do seu entorno físico em uma comunicação objeto-a-objeto [Liu and Zhou 2012]. O aspecto vago do termo "Coisas" ao mesmo tempo que dificulta a própria definição do conceito de IoT, dá uma ideia clara de sua heterogeneidade e de seu potencial de aplicação praticamente ilimitado. Dentre as várias possibilidades e vertentes da IoT, destacamos a computação vestível.

Com o advento da computação ubíqua/pervasiva, a noção de computação vestível dos anos 1980-1990 expandiu-se e hoje encontra-se incorporada ao conceito de IoT. A pesquisa e o desenvolvimento em computação vestível têm propiciado soluções mais funcionais e atrativas aos seus usuários em potencial, grande parte delas voltada para as áreas da saúde [Conjeti et al. 2012], [Molina-Markham et al. 2014] [Gope and Hwang 2016],

*fitness*, [Pels et al. 2014] [Rocha et al. 2015], [Salehizadeh et al. 2016], esportes [Nguyen et al. 2015], [Spelmezan 2012], como também para a área social [Kan et al. 2015] [Chen et al. 2015].

A computação vestível também permite ao usuário permanecer conectado e monitorado constantemente através de sua roupa ou de outro acessório sensor acoplado ao corpo. Dependendo do propósito, este recurso pode ser adequado quando o usuário deve ser monitorado para evitar eventos indesejados ou acidentes. Apesar da literatura recente incluir uma variedade de soluções para detecção preventiva em computação vestível ([Chen et al. 2015], [Forkan and Khalil 2016], [Jaimes et al. 2017] [Kang et al. 2015]), não há uma arquitetura ou *framework* específico para análise de dados corporais e predição em Computação Vestível.

Desse modo, apresentamos ASAPe, uma arquitetura para sensoriamento preditivo para computação vestível. O intuito é criar um mecanismo para organizar componentes em infraestruturas de computação vestível, ajudando com a reutilização e padronização. Além disso, o principal foco é agregar uma camada de predição para dar suporte à análise de dados e prevenção de eventos indesejados com base nos dados corporais do usuário.

## 2. Trabalhos Correlatos

Alguns trabalhos na área de Computação Vestível foram desenvolvidos para resolver problemas relacionados à essa temática. Usamos *strings* de busca como: "*wearable computing frameworks*", "*body sensor networks framework*", "*architectures for wearable computing*" e "*predictive sensing in wearable computing*", nas bases de dados IEEE<sup>1</sup>, ACM<sup>2</sup> e ISI<sup>3</sup>, entre 2014 e 2017. Aqui, listamos os que lidam com detecção preditiva em *Wearable Computing, frameworks* que ajudam com a heterogeneidade de componentes e arquiteturas *wearable* projetados para Computação Vestível.

O sensoriamento preditivo é usado em Computação Vestível principalmente para evitar eventos fisiológicos indesejados. Em [Forkan and Khalil 2016], os autores usaram Hidden Markov Model (HMM) a fim de identificar eventos fisiológicos de um paciente. O trabalho visto em [Jaimes et al. 2017] fala sobre o uso de HMM e previsão de suavização exponencial para evitar episódios de estresse. Em [Kan et al. 2015], os autores forneceram um trabalho sobre detecção preditiva para situações de saúde urgentes em tempo real.

Os *frameworks* voltados para Computação Vestível tem como objetivo fornecer abstração da heterogeneidade de infraestruturas *wearable*. O trabalho visto em [Gheith et al. 2016] é sobre o IBM Bluemix, uma nuvem oferecida pela IBM. É uma plataforma para aplicações móveis baseadas na nuvem. Em [Uddin et al. 2015] os autores forneceram um quadro genérico para monitorar continuamente as atividades diárias. Em [Galzarano et al. 2016] criou-se uma ferramenta específica de domínio denominada SPINE 2, para prototipagem rápida de redes sem fio de sensores corporais (WBSN).

As arquiteturas voltadas para Computação Vestível resolvem problemas relacionados à organização dos componentes envolvidos em infraestruturas *wearable*. O traba-

<sup>1</sup><http://ieeexplore.ieee.org/Xplore/home.jsp>

<sup>2</sup><https://dl.acm.org/>

<sup>3</sup><https://pcs.webofknowledge.com/>

lho visto em [Abdali-Mohammadi et al. 2015] é sobre uma arquitetura para a construção de sistemas vestíveis em aplicações médicas para observar situações críticas. Já [Chen et al. 2015] apresenta AIWAC, uma arquitetura para computação vestível, voltada para a interação afetiva. O trabalho [Wang et al. 2016] é relacionado a uma arquitetura para padronizar o desenvolvimento do tecido inteligente adequado para roupas inteligentes e WBSN.

## 2.1. Comparação

A detecção preditiva para sensores corporais tem sido usada em maioria para princípios de saúde. As arquiteturas encontradas na literatura não são usadas para detecção preditiva. Desta forma, escolhemos duas características que distinguem os trabalhos relacionados da nossa proposta: se o trabalho propõe algum modelo para a organização dos componentes em uma aplicação vestível (Organização), e se envolve previsão (Predição). A Tabela 1 dá uma visão geral sobre a literatura relacionada, bem como destaca algumas diferenças de nossa proposta.

**Tabela 1. Resumo comparativo de trabalhos correlatos.**

| Referência                     | Organização | Predição |
|--------------------------------|-------------|----------|
| [Forkan and Khalil 2016]       |             | ✓        |
| [Jaimes et al. 2017]           |             | ✓        |
| [Kan et al. 2015]              |             | ✓        |
| [Gheith et al. 2016]           | ✓           |          |
| [Uddin et al. 2015]            | ✓           |          |
| [Galzarano et al. 2016]        | ✓           |          |
| [Abdali-Mohammadi et al. 2015] | ✓           |          |
| [Chen et al. 2015]             | ✓           |          |
| [Wang et al. 2016]             | ✓           |          |
| Nossa proposta                 | ✓           | ✓        |

## 3. ASAPe

O Arcabouço para Sensoriamento Preditivo em Computação Vestível (ASAPe) é uma arquitetura para suportar aplicativos *wearable* que requerem gerenciamento proativo. Nesse sentido, nossa proposta pode ajudar no desenvolvimento de sistemas que precisam de algum ajuste de uma situação atual para evitar eventos adversos. ASAPe é composto por componentes de software que realizam serviços para um único componente residente no aplicativo cliente. Uma visão geral da arquitetura do sistema proposto pode ser vista na Figura 1.

- **Agent:** sensores responsáveis pela obtenção de dados corporais e ambientais
- **Coordinator Service:** camada responsável por capturar dados dos sensores e enviá-lo para o *Forecast Service*. Ele também chama o *Event Rules Database*, para acessar as regras de ativação, notificar o usuário e propor mudanças no ambiente.
- **ASAPe Application Interface:** especifica o conjunto de ações que podem ser executadas quando ocorre um evento.

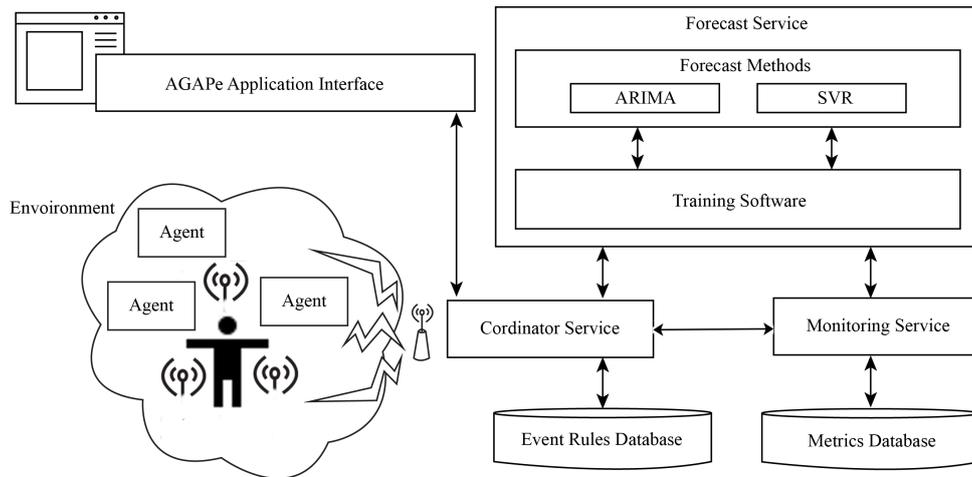


Figura 1. Arquitetura da ASAPe.

- **Event Rules Database:** é um banco de dados relacional que mantém as regras de ativação dos eventos.
- **Metrics Database:** é o banco de dados usado para salvar os valores capturados pelo componente *Agente*. Esses dados podem ser armazenados para definir padrões.
- **Monitoring Service:** essa camada recebe dados do *Coordinator Service* e os grava no *Metrics Database*.
- **Forecast Service:** é o componente que faz a previsão. Ele recebe dados do *Coordinator Service*, que, por sua vez, obtém dados do *Event Rules Database*, e *Monitoring Service*, que o leva do *Metrics Database*. Faz uso de três componentes.
  - **Training Software:** é responsável por tomar os dados de monitoramento e aplicar um dos modelos (ARIMA ou SVR) para obter quadros de predição.
  - **ARIMA:** é um algoritmo estatístico que faz o ajuste de dados de séries temporais para prever pontos futuros.
  - **SVR:** um método supervisionado de aprendizagem de máquina para o reconhecimento de padrões, usado principalmente para análise de classificação e regressão.

### 3.1. Visão geral dos componentes da ASAPe

Os componentes na primeira camada *wearable* são conhecidos como Sensores de Área de Corpo (*Wearable Body Area Sensors (WBAS)*), composto por *hardware* de sensores com potência de computação limitada e largura de banda de comunicação. Um *gateway* (por exemplo, *smartphone* ou *tablet*) envia dados para um ponto final (por exemplo, uma infraestrutura de computação em nuvem) que fornece gerenciamento de dados e oferece suporte a funcionalidades avançadas de mineração de dados, aprendizado de máquina e análise de dados médicos [Hiremath et al. 2014].

A área do corpo monitorado pode possuir uma coleção de nós sensores, ou nós de computador, como eletrocardiograma (ECG), eletromiografia (EMG), eletroencefalografia (EEG), pressão arterial (BP), etc. Para obter essas métricas sensoriais e reagir com os objetos no ambiente, o componente *Agente* pode ser implantado em nós de sensores

ou nós de computador. O *Agent* também pode utilizar alguns desses sensores, sistemas operacionais ou aplicativos para interagir ou reagir aos sensores na pessoa monitorada.

Cada *bit* de dados é enviado para o componente *Coordinator Service*, que os envia para o *Monitoring Service* e, finalmente, registra os dados em um banco de dados relacional, mostrado como o componente *Metrics Database* na arquitetura proposta. Além de receber dados do componente *Agent*, o *Coordinator Service* resolve as requisições do lado cliente e tem a responsabilidade de comunicar e coordenar o *Monitoring Service*, o *Forecast Service* e *Event Rules Database*. O *Coordinator Service* é executado continuamente e realiza eventos que foram especificados pela *ASAPe Application Interface*.

O aplicativo cliente utiliza a *ASAPe Application Interface* para especificar um conjunto de ações que podem ser executadas quando ocorre um determinado evento ou sua ocorrência foi prevista. Os eventos são ativados quando um conjunto de métricas atinge, ultrapassa ou chega perto de um determinado valor. As ações que podem ser realizadas quando ocorre um evento são especificadas por instruções codificadas pelo desenvolvedor e salvas no banco de dados *Event Rules Database*.

Existem dois tipos de bancos de dados em nossa arquitetura: *Event Rules Database* e *Metrics Database*. O Banco de Dados Relacional salva os valores capturados pelos sensores na área do corpo e é mostrado como o Banco de Dados de *Metrics*. Esses dados podem ser usados para prever séries temporais e serem armazenados para definir padrões.

O componente *Event Rules Database* é um banco de dados relacional que mantém as regras para a ativação de eventos reativos ou preditivos. Uma regra de ativação reativa é adicionada sempre que o sistema de monitoramento baseado na arquitetura proposta lê um ou mais valores de uma ou mais das métricas determinadas pelo aplicativo do cliente. O sistema implantado baseado na arquitetura utiliza métodos de predição para prever o instante no tempo em que uma métrica atingirá um certo valor, executando ajustes no ambiente e impedindo a ocorrência de valores ou fatores indesejados, nas métricas ou no meio ambiente.

O componente *Event Rules Database* é chamado continuamente pelo *Coordinator Service* para obter acesso às regras de ativação dos eventos do aplicativo cliente. Para esse fim, o *Coordinator Service* consulta o sistema de monitoramento sobre as métricas que são levadas em consideração pelos eventos e pela ativação. Se o *Coordinator Service* verificar que o evento deve ser disparado, ele realiza chamadas para os componentes *Agent* da Área de Sensor Corporal, para executar as ações do evento identificado.

O *Forecast Service* é o componente responsável para fornecer a possibilidade de estimar o valor de um conjunto de métricas em um quadro de observação futuro com base nos dados de monitoramento. Para verificar as previsões com base nas métricas monitoradas, o *Coordinator Service* recupera continuamente as definições das métricas preditivas e o quadro a ser previsto, bem como os dados de monitoramento dessas métricas, e as usa como entrada para o *Forecast Service*. Este retorna o valor da predição de cada métrica desejada. Usando esses dados, o *Coordinator Service* decide se a execução de qualquer ação no ambiente é necessária.

## 4. Resultados preliminares

O quadro proposto neste artigo foi aplicado a um contexto real. A ideia consiste em uma adaptação do modelo arquitetônico no jogo *wearable* descrito em [Rocha et al. 2015], para testar a eficiência de nossa proposta. Também foi feita uma simulação usando um banco de dados. Nosso objetivo é verificar se o aspecto preditivo da ASAPe pode ser usado em um contexto real e se pode fornecer um serviço eficiente, eficaz e em tempo real de processamento de dados. Também queremos verificar a taxa de sucesso dos algoritmos utilizados em nossa arquitetura.

### 4.1. Estudo de caso: Heart Wars

O jogo "Heart Wars" permite unir a realização de uma atividade física com o lúdico, envolvendo áreas como saúde, *fitness* e jogos. A história do jogo acontece em um cenário fictício, onde um exército de corações é guiado por um personagem chamado Heart, o bárbaro. Os batimentos medidos devem permanecer na área verde, como na Figura 2. Se o jogador está registrando uma frequência cardíaca muito alta, a faixa entra na área vermelha e a pontuação diminui. O mesmo ocorre se o batimento cardíaco for reduzido e a faixa entrar na área azul. Neste contexto, ASAPe poderia ajudar com a verificação do batimento cardíaco antes de obter demasiado acelerar ou não avaliar o mínimo, para manter a faixa na área verde.

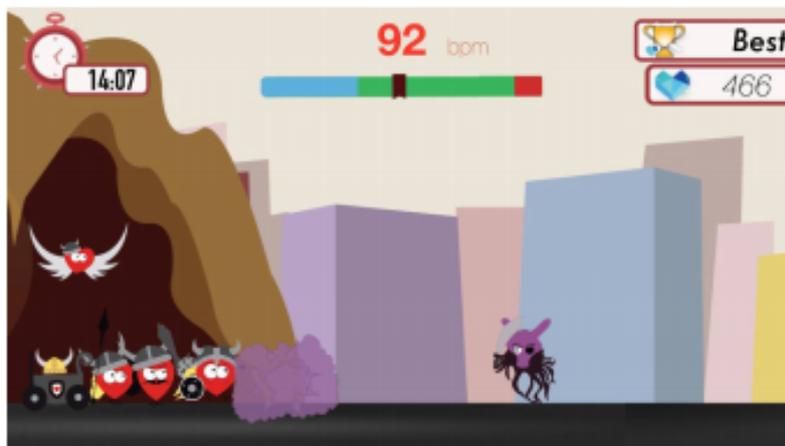


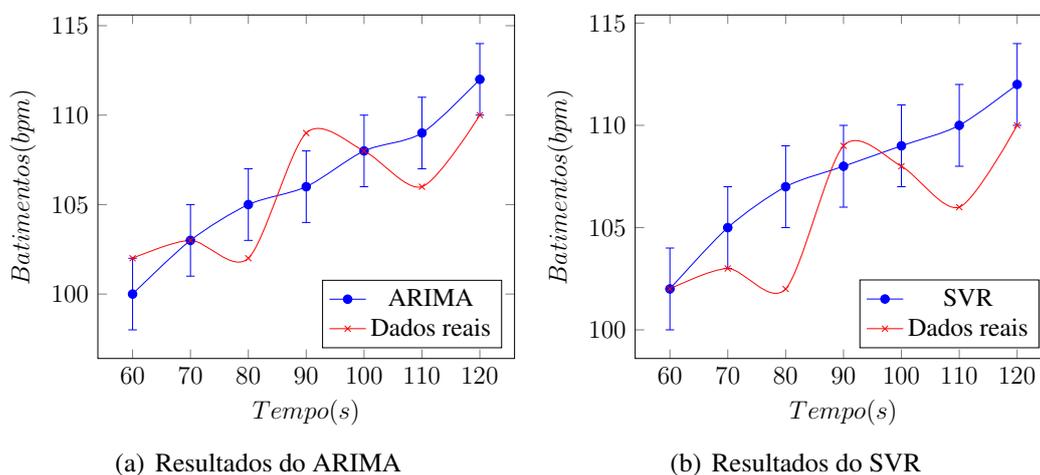
Figura 2. Imagem de jogo Heart Wars

[Rocha et al. 2015, p. 4]

Nestas experiências, fizemos medições reais. A métrica avaliada foi a acurácia, pois é o fator que indica a exatidão dos resultados. Nossos parâmetros de sistema incluem um *smartwatch LG G Watch R*, com um sensor de frequência cardíaca embutido. Os parâmetros de *workload* das medidas reais incluem séries de um exercício chamado *burpee* por 2 minutos, medindo o batimento cardíaco a cada 10 segundos, totalizando 14 medidas. Foi avaliada a previsão para verificar a exatidão dos valores obtidos com a ASAPe e, em seguida, utilizamos duas métricas para avaliar a precisão: *Root-mean-square error* (RMSE) e *Mean absolute percentage error* (MAPE).

Considere um conjunto de 14 amostras, para ambos os algoritmos, escolhemos um conjunto de treinamento composto por sete medidas e depois, classificamos os demais. Ambas as métricas mostraram que os valores preditos eram semelhantes aos dados

verdadeiros. Os resultados do ARIMA foram mais precisos do que os do SVR. O número obtido com RMSE de ARIMA foi 4.3077 e 4.3846 para SVR. Ambos os algoritmos tiveram bons resultados, com uma baixa taxa de erro, mas ARIMA teve o menor número. O resultado obtido com a equação de MAPE foi 0,0152 para ARIMA e 0,0180 para SVR. Novamente a taxa de erro foi pequena, mas ARIMA obteve melhores resultados do SVR.



**Figura 3. Resultados da avaliação do caso de estudo**

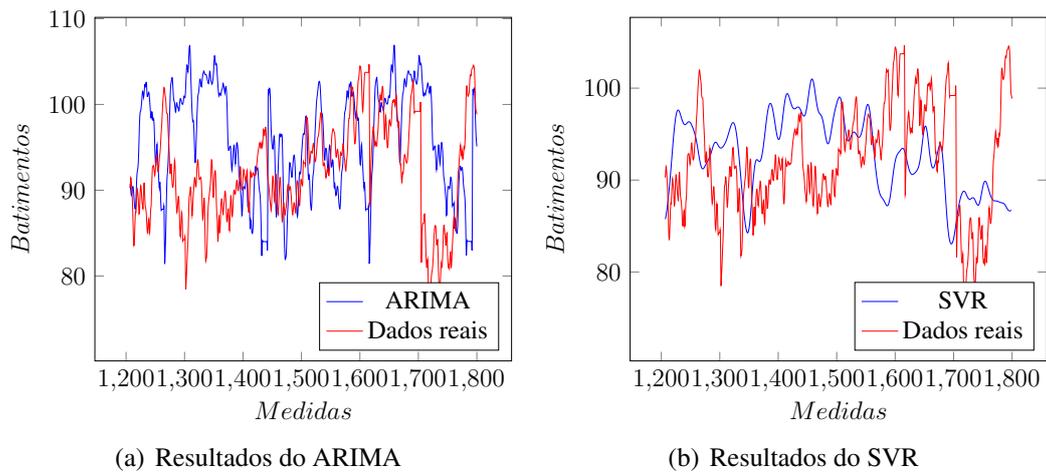
Como discutido anteriormente, o *Forecast Service* decide o método a ser usado para fazer a previsão. Nesse caso, ARIMA foi escolhido para prever o intervalo de tempo, pois como pode ser observado na Figura 3, foi o que obteve melhores resultados em termos de acurácia.

#### 4.1.1. Simulação

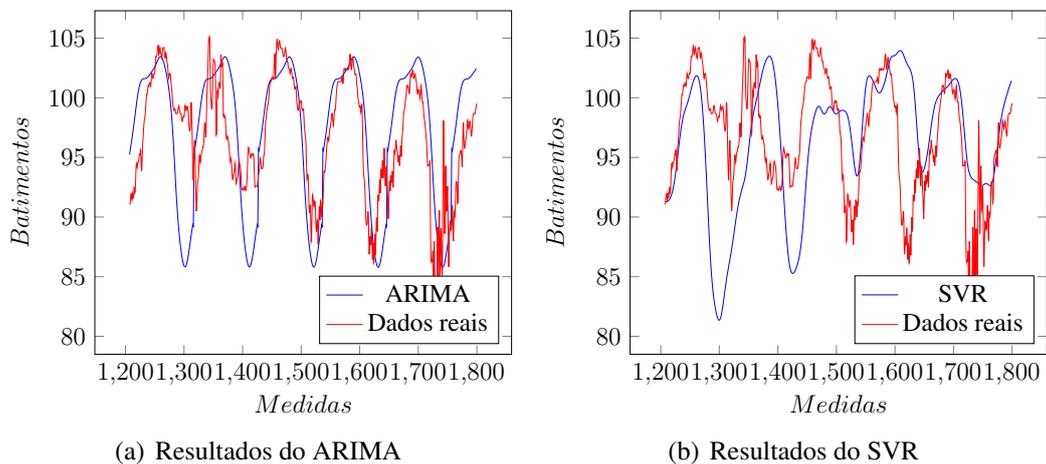
Para essa avaliação, usamos o banco de dados MIT-BIH dataset [Glass et al. 2012], um conjunto de medidas de frequência cardíaca, composto por quatro séries temporais de 1800 dados cada um. As medidas, em unidades de batimentos por minuto (bpm), foram retiradas de dois indivíduos praticando atividades físicas durante 15 minutos, medida em cada 0,5 segundo.

Considere dois conjuntos de 1800 amostras cada um. Foram usados 33% (594 amostras) para testes e 66% (1188 amostras) para treinamento e validação para SVR e ARIMA. Utilizamos os dados de validação (compostos por 30% dos dados de treinamento) para sintonizar todos os hiper parâmetros para SVR e para o modelo ARIMA, usamos o procedimento Hyndman e Khandakar [Hyndman et al. 2007], obtendo os resultados vistos nas Figuras 4 e 5.

Conforme observado nas Figuras 4 e 5, tanto SVR como ARIMA apresentaram valores semelhantes aos dados verdadeiros. Nas simulações, o número obtido com RMSE do primeiro conjunto de dados (Figura 4) de ARIMA foi de 56,44 e 54,90 para SVR. Ambos tiveram bons resultados, mas o SVR apresentou o menor número de erros. No segundo (Figura 5), ARIMA obteve 21.13 e SVR, 43.27. O resultado obtido com a equação de MAPE no primeiro conjunto de dados foi de 6,71 para ARIMA e 6,55 para SVR. No segundo, o resultado foi de 3.77 para ARIMA e 5.22 para SVR.



**Figura 4. Resultados da ASAPe para a série 1.**



**Figura 5. Resultados da ASAPe para a série 2.**

## 5. Conclusão

Nossa proposta consiste em uma arquitetura para predição em infraestruturas *wearable*. Este tipo de aplicação pode ajudar no desenvolvimento de muitos sistemas vestíveis que necessitam da característica preditiva. Poderia também ser adaptado aos sistemas existentes, para melhorar a previsão destes ou, quando não tem uma camada de predição, obter esta característica. A avaliação mostrou que a ASAPe é útil em contextos de recolha de dados em tempo real.

Como trabalhos futuros, vamos incluir a análise de mais dados em diferentes ambientes para obter dados confiáveis sobre precisão. Além disso, planejamos testar diferentes sensores e dispositivos para verificar a forma como a ASAPe funciona com uma variedade de dados. Pretendemos incorporar na ASAPe a outras aplicações para testar sua portabilidade e a precisão em outros contextos. Finalmente, outros recursos podem ser adicionados à ASAPe, como eficiência energética e segurança.

## Agradecimentos

Michelle Cacaís agradece a CAPES pela bolsa de mestrado (processo #1589313). Danielo G. Gomes agradece ao CNPq pela bolsa de produtividade PQ #311878/2016-4 e projeto Universal #432585/2016-8.

## Referências

- Abdali-Mohammadi, F., Bajalan, V., and Fathi, A. (2015). Toward a fault tolerant architecture for vital medical-based wearable computing. *Journal of medical systems*, 39(12):149.
- Chen, M., Zhang, Y., Li, Y., Hassan, M. M., and Alamri, A. (2015). Aiwac: Affective interaction through wearable computing and cloud technology. *IEEE Wireless Communications*, 22(1):20–27.
- Conjeti, S., Singh, R. R., and Banerjee, R. (2012). Bio-inspired wearable computing architecture and physiological signal processing for on-road stress monitoring. In *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pages 479–482. IEEE.
- Forkan, A. R. M. and Khalil, I. (2016). A probabilistic model for early prediction of abnormal clinical events using vital sign correlations in home-based monitoring. In *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*, pages 1–9. IEEE.
- Galzarano, S., Giannantonio, R., Liotta, A., and Fortino, G. (2016). A task-oriented framework for networked wearable computing. *IEEE Transactions on Automation Science and Engineering*, 13(2):621–638.
- Gheith, A., Rajamony, R., Bohrer, P., Agarwal, K., Kistler, M., Eagle, B. W., Hambridge, C., Carter, J., and Kaplinger, T. (2016). Ibm bluemix mobile cloud services. *IBM Journal of Research and Development*, 60(2-3):7–1.
- Glass, L., Hunter, P., and McCulloch, A. (2012). *Theory of heart: biomechanics, biophysics, and nonlinear dynamics of cardiac function*. Springer Science & Business Media.
- Gope, P. and Hwang, T. (2016). Bsn-care: A secure iot-based modern healthcare system using body sensor network. *IEEE Sensors Journal*, 16(5):1368–1376.
- Hiremath, S., Yang, G., and Mankodiya, K. (2014). Wearable internet of things: Concept, architectural components and promises for person-centered healthcare. In *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, pages 304–307. IEEE.
- Hyndman, R. J., Khandakar, Y., et al. (2007). *Automatic time series for forecasting: the forecast package for R*. Number 6/07. Monash University, Department of Econometrics and Business Statistics.
- Jaimes, L. G., Gagneja, K., Akbaş, M. İ., and Vergara-Laurens, I. J. (2017). Future stress, forecasting physiological signals. In *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*, pages 1–5. IEEE.

- Kan, V., Fujii, K., Amores, J., Zhu Jin, C. L., Maes, P., and Ishii, H. (2015). Social textiles: Social affordances and icebreaking interactions through wearable social messaging. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 619–624. ACM.
- Kang, J. J., Adibi, S., Larkin, H., and Luan, T. (2015). Predictive data mining for converged internet of things: A mobile health perspective. In *Telecommunication Networks and Applications Conference (ITNAC), 2015 International*, pages 5–10. IEEE.
- Liu, Y. and Zhou, G. (2012). Key technologies and applications of internet of things. pages 197 – 200, Zhangjiajie, Hunan, China. IEEE.
- Molina-Markham, A., Peterson, R., Skinner, J., Yun, T., Golla, B., Freeman, K., Peters, T., Sorber, J., Halter, R., and Kotz, D. (2014). Amulet: a secure architecture for mhealth applications for low-power wearable devices. In *Proceedings of the 1st Workshop on Mobile Medical Applications*, pages 16–21. ACM.
- Nguyen, L. N. N., Rodríguez-Martín, D., Català, A., Pérez-López, C., Samà, A., and Cavallaro, A. (2015). Basketball activity recognition using wearable inertial measurement units. In *Proceedings of the XVI International Conference on Human Computer Interaction*, page 60. ACM.
- Pels, T., Kao, C., and Goel, S. (2014). Fatbelt: motivating behavior change through isomorphic feedback. In *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology*, pages 123–124. ACM.
- Rocha, U., Oliveira, D., Carvalho, L., Moreira, L., and Viana, W. (2015). Heart wars: um exergame de batalha controlado por dispositivos vestíveis. In *Anais do XV Workshop de Informática Médica (WIM)*, pages 1–10, Porto Alegre, RS, Brasil. SBC.
- Salehizadeh, S., Noh, Y., and Chon, K. (2016). Heart rate monitoring during intense physical activities using a motion artifact corrupted signal reconstruction algorithm in wearable electrocardiogram sensor. In *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on*, pages 157–162. IEEE.
- Spelmezan, D. (2012). An investigation into the use of tactile instructions in snowboarding. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 417–426. ACM.
- Uddin, M., Salem, A., Nam, I., and Nadeem, T. (2015). Wearable sensing framework for human activity monitoring. In *Proceedings of the 2015 workshop on Wearable Systems and Applications*, pages 21–26. ACM.
- Wang, K., Wang, Y., Sun, Y., Guo, S., and Wu, J. (2016). Green industrial internet of things architecture: an energy-efficient perspective. *IEEE Communications Magazine*, 49.

## Avaliação de técnicas de inteligência computacional para identificação de atividades de vida diária

Wylken S. Machado<sup>1</sup>, Pedro H. Barros<sup>1</sup>, Eliana S. Almeida<sup>1</sup>, Andre L. L. Aquino<sup>1</sup>

<sup>1</sup>Instituto de Computação, Universidade Federal de Alagoas, Maceió, Alagoas, Brasil.  
wylken.ufal@gmail.com, pedro\_h\_nr@laccan.ufal.br  
{eliana.almeida, alla}@laccan.ufal.br

**Abstract.** *This work presents the performance evaluation of machine learning algorithms to identify Activity Daily Living (ADLs) and falls. We assess the following algorithms: K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, Extra-Trees and Recurrent Neural Network. We use a dataset collected through a Body Sensor Network with five sensor devices connected through Bluetooth Low Energy interface, called UMA-Fall. Satisfactory results were found, mainly for ADL sitting down and frontal fall, with an accuracy of 100 % with Extra-Trees algorithm.*

**Resumo.** *Neste trabalho apresentamos a avaliação do desempenho de algoritmos de machine learning para identificar Atividades de Vida Diária (ADLs) e quedas. Nós avaliamos os seguintes algoritmos: K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, Extra-Trees e Redes Neurais Recorrentes. Utilizamos um conjunto de dados coletados por uma Body Sensor Networks com cinco dispositivos sensores conectados através da interface Bluetooth Low Energy, chamado UMAFall. Obtivemos resultados satisfatórios, principalmente para as atividades saltar e queda frontal, com 100 % de acurácia, utilizando o algoritmo Extra-Trees.*

### 1. Introdução

O monitoramento da vida diária pode ser utilizado para ajudar idosos ou pessoas com doenças crônicas, promovendo melhorias na qualidade de vida. Quando tecnologias como Smart Homes são utilizadas para ajudar pessoas com necessidades especiais, temos o conceito de Health Smart Home (HSM), que emerge da combinação da telemedicina, automação residencial e sistemas da informação, definida como uma casa inteligente com dispositivos de assistência a saúde e que podem ser acessados remotamente [Mano et al. 2016].

Com o intuito de melhorar aplicações de inferência sobre a estrutura de uma HSM, este trabalho apresenta uma avaliação dos principais algoritmos de aprendizado de máquina para identificar atividades da vida diária (ADLs) e quedas [Katz 1984]. Para isso, um conjunto de dados gerados por uma Body Sensor Networks (BSN), chamada UMAFall, foi utilizado [Casilaria et al. 2017b].

Esta base de dados possui cinco dispositivos sensores, conectados por Bluetooth, para identificar atividades diárias de dezessete indivíduos. São elas: caminhar, correr, inclinar o corpo, pular, deitar, sentar, queda lateral, queda frontal e queda para trás. Cada dispositivo grava simultaneamente os valores do acelerômetro, giroscópio e

magnetômetro. Sobre esses dados foram aplicados seis algoritmos de aprendizado de máquina: K-Nearest Neighbors (k-NN) [Shalev-Shwartz and Ben-David 2014], Naive Bayes [Smola and Vishwanathan 2008], Support Vector Machine (SVM) [Micucci et al. 1995], Decision Tree [Shalev-Shwartz and Ben-David 2014], Random Forest [Breiman 2001], Extra Trees [Geurts et al. 2006] e Redes Neurais Recorrentes [Goodfellow et al. 2016]. Para cada algoritmo foi verificado o desempenho alcançado e o tempo de execução na detecção de ADLs e quedas. Também foi verificado qual dos sensores espalhados pelo corpo apresentou o melhor resultado.

A principal contribuição deste artigo é a análise da acurácia alcançada pelos algoritmos na detecção de ADLs e quedas utilizando o conjunto de dados público UMAFall, além de verificar o tempo de execução e qual a melhor parte do corpo para utilizar sensor que irá coletar as informações dos movimentos, contribuindo com futuros trabalhos que desejam realizar a detecção de ADLs e quedas. Essa análise consiste no estudo dos seis algoritmos citados anteriormente para identificar qual é o mais apropriado para aplicações que utilizam BSN, com as mesmas características do UMAFall, para detectar atividades diárias, e qual a melhor parte do corpo para se instalar os sensores (tórax, pulso, tornozelo, cintura e bolso).

## 2. Trabalhos Relacionados

Existem vários estudos relacionados ao monitoramento de atividades das pessoas em suas residências. Existem alguns trabalhos na literatura, como proposto por Mano et al. [Mano et al. 2016], que utiliza imagens de câmeras para ajudar enfermeiras e cuidadores a identificar emoções nos pacientes, ajudando no processo de recuperação. Outro trabalho, proposto por Aloulou et al. [Aloulou et al. 2013], realiza a implantação de um sistema HSM em um ambiente real para avaliar seu desenvolvimento e usabilidade. Este sistema consiste de sensores e dispositivos controlados por software para detectar precocemente a degradação das condições do paciente. Trabalhos como o proposto por Gope et al. [Gope and Hwang 2016], já abordam o conceito de Body Sensor Networks (BSNs) e os problemas relacionados à segurança da informação que essas aplicações podem apresentar.

Outros trabalhos abordam os métodos de reconhecimento de Atividade da Vida Diária (ADLs). Khan et al. [Khan et al. 2010] utiliza os dados gerados pelo acelerômetro de um smartphone para detectar quatro ADLs, alcançando uma acurácia de até 96 %, apesar de algumas limitações quanto ao posicionamento e orientação do smartphone. Mohammad et al. [Alam and Roy 2017] propõe utilizar dados gerados por uma BSN contextualizada e técnicas de deconvolução para o reconhecimento de ADLs. Farhad et al. [Shahmohammadi et al. 2017] propõe a utilização de dados gerados pelo acelerômetro e giroscópio de um smartwatch para calcular nove características estatísticas e, utilizando essas características, identificar cinco ADLs. Seus resultados apresentaram uma acurácia de até 92 %. Weiss et al. [Weiss et al. 2016] faz uma comparação entre a acurácia alcançada na identificação de 18 ADLs, utilizando dados gerados por um smartphone e um smartwatch, concluindo que os dados gerados pelo smartwatch apresentaram melhores resultados, principalmente para ADLs que são executadas predominantemente com as mãos.

Existe uma variedade de técnicas e soluções para o problema apresentado. Con-

tudo, é necessário um estudo comparativo mais aprofundado sobre os impactos das técnicas de Machine Learning no cenário de detecção de ADLs, utilizando diferentes tipos de sensores em várias partes do corpo, e assim, contribuir para a escolha do melhor local de instalação dos sensores. Além disso, o trabalho apresenta o tempo de execução dos algoritmos para classificar as ADLs.

### 3. Algoritmos de Machine Learning

Nessa seção é apresentada uma breve descrição dos algoritmos de Machine Learning utilizados neste estudo. Todos utilizam a abordagem de aprendizado supervisionado. Como esse trabalho se propõe a verificar a acurácia de diferentes técnicas, os algoritmos foram escolhidos com o objetivo de representar os principais métodos matemáticos e computacionais aplicados ao Aprendizado de Máquina. Foi utilizada o tipo de rede neural RNN para possibilitar a captura de features temporais.

O primeiro algoritmo utilizado foi o k-Nearest Neighbors (k-NN) [Cover and Hart 1967] é um algoritmo de aprendizado supervisionado baseado em instâncias, onde é necessário ter uma base de dados rotulada para realizar o treinamento do classificador e identificar novos elementos. Essencialmente o k-NN calcula as distâncias entre um elemento desconhecido e os elementos do conjunto de treinamento. Existem vários métodos para calcular a distância, sendo que a mais utilizada é a distância Euclidiana [Mulak and Talhar 2015],

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \dots + (x_n - x'_n)^2}. \quad (1)$$

Além da distância Euclidiana, também são utilizadas as distâncias de *Chebyshev* e *Manhattan*. O algoritmo consiste em percorrer todo o conjunto e computar a distância  $d$  entre  $x$  e as observações de treinamento. Logo após, escolhe os  $k$  elementos mais próximos a  $x$  formando o conjunto  $A$ . Então, estima-se a probabilidade condicional para cada classe em  $A$

$$p(y = j|X = x) = \frac{1}{k} \sum_{i \in A} I(y_i = j), \quad (2)$$

onde  $I(z)$  retorna **1** quando o argumento  $z$  for verdadeiro e **0** se for falso. E, por fim, atribui-se a classe com maior probabilidade a  $x$ .

Um passo essencial para os cálculos do k-NN é definir o valor de  $k$ , que deve ser um valor ímpar para evitar complicações na inferência. Além disso, se o valor de  $k$  for muito pequeno, a classificação pode ser sensível a ruído, e se for muito grande, pode incluir elementos de outras classes. Nesse trabalho o valor que apresentou o melhor resultado foi  $k = 5$ .

O segundo algoritmo considerado foi o Naive Bayes. Amplamente discutido em [Lewis 1998], este método utiliza uma abordagem probabilística baseada no Teorema de Bayes,

$$p(c|x) = \frac{p(x|c)p(c)}{p(c)}. \quad (3)$$

Os atributos são considerados condicionalmente independentes, dado uma classe  $C_k$ . Ou seja, a presença de uma característica não está relacionado a nenhuma outra.

Dado um elemento  $x = \{x_1, x_2, \dots, x_n\}$  com  $n$  atributos, o Naive Bayes prediz a classe  $C_k$  para  $x$  utilizando a probabilidade calculada pelo Teorema de Bayes,

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x_1, x_2, \dots, x_n|C_k)p(C_k)}{p(x_1, x_2, \dots, x_n)}, \quad (4)$$

onde  $C$  é o conjunto com as  $k$  classes possíveis. Após calcular as probabilidade é atribuído a classe  $C_k$  com maior probabilidade ao elemento  $x$ .

O terceiro algoritmo considerado foi o Support Vector Machine (SVM). No nosso caso o objetivo do algoritmo é encontrar o hiperplano que melhor separa as classes do conjunto de treinamento, ou seja, que maximiza sua margem entre as classes do conjunto.

Os elementos mais próximos do hiperplano são chamados de vetores de suporte. Encontrar esses vetores é a principal tarefa da fase de treinamento do SVM. A classificação dos novos elementos é realizada utilizando apenas os vetores de suporte. Após o treinamento o SVM pode descartar todo o resto dos dados de treinamento, o que faz com que esse algoritmo seja bastante eficiente para realizar a classificação, já que utiliza apenas alguns vetores de suporte, e não todo o conjunto.

Em conjuntos não separáveis linearmente, o SVM utiliza funções matemáticas chamadas de kernels que transformam esse conjunto em outro linearmente separável. Normalmente são utilizados os kernels: linear, polinomial, radial e sigmoide. Para construir um hiperplano ótimo, o SVM aplica um algoritmo de treinamento iterativo, utilizado para maximizar a margem entre esse hiperplano e os elementos de diferentes classes.

Suponha que temos um conjunto de dados  $[(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)]$  com classes linearmente separáveis, onde  $y_i = -1$  para  $x_i$  da Classe 0 e  $y_i = 1$  para  $x_i$  da Classe 1. A reta que separa essas duas classes pode ser definida como  $\vec{w} \cdot \vec{x} + b = 0$ , onde  $\vec{w}$  e  $\vec{x}$  são dois vetores dimensionais. Definimos o vetor negativo  $\vec{x}_n$  em que todas as entradas são da Classe 0, e o vetor positivo  $\vec{x}_p$  onde todas as entradas são da Classe 1. Agora podemos definir os limites inferior e superior da nossa reta:  $\vec{w} \cdot \vec{x}_n + b = -1$  para Classe 0, e  $\vec{w} \cdot \vec{x}_p + b = 1$  para Classe 1. A distância entre esses dois limites é dada por  $\frac{2}{\|\vec{w}\|}$ , e a margem  $M$  é dada por  $\frac{1}{\|\vec{w}\|}$ . Para maximizar  $M$  devemos minimizar  $\|\vec{w}\|$ . O SVM utilizara técnicas para resolver problemas de otimização com o objetivo de minimizar  $\|\vec{w}\|$ .

Esse é um exemplo em um ambiente com duas dimensões. A mesma técnica pode ser utilizada para  $N$  dimensões.

Em seguida avaliamos os algoritmos de Árvore de Decisão [Quinlan 1986]. Segundo Shalev-Shwartz et al. [Shalev-Shwartz and Ben-David 2014], uma Árvore de Decisão é um preditor  $h : x \rightarrow y$  que prediz a classe associada com a uma instância  $x$  percorrendo uma estrutura de dados em árvore, dos seus nós raízes até um nó folha. Uma Árvore de Decisão é composta por vários nós. As folhas contêm as classes de interesse, e os nós intermediários representam as características que definem as classes. Esse classificador é de fácil entendimento e interpretação, sendo essa uma das vantagens em utilizá-lo.

Para obter bons resultados se faz necessário escolher métricas adequadas para selecionar as características que irão formar os nós. Geralmente é utilizado o Ganho de

Informação que usa o conceito de entropia,

$$h(x) = \sum_{i=1}^k p_i \log_2(p_i), \quad (5)$$

onde  $x$  é o atributo,  $k$  é a quantidade de classes do conjunto de dados e  $p_i$  é a probabilidade de cada uma das classes acontecer para um dado atributo. Para o cálculo do Ganho de Informação utilizamos

$$ig(x) = h(classe) - h(x). \quad (6)$$

Uma variação da árvore de decisão é o algoritmo Random Forest [Breiman 2001] que cria diversas árvores de decisão e, para cada árvore,  $k$  atributos são escolhidos aleatoriamente para formá-la. Após a escolha dos atributos pelo sistema, as árvores são construídas da forma tradicional, utilizando o ganho de informação. Sua estrutura permite executar o algoritmo de forma distribuída, tornando-se eficiente para problemas de classificação em enormes bases de dados. A acurácia tende a aumentar de acordo com o número de árvores utilizadas.

O algoritmo Extra-Trees [Geurts et al. 2006] é similar ao Random Forest. A diferença entre eles é que o Extra-Trees adiciona mais uma camada de aleatoriedade para montar as árvores, ou seja, o algoritmo utiliza uma estratégia aleatória para montar os nós, em vez do ganho de informação ou outras métricas.

Por fim, consideramos o algoritmo baseado em redes neurais recorrentes, ou RNN. Uma Rede neural tradicional (FNN) tem seus parâmetros separados para cada vetor de features de entrada, ou seja, esta necessita aprender todas as regras separadamente. Diferentemente disto, as RNN compartilham os mesmos pesos através do tempo.

Logo, para um modelo dinâmico clássico com dependência temporal, a rede pode ser caracterizada como,

$$s^{(t)} = f(s^{(t-1)}; \theta), \quad (7)$$

onde  $s^t$  é um estado do sistema no tempo  $t$  e  $\theta$  é um vetor de features de entrada para o sistema.

Vemos em Goodfellow et al. [Goodfellow et al. 2016] que a RNN representada na Equação 8 é universal (como pode ser visto na equação 7 que representa a mesma equação de um modelo dinâmico clássico) no sentido de que qualquer função que pode ser computada por uma Máquina de Turing pode ser calculada por uma rede recorrente de tamanho finito. Ainda em Goodfellow et al. [Goodfellow et al. 2016], vemos que as RNN são caracterizadas pelas equações:

$$\mathbf{a}^{(t)} = b + W\mathbf{s}^{(t-1)} + U\mathbf{x}^{(t-1)} \quad (8)$$

$$\mathbf{s}^{(t)} = \tanh(\mathbf{a}^{(t)}) \quad (9)$$

$$\mathbf{d}^{(t)} = c + V\mathbf{s}^{(t)} \quad (10)$$

$$\mathbf{o}^{(t)} = \text{softmax}(\mathbf{d}^{(t)}), \quad (11)$$

onde  $\mathbf{x}^{(t-1)}$  é a entrada do passo temporal  $t$ ,  $b$  e  $c$  são os vetores de viés das redes neurais,  $U$ ,  $V$  e  $W$  são, respectivamente, matriz de peso para as conexões entrada-camada oculta, camada oculta-saída e camada oculta-camada oculta.

Porém, como a RNN tem dificuldades de aprender longas dependências temporais por conta do problema do desaparecimento do gradiente (intrínseco ao treinamento de redes neurais artificiais), é indicado usar um tipo especial de RNN chamada LSTM (Long Short Term Memory), devido a adição de alguns elementos internos de memória. Para este trabalho foi usada a arquitetura proposta por Karien et al. [Karim et al. 2018] onde os autores utilizaram uma rede LSTM concatenada com algumas camadas de redes neurais para refino na performance dos modelos treinados.

#### 4. Base de dados

Foram encontradas várias bases de dados para detecção de ADLs e quedas, tais como [Ojetola et al. 2015], [Vavoulas et al. 2013], [Vavoulas et al. 2016], [Gasparrini et al. 2014], [Sucerquia et al. 2017] e [Micucci et al. 2016]. Porém, a UMA-Fall<sup>1</sup> [Casilaria et al. 2017a] foi a que apresentou uma maior quantidade de dispositivos e sensores espalhados pelo corpo, característica fundamental para o estudo realizado.

O UMAFall contém o registro de uma BSN que utiliza três tipos de sensores, acelerômetro, giroscópio e magnetômetro. Ela identifica seis tipos de ADLs: caminhar, correr, inclinar o corpo, pular, deitar, sentar, e três tipos de quedas: queda lateral, queda frontal e queda para trás. A BSN utilizada consiste de um smartphone android colocado no bolso direito dos voluntários, usado como nó central, e quatro dispositivos sem fio posicionados em diferentes partes do corpo: tórax, pulso, tornozelo e cintura. Durante a coleta de dados foram utilizados dois modelos de smartphones, Samsung S5 e LG G4, ambos integrados com acelerômetro e conectividade Bluetooth. Cada dispositivo sem fio consiste em um SensorTag CC2650 Multi-Standard da Texas Instruments, que possui interface Bluetooth Low Energy, acelerômetro tri-axial, giroscópio tri-axial e magnetômetro. Cada dispositivo utiliza uma bateria de lítio CR2032 3v.

Originalmente o UMAFall consiste em 531 arquivos CSV, onde cada um é nomeado de forma a identificar o voluntário, a ADL ou o tipo de queda e a data. Dentro de cada arquivo encontramos os dados: timestamp, número da amostra, eixo X, eixo y, eixo z, tipo do sensor (acelerômetro, giroscópio e magnetômetro) e local do sensor (bolso, cintura, tornozelo, pulso e tórax).

#### 5. Metodologia

Todo o experimento foi realizado utilizando os dados contidos no conjunto de dados UMAFall com o total de 3.277.950 registros. Foram utilizados os dados individualizados por voluntário, ou seja, os algoritmos executaram a classificação utilizando os dados gerados por uma única pessoa em cada interação até passar por todos os voluntários. As características do movimento variam de um indivíduo para outro. Essa abordagem individualizada apresenta melhores resultados [Weiss et al. 2016], porém não pode ser generalizada, ou seja, utilizar dados de indivíduos diferentes para os conjuntos de treinamento e teste.

<sup>1</sup>Dataset público disponível em [http://webpersonal.uma.es/de/ECASILARI/Fall\\_ADL\\_Traces/UMA\\_FALL\\_ADL\\_dataset.html](http://webpersonal.uma.es/de/ECASILARI/Fall_ADL_Traces/UMA_FALL_ADL_dataset.html)

Utilizamos a linguagem de programação Python [Python 2017], versão 3.6.4, em conjunto com a biblioteca de machine learning Scikit-learn [Pedregosa et al. 2011], versão 0.19.1, TensorFlow [Brain 2017], versão r1.6, e o ambiente de desenvolvimento PyCharm Community [JetBrains 2018], versão 2018.1. A RNN foi executada em uma GPU GTX 1080 TI, e os demais algoritmos em um desktop com CPU FX 8300, 3.3GHz e 8GB de memória RAM.

Nosso objetivo é verificar a precisão dos algoritmos utilizando os dados de diferentes sensores, localizados em diversas partes do corpo. Para isso, foi utilizado o método validação cruzada  $k$ -fold [Stone 1974]. Nesse método o conjunto de dados é dividido em  $k$  subconjuntos, logo após, são realizadas  $k$  interações de 1 até  $k$ , onde o subconjunto  $k$  é utilizado para teste e o restante ( $k - 1$ ) para treinamento do classificador. Nesse trabalho utilizamos  $k = 10$ , valor normalmente utilizado na literatura.

## 6. Resultados

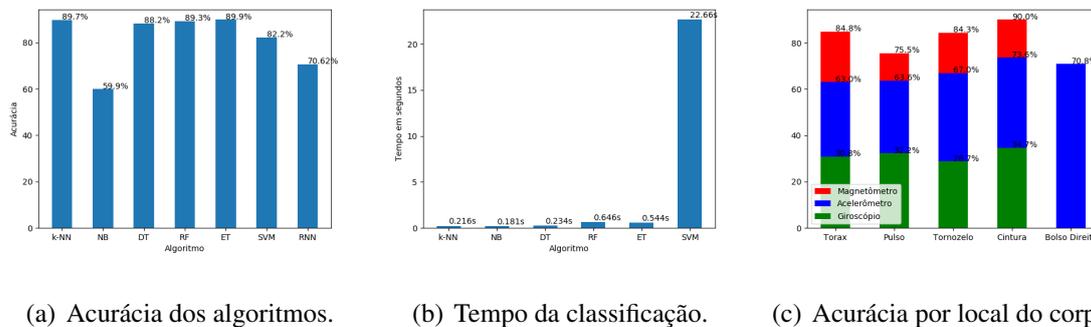


Figura 1: Acurácia e tempo de execução por algoritmo.

A Figura 1(a) mostra a acurácia obtida com todos os métodos propostos. Verifica-se que o Extra-Trees obteve o melhor resultado, seguido do k-NN e do Random Forest, com 89.9%, 89.7% e 89.3%, respectivamente. Na figura NB = Naive Bayes, DT = Decision Tree, RF = Random Forest, ET = Extra-Trees. Sabendo que o Extra-Trees obteve o melhor resultado, montamos a Figura 1(c) para identificar qual local do corpo apresentou o melhor resultado. Observamos que o sensor que obteve o melhor resultado foi o magnetômetro instalado na cintura do voluntário. Também podemos identificar que o dispositivo do bolso direito registrou apenas os dados do acelerômetro.

Para analisar com mais detalhes como o Extra-Trees realiza a classificação, utilizamos a matriz de confusão apresentada na Figura 2. Nesta matriz observa-se que todas as ADLs e tipos de queda, exceto a ADL correr, obtiveram uma acurácia superior ou igual a 90% que, em termos de machine learning, são resultados promissores. O mais notável foram os resultados para a ADL saltar e para a queda frontal, onde todas as ocorrências foram classificadas de forma correta, alcançando 100% de precisão nessas atividades.

Tendo em vista que os algoritmos Extra-Trees, k-NN e Random Forest obtiveram resultados próximos, a Figura 1(b) foi utilizada para verificar qual desses algoritmos seria mais indicado para realizar a classificação em aplicações de tempo real. Verificamos que os três realizaram a classificação em menos de 1s, porém o k-NN foi duas vezes mais rápido que o Extra-Trees. Nessa situação, para uma aplicação específica, é recomendável

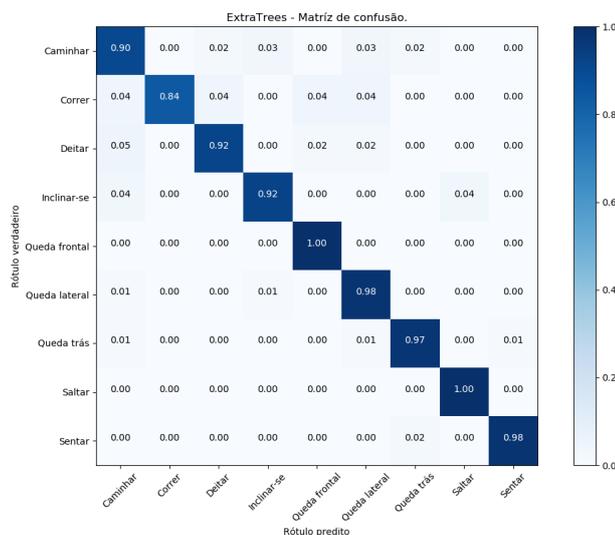


Figura 2: Matriz de confusão utilizando o Extra-Trees com os dados gerados pelo magnetômetro posicionado na cintura.

averiguar qual algoritmo seria o mais indicado, ou seja, se devemos priorizar a acurácia ou o tempo de classificação. O RNN demorou cerca de 27 min para realizar a classificação, muito mais que os outros algoritmos, dessa forma ele não foi apresentado nesse gráfico.

## 7. Conclusão

Uma das funções essenciais do monitoramento é a identificação de queda do indivíduo. Uma simples falha nesse tipo de monitoramento pode colocar o indivíduo monitorado em risco. Os resultados apresentam o Extra-Trees, o k-NN e o Random Forest com uma precisão aceitável para esse tipo de aplicação, principalmente para o algoritmo Extra-Trees na identificação da ADL saltar e da queda frontal, com 100% de precisão. O Extra-Trees também apresenta uma ótima acurácia para os outros tipos de ADL e quedas. O Extra-Trees e o k-NN apresentaram resultados muito semelhantes, mas o k-NN foi duas vezes mais rápido que o Extra-Trees.

Em trabalhos futuros pretendemos utilizar o conhecimento adquirido para elaborar modelos com o intuito de generalizar a classificação, ou seja, utilizar um conjunto de treinamento padrão para realizar detecção de ADLs e quedas de outras pessoas, mantendo a acurácia da abordagem individual. Também desejamos implementar estratégias para descoberta de novas ADLs que não pertençam ao conjunto inicial de treinamento, possibilitando ampliar e personalizar o leque de atividades monitoradas.

## Agradecimentos

Nós agradecemos o apoio financeiro ao CNPq, FAPEAL, FAPESP.

## Referências

Alam, M. A. U. and Roy, N. (2017). Single bsn-based multi-label activity recognition. *The Third IEEE International Workshop on Sensing Systems and Applications Using Wrist Worn Smart Devices, 2017*.

- Aloulou, H., Mokhtari, M., Tiberghien, T., Biswas, J., Phua, C., Lin, J. H. K., and Yap, P. (2013). Deployment of assistive living technology in a nursing home environment: methods and lessons learned. *BMC Medical Informatics and Decision Making*, pages 1–17.
- Brain, G. (2017). Tensorflow api for python, version r1.6. Available at [https://www.tensorflow.org/api\\_docs/python/](https://www.tensorflow.org/api_docs/python/).
- Breiman, L. (2001). Random forests. *Machine Learning*, pages 5–32.
- Casilaria, E., Santoyo-Ramón, J. A., and Cano-García, J. M. (2017a). Umafall: A multisensor dataset for the research on automatic fall detection. *The 14th International Conference on Mobile Systems and Pervasive Computing*, pages 32–29.
- Casilaria, E., Santoyo-Ramón, J. A., and Cano-García, J. M. (2017b). Umafall: A multisensor dataset for the research on automatic fall detection. *The 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017)*, pages 32–39.
- Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, pages 21–27.
- Gasparri, S., Cippitelli, E., Spinsante, S., and Gambi, E. (2014). A depth-based fall detection system using a kinect@ sensor. *Sensors*, page 2756–2775.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, pages 3–42.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Gope, P. and Hwang, T. (2016). Bsn-care: A secure iot-based modern healthcare system using body sensor network. *IEEE Sensors Journal*, pages 1368–1376.
- JetBrains (2018). Pycharm v2018.1. Available at <https://www.jetbrains.com/pycharm/download/>.
- Karim, F., Majumdar, S., Darabi, H., and Chen, S. (2018). Lstm fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669.
- Katz, S. (1984). Assessing self-maintenance: Activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society*, 31:721–727.
- Khan, A. M., Lee, Y. K., Lee, S. Y., and T.S. Kim, T. K. (2010). Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. *5th International Conference on Future Information Technology. IEEE*, pages 1–6.
- Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. *European Conference on Machine Learning*, pages 4–15.
- Mano, L. Y., Façal, B. S., Nakamura, L. H. V., Gomes, P. H., Libralon, G. L., Menequete, R. I., Filho, G. P. R., Giancristofaro, G. T., Pessin, G., Krishnamachari, B., and Ueyama, J. (2016). Exploiting iot technologies for enhancing health smart homes through patient identification and emotion recognition. *Elsevier Computer Communications*, pages 178–190.

- Micucci, D., Mobilio, M., and Napolitano, P. (1995). Support-vector networks. *Springer Netherlands*, pages 273–297.
- Micucci, D., Mobilio, M., and Napolitano, P. (2016). Unimib shar: a new dataset for human activity recognition using acceleration data from smartphones. *IEEE Sensors Lett*, pages 15–18.
- Mulak, P. and Talhar, N. (2015). Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset. *International Journal of Science and Research (IJSR)*, pages 2101–2104.
- Ojetola, O., Gaura, E., and Brusey, J. (2015). Data set for fall events and daily activities from inertial sensors. *Proceedings of the 6th ACM Multimedia Systems Conference (MMSys'15)*, pages 243–248.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, pages 2825–2830.
- Python (2017). Python language reference, version 3.6.4. Available at <https://docs.python.org/release/3.6.4/>.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, pages 81–106.
- Shahmohammadi, F., Hosseini, A., King, C. E., and Sarrafzadeh, M. (2017). Smartwatch based activity recognition using active learning. *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 321–329.
- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
- Smola, A. and Vishwanathan, S. (2008). *Introduction to Machine Learning*. Cambridge University Press.
- Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, pages 111–147.
- Sucerquia, A., López, J., and Vargas-bonilla, J. (2017). Sisfall: A fall and movement dataset. *Sensors (Basel)*, pages 1–14.
- Vavoulas, G., Chatzaki, C., Malliotakis, T., and Padiaditis, M. (2016). The mobiact dataset: Recognition of activities of daily living using smartphones. *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health (ICT4AWE)*, pages 143–151.
- Vavoulas, G., Padiaditis, M., Spanakis, E., and Tsiknakis, M. (2013). The mobifall dataset: An initial evaluation of fall detection algorithms using smartphones. *Proceedings of the IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE 2013)*, pages 1–4.
- Weiss, G. M., Timko, J. L., Gallagher, C. M., Yoneda, K., and Schreiber, A. J. (2016). Smartwatch-based activity recognition : A machine learning approach. *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pages 426–429.

## Avaliando o Tardigrade: um framework para o desenvolvimento de jogos de cartas ubíquos

Willian Almeida Rodrigues<sup>1</sup>, Cecília Costa<sup>2\*</sup>, Nécio de Lima Veras<sup>3</sup>,  
Vitor Cortez<sup>4</sup>, Ricardo Andrade<sup>5</sup>, Fernando Trinta<sup>6</sup>  
Glaudiney M. Mendonça Junior<sup>7</sup>, Windson Viana<sup>8</sup>

<sup>1</sup>Universidade Federal do Ceará (UFC)  
Campus do Pici - Bloco 942-A  
Fortaleza — Ceará — Brasil

w.almeida.w@gmail.com, ceciliaaguilar@great.ufc.br

necio.veras@ifce.edu.br, vitoracortez@gmail.com

ricardoalr@ufpi.edu.br, fernandotrinta@great.ufc.br

glaudiney@virtual.ufc.br, windson@great.ufc.br

**Abstract.** *Researchers in Ubiquitous Games aim at discovering how computing devices and sensors can promote entertainment. They try to extend game experience beyond the virtual world by extrapolating its boundaries to the player's physical world. These games inherit traditional challenges of both domains (i.e., game development and ubiquitous systems), such as device heterogeneity, resource discovery, communication, and limitation of available resources. In this paper, we propose a framework for the development of ubiquitous card games, called Tardigrade. It seeks to minimize the challenges of implementing this kind of ubiquitous game. We implemented an evaluation experience with eight mobile developers. The results are discussed further in this article.*

**Resumo.** *Pesquisadores de Jogos Ubíquos buscam descobrir como dispositivos de computação e sensores podem promover entretenimento. Eles tentam estender a experiência do jogo para além do mundo virtual extrapolando seus limites para o mundo físico. Esses jogos herdam desafios tradicionais de ambos os domínios (i.e., desenvolvimento de jogos e sistemas ubíquos), como heterogeneidade, descoberta de recursos, comunicação, e limitação de recursos. Neste artigo, é proposto e avaliado um framework de jogos de cartas ubíquos, chamado Tardigrade. O framework minimiza os desafios de implementar este tipo de jogo. Por fim, o artigo apresenta uma avaliação com oito desenvolvedores de dispositivos móveis e discute os resultados alcançados.*

---

\*Bolsista de mestrado pela Funcap

## 1. Introdução

Os jogos ubíquos ou pervasivos representam uma nova era para o entretenimento e a interação, estendendo a experiência dos jogos para além do mundo virtual [Silva et al. 2017]. Jogos ubíquos borram a fronteira dos mundos virtual e real do jogador nos quais um ou mais elementos do jogo são influenciados pelos limites espaciais, temporais e sociais do jogador [Montola et al. 2009]. Esses jogos usam sensores, informações contextuais e elementos tangíveis (e.g., lugares do mundo real, brinquedos, cartas) e contemplam um amplo espectro: desde jogos *singleplayer* em smartphones até jogos massivos de realidade mista como o recente sucesso do Pokemon Go [Paavilainen et al. 2017]. Os jogos ubíquos herdam, ao mesmo tempo, desafios tradicionais do desenvolvimento de jogos (e.g., gerenciamento de equipes multidisciplinares, autoração de multimídia) e problemas intrínsecos aos sistemas ubíquos, tais como: a heterogeneidade dos dispositivos, a descoberta de dispositivos, a comunicação volátil e a limitação dos recursos disponíveis.

Esta pesquisa endereça um subtipo de jogos ubíquos focado no aumento da experiência de jogos de mesa (i.e., do inglês *boardgames*). Neste domínio, é possível encontrar alguns trabalhos que relatam aplicações e jogos com características ubíquas [Römer and Domnitcheva 2002] [Sakamoto et al. 2014] [Cheung et al. 2013]. Contudo, esses trabalhos não focam em nenhuma abordagem que promova o reuso ou a redução da complexidade de desenvolvimento, possuindo basicamente o objetivo de construir um ambiente de jogo e a integração com dispositivos distintos. Por exemplo, o trabalho [Barhaug 2017] que faz uso de uma ferramenta com propósito similar ao *framework*, porém seu foco é em técnicas de *design* de jogos híbridos para ensino.

Nesse sentido, esta pesquisa propõe o conceito de jogo de cartas ubíquos, pautado em um mecanismo que auxilia sua utilização por meio de dispositivos móveis. Neste artigo, em especial, o principal objetivo é propor e avaliar um *framework* para auxiliar o desenvolvimento de um tipo de jogo ubíquo que envolve cartas tangíveis (físicas). A pesquisa reflete a materialização de um modelo para criação de jogos de cartas ubíquos, denominado Tardigrade, cujas ideias iniciais são descritas em [Rodrigues et al. 2016]. O *framework* foca em minimizar o esforço de implementação deste tipo de jogo na plataforma Android. Uma das contribuições diretas obtidas a partir desse tipo de aplicação é o aumento da experiência de um jogo analógico com aplicativos móveis para reconhecer cartas, apresentar o estado do tabuleiro ou mostrar animações digitais. Estes podem ser vistos como exemplos da materialização dos princípios de Computação Ubíqua no domínio dos jogos de cartas, tendo em vista que a Computação Ubíqua pode incorporar dispositivos computacionais em objetos comuns, de uma forma que a interação com o usuário se torna natural e quase imperceptível [Weiser 1999].

## 2. Cenário Motivador

”Com seu baralho de cartas colecionáveis no bolso, David caminha em um shopping de Fortaleza. Seu Android vibra anunciando que existe um oponente nas proximidades dele, Jack. David começa o combate juntando uma combinação de cartas na parte de trás do seu aparelho, fazendo Jack contra-atacar colocando também uma carta e começando a balançar seu celular com afinco. David na verdade utilizou uma série de cartas de ataque, e Jack utilizou uma carta de defesa que tem seu resultado aprimorado com o movimento de chacoalhar o aparelho. Os ruídos do shopping permitiram Jack contra-atacar utilizando uma das suas cartas que aumenta o ataque de acordo com o ruído do local.

David nota que a partida está perdida, porém decide continuar defendendo-se do combate cobrindo os sensores de iluminação e proximidade com o intuito de impedir os danos do próximo ataque. Como recompensa, David deveria entregar uma de suas cartas usadas em combate a Jack. David entrega a carta que já não o pertence mais e parabeniza Jack pela vitória, mas deixando claro que voltará para reconquistar a carta perdida. Durante a batalha, informações de contato dos jogadores foram trocadas, permitindo com que David envie mensagens para Jack e, assim, possa ter a chance de uma revanche.”

O cenário motivador apresenta mecânicas de combate baseadas em cartas tangíveis e um aplicativo de combates virtuais. Cartas são reconhecidas multi ou unicamente por proximidade pelos dispositivos, que reagem de acordo com os efeitos de cada carta. Além disso, ele incorpora o conceito de cartas sensíveis ao contexto apresentado em [Rodrigues et al. 2016]. Essas cartas necessitam de informações contextuais (i.e., que descrevem a situação do jogador) para serem bem-sucedidas, como aumentar o dano baseado no clima do local ou no nível de ruído onde o combate está sendo realizado. Já outras cartas utilizam sensores, como o de proximidade, iluminação e acelerômetro, para terem seus efeitos maximizados ou minimizados. Existem ainda cartas que são apenas identificadas pela aplicação para que seus efeitos sejam aplicados. Essas e outras funcionalidades foram implementadas no *framework* descrito a seguir.

### 3. Tardigrade

O *framework* Tardigrade tem como propósito disponibilizar uma forma de desenvolver jogos de cartas ubíquos que apresentem interação do real com o virtual, podendo ou não fazer uso do contexto do jogador como componente de jogo (e.g., localização física). O *framework* proporciona a detecção das cartas, a associação delas ao estado do jogo virtual, a captura do contexto do jogador e o compartilhamento dessas informações entre múltiplos dispositivos. O modelo base do Tardigrade foi projetado para três possíveis situações de jogo. Na primeira situação, tem-se um único dispositivo móvel sendo compartilhado por todos os jogadores. Uma segunda situação, que representa uma variação da anterior, indica que mais de um dispositivo permitiria acessar o estado de jogo a partir de alguma referência compartilhada fisicamente (e.g., Qr Code). Ampliando o conceito de compartilhamento da segunda situação, tem-se uma terceira visão de jogos suportados pelo modelo, que é composta por múltiplos dispositivos conectados em rede. Nessa situação, os dispositivos podem compartilhar e trocar o estado de jogo e compartilhar eventos detectados em cada dispositivo de forma individual, ou seja, cada jogador tem uma visão distinta do jogo.

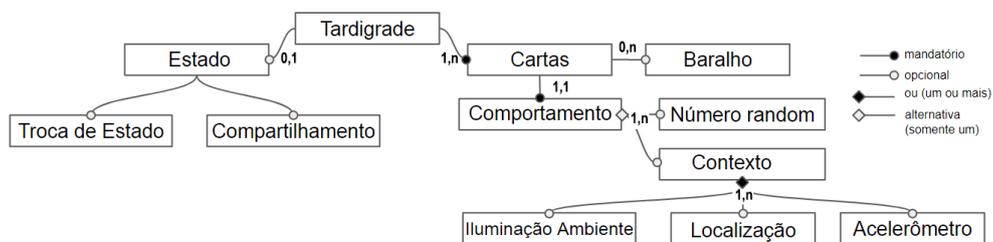


Figura 1. Diagrama de *Features* do Modelo de Jogos do Tardigrade.

No diagrama, as ligações terminadas em círculos preenchidos representam relações mandatórias. Dessa forma pode-se observar que um jogo de cartas ubíquo base-

ado no modelo do Tardigrade tem as seguintes características: **a)** estados que descrevem o jogo, **b)** esses estados podem ser compartilhados e/ou trocados entre diferentes instâncias (jogadores); **c)** todos os jogos devem ter cartas que **d)** podem ser gerenciadas e armazenadas em um baralho; e **e)** essas cartas possuem comportamento de um ou mais tipos.

As visões de sistemas ubíquos [Spínola and Travassos 2012] e de jogos ubíquos [Cheung et al. 2013] foram utilizadas para guiar as características do modelo proposto. O framework Tardigrade e sua documentação associada estão disponíveis no repositório <sup>1</sup>. O *framework* foi desenvolvido para plataforma móvel Android, onde o SDK da referida plataforma serviu de fonte de recursos para as funções providas pelo *framework*. O Tardigrade se caracteriza como um *framework* do tipo caixa cinza, no qual alguns classes já são fornecidas completas e outras precisam ser implementadas ou estendidas para cada jogo instanciado. A Figura 2 ilustra os módulos principais do Tardigrade e suas relações.

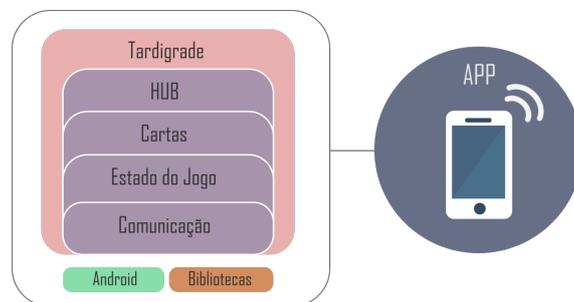


Figura 2. Organização estrutural dos módulos principais do Tardigrade.

### 3.1. Cartas Ubíquas

O Tardigrade materializa o conceito de jogos de cartas ubíquos proposto em [Cheung et al. 2013] que foi estendido em pesquisa anterior [Rodrigues et al. 2016] por meio de três categorias de cartas descritas a seguir.

**Cartas tangíveis-identificáveis** são cartas ubíquas que possuem um ou mais elementos de identificação que permitem a passagem de suas propriedades da realidade para a virtualidade. O elemento de identificação pode ser uma etiqueta NFC, um *QR Code* ou um padrão visual. Uma aplicação que reconhece as cartas do baralho de UNO e faz sua transcrição em áudio é um exemplo de uso de cartas tangíveis-identificáveis que pode criar interação de jogadores com deficiência visual com jogo sem a necessidade de modificações físicas.

**Cartas tangíveis-interativas** são aquelas que, além da identificação, possuem propriedades que variam ou são ativadas de acordo com elementos de interação (e.g., sacudir o celular) ou elementos do mundo digital (e.g., número aleatório dentro de um intervalo). Essas cartas permitem jogos com características híbridas, isto é, jogos analógicos que se misturam com o virtual, como um jogo que abre uma janela de interação onde o jogador precisa sacudir o dispositivo para obter um resultado.

Por fim, as **Cartas sensíveis ao contexto** usam a ideia de que elementos do contexto de um ou mais jogadores podem ser incorporados ao jogo para aumentar, diminuir,

<sup>1</sup><https://github.com/walmeidaw/tardigrade>

ativar ou desativar efeitos de cartas físicas. Por exemplo, em um jogo inspirado no *Yu-gi-oh!*, cartas de polimerização só poderiam ser utilizadas se o jogador estivesse em um contexto específico (e.g., o jogador deve estar sorrindo em uma manhã chuvosa). O dispositivo digital funciona como verificador dessa condição contextual (detecta o sorriso do jogador e checka a meteorologia daquela localização).

Esses três tipos de cartas apresentadas permitem o desenvolvimento de diversos jogos de cartas que podem ser entendidos como ubíquos. Uma das características que permite isso é a tangibilização de componentes reais para o meio virtual, que podem ou não fazer o uso do contexto dos jogadores como componentes do jogo. Um vídeo de execução desses conceitos de carta pode ser acessado em no link<sup>2</sup>.

### 3.2. Módulo Hub

O Hub funciona como um agregador de informações e funções externas ao *framework*, uma espécie de padrão *Facade* para o acesso a informações (e.g., dados contextuais capturados pela *Context API*). Sua responsabilidade é agrupar métodos de integração, sejam com sensores nativos, plataformas de *middleware* ou bibliotecas externas. Este módulo foi implementado seguindo o padrão *Singleton*, de forma que, com sua instância, é possível acessar o mesmo serviço de diferentes partes da aplicação.

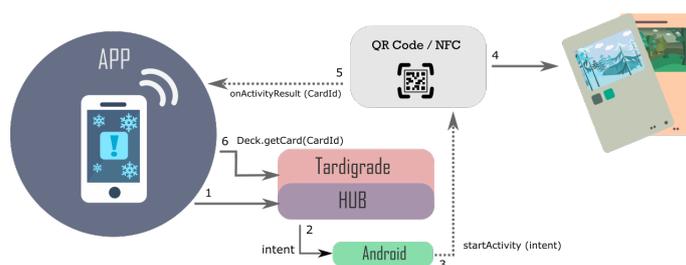


Figura 3. Fluxo do Hub para integração de serviços.

A Figura 3 apresenta o fluxo realizado na leitura de uma carta por *QR Code*. O *Hub* é quem intermedia a interação de leitura de uma carta física. O *Hub* inicia uma intenção de leitura de *QR Code* e a solicitação abre a aplicação que faz a leitura e retorna o resultado para o *Hub* que a repassa para a aplicação. O último passo consiste em tratar o resultado obtido para recuperar as informações da carta identificada. Uma outra alternativa permite a leitura de cartas de forma mais simples, utilizando etiquetas NFC, permitindo que as cartas sejam identificadas apenas aproximando-as de dispositivos compatíveis com a tecnologia.

### 3.3. Módulo de Cartas

O módulo de cartas é composto pelas classes *Deck* e *Card*. A classe *Deck* representa o baralho e é responsável por agregar e gerenciar um conjunto de objetos *Card*, podendo fornecer informações desses objetos. O *Tardigrade* pode suportar três tipos de cartas com comportamentos e utilidades distintas descritos anteriormente. Todas essas cartas podem ter efeitos de aumento de realidade, revelando sons, animações e interações gráficas que tornem a experiência de jogo mais imersiva. O comportamento dessas cartas

<sup>2</sup><https://goo.gl/dccWSJ>

é encapsulado na classe `Card` seguindo o padrão de projeto *Command*. A implementação desse comportamento é feita no método `execute()` do objeto `Card`. O comportamento ubíquo e/ou interativo tem o `Hub` como fonte, onde também acontece a interação com uma plataforma de *middleware*, biblioteca externa, sensores, ou com o serviço responsável por adquirir o contexto.

### 3.4. Módulo de Estado de Jogo e Módulo de Comunicação

O estado de um jogo no Tardigrade representa a memória do andamento e do progresso do jogo. Esse estado descreve informações do cenário, dos objetos envolvidos e dos jogadores em um dado momento do jogo. O estado possui uma estrutura única para as regras de cada jogo, ficando a cargo do desenvolvedor do jogo explicitar suas peculiaridades ao usar o *framework*. Já o Módulo de Comunicação é responsável por descobrir, conectar e gerenciar a troca de mensagens entre dispositivos conectados. Esse módulo é composto por quatro componentes: a classe `Network` que ativa um canal de recebimento de mensagens; a classe `Discovery` que implementa um serviço de rede e a descoberta de outros dispositivos que possuam o mesmo serviço ativado (integrando-se com a classe `Network` para criar uma conexão entre dispositivos); cada conexão é representada por um objeto `Channel` que envia mensagens para o canal de rede correspondente; e a classe `Manager` que implementa o padrão *Observer* e envia mensagens a um grupo de canais observadores deste objeto. Nos eventos em que um dispositivo é descoberto ou perdido, o conteúdo do objeto de mensagem é um objeto `Channel` que pode ser utilizado para manter comunicação com o dispositivo encontrado. O serviço de descoberta disponível no *framework* limita-se a encontrar dispositivos que estejam conectados numa mesma WLAN.

## 4. Avaliação Inicial com Desenvolvedores

Foi realizada uma avaliação inicial do *framework* Tardigrade e suas funcionalidades para o desenvolvimento de jogos de cartas ubíquos, tendo como foco verificar se desenvolvedores com alguma experiência em aplicações Android conseguiriam utilizar o *framework* e os seus componentes disponíveis com facilidade. O experimento de avaliação foi inspirado nas metodologias de avaliação de usabilidade propostas por [Brad A. Myers 2016]

### 4.1. Perfil dos Usuários

Foi escolhido de forma não probabilística, e por conveniência, um grupo de oito voluntários (um graduando, cinco mestrandos e dois doutorandos) com experiência prévia em desenvolvimento Android, onde apenas um tinha algum conhecimento no desenvolvimento de jogos digitais. A Figura 4 apresenta mais informações sobre o perfil do grupo de avaliadores. Sobre o perfil de programador dos avaliadores, todos reportaram ter conhecimento relevante em programação Java e desenvolvimento de aplicações Android.

### 4.2. Procedimento

Para cada um dos voluntários, foi solicitado executar uma sequência de atividades e o preenchimento de formulários de avaliação. O tempo que cada um dos voluntários levou para realizar as tarefas foi coletado durante cada avaliação.

A aplicação do experimento foi dividida em quatro módulos. Cada um deles consistia na capacitação sobre as principais funcionalidades do *framework* seguido por uma

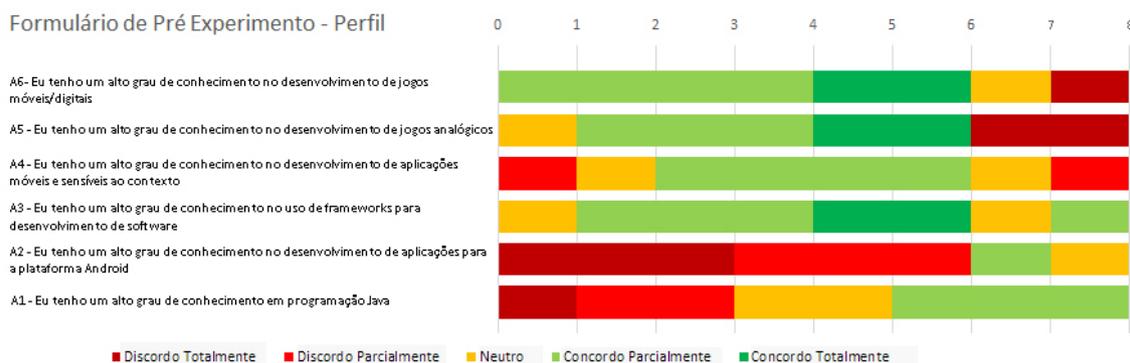


Figura 4. Perfil dos voluntários.

atividade relacionada às características elencadas no módulo em avaliação. As atividades consistiam em codificar os módulos conforme as instruções, sendo que a atividade era considerada como concluída quando o voluntário autodeclarava que as instruções dadas tinham sido codificadas por completo. O conjunto realizado tinha como objetivo criar uma aplicação simples que fizesse uso dos componentes do *framework*.

Ao fim de cada atividade, o comportamento esperado do jogo implementado era validado como relatado a seguir. A **Atividade 1** deveria reconhecer uma carta e apresentar seu valor na interface. A **Atividade 2** deveria reconhecer uma carta identificável, uma carta interativa e uma carta sensível ao contexto, apresentando o resultado baseado na interação realizada na interface. A **Atividade 3** deveria somar o valor das cartas reconhecidas ao estado do jogo e apresentar a soma na tela. Por fim, a **Atividade 4** deveria realizar a descoberta de um segundo dispositivo, conectar-se a ele e trocar informações de cartas reconhecidas, devendo apresentar a soma das cartas em ambos os dispositivos.

### 4.3. Materiais e Instrumentos

Durante o experimento, foram utilizados seis formulários digitais<sup>3</sup> criados no *Google Forms* para realizar a coleta das informações durante as etapas da avaliação.

O formulário de pré-experimento tinha como objetivo coletar informações sobre o perfil dos voluntários. Ao fim de cada atividade, era aplicado um formulário de acompanhamento com o objetivo de saber quanto o voluntário compreendeu da atividade e das dificuldades encontradas. O formulário pós-experimento tinha como objetivo coletar informações sobre a utilização do *framework* e foi inspirado no SUS (*System Usability Scale*) [Brooke 2013]. Os desenvolvedores deviam responder usando a escala Likert de cinco níveis: “discordo totalmente”, “discordo parcialmente”, “indiferente” ou “neutro”, “concordo parcialmente” e “concordo totalmente”. O objetivo desse formulário é obter uma avaliação preliminar a cerca do uso do *framework*. Para o experimento, foi criado um documento de capacitação dos voluntários para o uso do *framework*. O documento<sup>4</sup> apresenta o conceito de jogos de cartas ubíquos, a utilização das funcionalidades do *framework* e as atividades a serem realizadas. Para desenvolver as atividades, um ambiente de desenvolvimento no *Android Studio* foi preparado com um projeto base já configurado. Esse projeto consistia no *framework* Tardigrade já implantado com um projeto de

<sup>3</sup>Os formulários estão disponíveis em <https://sites.google.com/site/avaliacaotardigrade/>

<sup>4</sup>O documento está disponível em <https://sites.google.com/site/avaliacaotardigrade/>

interface já pronta. Assim o experimento consistia apenas no que tangia as funcionalidades do *framework*. Os voluntários tinham disponível um dispositivo Android para testar a aplicação ao fim de cada atividade. Na última atividade, relacionada a descoberta de dispositivos, um segundo dispositivo era exigido para completar o teste. Para este fim, foi fornecido um outro dispositivo, com uma versão completa e funcional da aplicação já previamente instalada.

#### 4.4. Resultados e Discussão

Os voluntários de 1 ao 6 executaram cada tarefa em duas etapas, uma de capacitação e outra para desenvolver a atividade. Os voluntários tiveram o maior gasto de tempo na capacitação do primeiro módulo (média de 10 minutos), que incluía também uma introdução aos jogos ubíquos, uma visão geral do *framework* para então abordar o módulo de reconhecimento de cartas de forma detalhada. Nos demais módulos, a média foi de quatro minutos e meio para cada um dos três módulos finais. Já para desenvolver as instruções das atividades, eles gastaram em média 11 minutos e meio.

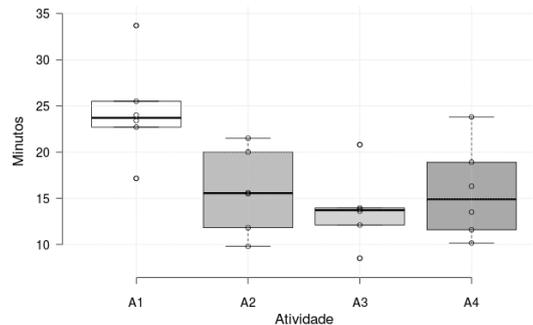


Figura 5. Medições da primeira sessão (voluntários 1 à 6)

O experimento foi feito com mais quatro voluntários. Apenas dois, considerados como voluntários 7 e 8, para fins de análise, puderam concluir o experimento. Os demais tiveram de se ausentar devido compromissos. Essa sessão do experimento foi feita de forma conjunta e sem uma etapa de capacitação separada para cada módulo. Os voluntários receberam o mesmo documento com a descrição do *framework*, módulos e atividades. O tempo total para a conclusão foi superior ao dos demais voluntários. Assim como nos demais testes, foi observado um tempo maior na execução da tarefa referente ao primeiro módulo. Uma voluntária que não pode concluir o experimento sofreu com atrasos adicionais para a conclusão do módulo 1 devido à problemas no ambiente de desenvolvimento. Ela foi capaz de executar a tarefa do módulo 2 em apenas 11 minutos, tempo inferior ao de todos os demais candidatos. Para os 3 módulos restantes, a média foi de 29 minutos. Foi possível avaliar o tempo de execução da tarefa sem uma etapa prévia de capacitação. Para concluir cada módulo com sucesso, era necessário que o voluntário extraísse as informações pertinentes à atividade, da descrição do módulo respectivo.

O primeiro módulo tinha o objetivo de avaliar o reconhecimento de cartas. Apesar de cinco dos voluntários afirmarem terem compreendido a atividade, sete deles apresentaram problemas no desenvolvimento deste módulo, principalmente em relação ao uso da classe `Card`. Já o voluntário 4 não teve dificuldades em implementar a atividade e não apresentou problemas na compilação. Ele teve um tempo total de 17 minutos no módulo

1, uma vez que o tempo médio dos voluntários 1 ao 6 foi de 24 minutos. Os voluntários 7 e 8 tiveram um tempo médio de 45 minutos para finalizar o primeiro módulo.

O segundo módulo tinha como foco os diferentes comportamentos das cartas reconhecidas. Dentre os voluntários 1 à 6, a média de tempo finalização do módulo foi de 16 minutos. Para os voluntários 7 e 8, essa média foi de 30 minutos. Diferente da primeira atividade, os voluntários não tiveram problemas para executar as instruções que desenvolviam o comportamento das cartas, com exceção do voluntário 7 que informou não ter compreendido o que devia ser feito no passo a do item 1, os voluntários 2, 3 e 6 tiveram mais dificuldades na abstração das instruções e precisaram de orientações durante esta etapa do experimento, necessitando de mais tempo para concluí-la. Os voluntários 1, 4 e 5 levaram aproximadamente metade do tempo para finalizar as mesmas instruções.

O módulo 3 desenvolveu um estado de jogo e a interação de cartas com esse estado. Com tempo médio total para os voluntários 1 à 6 de 13 minutos e de 26 minutos para os voluntários 7 e 8, todos tiveram sucesso ao realizar todas as instruções. O voluntário 4 apresentando novamente um tempo consideravelmente inferior aos demais (apenas oito minutos e 27 segundos).

O quarto módulo tinha como propósito apresentar a implementação da comunicação. Neste módulo, os voluntários 1 à 6 demonstraram menos dificuldade na compreensão do seu funcionamento. Com uma média de 15 minutos e 46 segundos, todos foram capazes de finalizar a atividade com eficiência. Os voluntários 7 e 8, concluíram em 22 e 40 minutos, respectivamente. Em ambos os casos apenas a parte de descoberta do dispositivo não funcionou como esperado. A tarefa foi dada como concluída devido ao tempo já estendido. A Figura 6 apresenta os resultados obtidos com o formulário de avaliação pós-experimento.

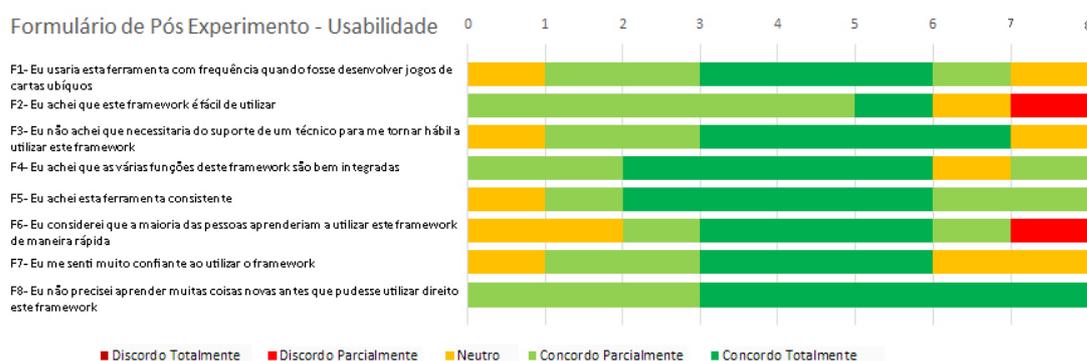


Figura 6. Resultado do questionário pós-experimento.

## 5. Conclusões e Trabalhos Futuros

Este trabalho apresentou o Tardigrade, um *framework* Android para o desenvolvimento de jogos de carta com características de aplicações ubíquas. A concepção de um *framework* para jogos de carta ubíquos encapsula conhecimento deste domínio e proporciona uma base para o desenvolvimento mais rápido desses jogos. O Tardigrade implementa três tipos de cartas que podem ser utilizadas como mecânicas que envolvem interações com o

mundo real dos jogadores. Um procedimento de experimentação do *framework* foi realizado com um grupo de 8 desenvolvedores para verificar a qualidade das funcionalidades oferecidas, a usabilidade dessas funcionalidades e a facilidade no desenvolvimento de novos jogos ubíquos. Com os resultados da experimentação, foi possível observar que o *framework* foi bem aceito pelos desenvolvedores. No entanto, apesar da avaliação positiva, é importante mencionar que a experimentação foi realizada com um baixo número de desenvolvedores, e que o desenvolvimento de um único jogo de carta ubíquo não é o suficiente para evidenciar a completude do Tardigrade. Além disso, a subjetividade das respostas e a relação próxima entre os candidatos e avaliadores, pode ser um fator de ameaça ou validade da abrangência para os resultados. Para minimizar esse elemento, foram coletados os tempos durante cada atividade. Assim, é possível comparar as impressões coletadas nos formulários com os resultados práticos. Essas limitações estabelecem possíveis problemas a serem explorados em trabalhos futuros.

## Referências

- Barhaug, M. (2017). Serious games: improving the learning effect with hybrid games. Master's thesis.
- Brad A. Myers, J. S. (2016). Improving api usability. *Magazine Communications of the ACM*.
- Brooke, J. (2013). Sus: A retrospective. *J. Usability Studies*, 8(2):29–40.
- Cheung, G., Lee, A., Cheng, K., and Lee, H. J. (2013). Dispensable, tweakable, and tangible components: Supporting socially negotiated gameplay. *Games and Culture*.
- Montola, M., Stenros, J., and Waern, A. (2009). *Pervasive Games: Theory and Design*. Morgan Kaufmann Publishers Inc.
- Paavilainen, J., Korhonen, H., Alha, K., Stenros, J., Koskinen, E., and Mayra, F. (2017). The pokémon go experience: A location-based augmented reality mobile game goes mainstream. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2493–2498, New York, NY, USA. ACM.
- Rodrigues, W., Mendonça, G., and Viana, W. (2016). Cartas sensíveis ao contexto: Aumentando a experiência em jogos de carta ubíquos. In *SBC – Proceedings of SBGames*, pages 630–638.
- Römer, K. and Domnitcheva, S. (2002). Smart playing cards: A ubiquitous computing game. *Personal Ubiquitous Comput.*, 6(5-6):371–377.
- Sakamoto, M., Alexandrova, T., and Nakajima, T. (2014). Introducing virtuality to enhance game-related physical artifacts. *International Journal of Smart Home*, 8(2):137–152.
- Silva, L. F. M., Noletto, C., Lima, M., Ferreira, C., Marinho, C., Viana, W., and Trinta, F. (2017). LAGARTO: A location based games authoring tool enhanced with augmented reality features. *Entertainment Computing*, 22:3–13.
- Spínola, R. O. and Travassos, G. H. (2012). Towards a framework to characterize ubiquitous software projects. *Inf. Softw. Technol.*, 54(7):759–785.
- Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11.

# Caracterização de topologia de Redes Veiculares baseada em Teoria da Informação

Cristopher G. S. Freitas<sup>1</sup>, Heitor S. Ramos<sup>1</sup>, Raquel S. Cabral<sup>3</sup>, Osvaldo A. Rosso<sup>2</sup>,  
André L. L. Aquino<sup>1</sup>

<sup>1</sup>Instituto de Computação, Universidade Federal de Alagoas

<sup>2</sup>Instituto de Física, Universidade Federal de Alagoas

<sup>3</sup>Curso de Ciência da Computação, Arapiraca, Universidade Federal de Alagoas

cgsf@ic.ufal.br, heitor@laccan.ufal.br, raquel.cabral@gmail.com  
oarosso@gmail.com, alla@laccan.ufal.br

**Abstract.** *Vehicular networks can be studied using vehicle's behavior individually varying with time, characterized by displacement or velocities. However, on this work we analyze the aggregated graph-based representation, which describes a global aspect of the network, encapsulating the dynamics of each vehicle during a sampled period, thus, verifying its structural behavior with Information Theory quantifiers for mapping these data onto a Complexity-Entropy plane. This method was applied to 17 vehicular networks, varying synthetically its topologies in V2V, V2I and V2V2I, such way its graphs presented a variable dynamic between Watts-Strogatz and Barabási-Albert models behaviors.*

**Resumo.** *Redes Veiculares podem ser estudadas utilizando o comportamento individual de cada veículo em relação ao tempo, caracterizados pelo deslocamento ou velocidade. No entanto, neste trabalho iremos analisar o comportamento do grafo agregado, que descreve a rede em um aspecto global, encapsulando toda a dinâmica dos veículos durante o intervalo total amostrado, assim, verificando seus aspectos estruturais com quantificadores de Teoria da Informação para mapear esses dados no plano Complexidade-Entropia. Este método foi aplicado à 17 redes veiculares, variando suas topologias em V2V, V2I e V2V2I, de forma que seus grafos agregados apresentaram uma dinâmica variável entre o comportamento dos modelos Watts-Strogatz e Barabási-Albert.*

## 1. Introdução

As Redes Veiculares podem ser representadas por um grafo  $G(V, E)$ , em que os veículos ou infraestruturas representam um nó  $v_i \in V$ , e cada nó possui um equipamento de comunicação sem fio que ao conectar-se com outro veículo, constitui uma aresta  $e_i \in E$ . O objetivo de uma VANET (*Vehicular Ad-hoc Network*) é manter a comunicação entre veículos tornando a interação entre eles possível, nesse caso, cada veículo é capaz de enviar e receber dados na rede [Yousefi et al. 2006]. Existem diferentes arquiteturas para a comunicação entre veículos em uma VANET [Liang et al. 2015], dentre elas, temos V2V, V2I e V2V2I. Em uma rede V2V (*Vehicle-To-Vehicle*) os veículos comunicam-se entre si, sem a necessidade de um intermediário; em uma abordagem V2I (*Vehicle-To-Infrastructure*) existe uma infraestrutura responsável por entrega e recebimento das men-

sagens entre veículos; e na abordagem híbrida, denominada V2V2I (*Vehicle-To-Vehicle-To-Infrastructure*), os nós comunicam-se tanto entre si quanto com as infraestruturas disponíveis.

Essas redes são amplamente estudadas com o objetivo de otimizar a conectividade nas vias ao redor do mundo [Hajlaoui et al. 2016], de forma que a disseminação de informação pode ajudar nos aspectos de segurança e eficiência no trânsito. Com isso, entender as características da topologia da Rede Veicular é muito relevante para a escolha de qual abordagem de roteamento utilizar na rede [Liu et al. 2016], caso os veículos tenham informações da posição dos demais nós, pode-se variar o protocolo de roteamento visando melhorar o processo de disseminação de dados de acordo com a dinâmica atual da rede [Zhang et al. 2013].

Redes Veiculares em geral são muito dinâmicas, variando sua topologia constantemente ao longo do tempo, assim, alguns estudos caracterizam a dinâmica dessas redes em busca de identificar padrões de comportamento que possam ajudar na tomada de decisões para projetar e manter essas redes; [Aquino et al. 2015] caracterizam o comportamento de cada veículo individualmente, utilizando séries temporais de velocidade com o método proposto por [Rosso et al. 2007], verificando a variação das velocidades em diferentes intervalos de amostragem; [Tang et al. 2016, Yan et al. 2017] caracterizam a dinâmica do fluxo de tráfego multivariado baseado nas séries temporais de deslocamento, utilizando a teoria de Redes Complexas, analisando métricas como coeficiente de agrupamento e distribuição do grau, identificando os limites onde há transição no espaço de fase, de forma que o tráfego muda de livre para congestionado ao decorrer do dia; [Pappalardo et al. 2013] estudam os padrões de mobilidade humana com carros utilizando séries temporais de deslocamento, avaliando o comportamento local de cada veículo de acordo com as leis de mobilidade, introduzindo métodos para monitoramento de tráfego em tempo real.

Neste trabalho, considera-se o aspecto global do grafo agregado da rede veicular, utilizando a teoria de Redes Complexas para caracterizar a estrutura e dinâmica da rede baseada em sua topologia. O grafo agregado representa todas as conexões realizadas com um veículo em um intervalo de tempo amostrado, onde cada conexão torna-se uma aresta que persiste por todo intervalo, criando uma representação global da rede veicular. Neste sentido, o problema tratado neste trabalho é:

*Qual comportamento de uma rede veicular, modelada como um grafo agregado, ao avaliarmos a estrutura da rede e dinâmica dos nós usando as conexões V2V, V2I e V2V2I?*

A análise das redes utilizadas neste trabalho será composta principalmente por dois quantificadores de Teoria da Informação: **Entropia de Shannon  $S$**  e a **Complexidade Estatística  $C$** , de forma que suas aplicações levarão em consideração uma distribuição de probabilidade  $P$  denominada *Random Walk Betweenness* [Newman 2005], utilizada para caracterizar as redes reais e o modelo de referência [Erdos 1959] no cálculo do divergente descrito na Seção 2.4.  $S$  é capaz de medir a desordem em um sistema, já  $C$  quantifica a estrutura do conteúdo deste sistema, onde podemos aplicar esses quantificadores à Redes Complexas utilizando o método proposto por [Wiedermann et al. 2017].

**Tabela 1: Redes Veiculares utilizadas para caracterização em sua topologia V2V. Sendo  $N$  o número de nós, a probabilidade de conexão  $\rho$ , Entropia da Rede  $S$  e a Complexidade Estatística  $C$ .**

| Descrição da Rede Veicular      | $N$   | $\rho$ | $S$   | $C$   |
|---------------------------------|-------|--------|-------|-------|
| Anaheim                         | 416   | 0.007  | 0.172 | 0.166 |
| Austin                          | 7387  | 0.000  | 0.112 | 0.056 |
| Barcelona                       | 930   | 0.004  | 0.191 | 0.188 |
| Berlin-Center                   | 12981 | 0.000  | 0.119 | 0.059 |
| Berlin-Friedrichshain           | 224   | 0.015  | 0.210 | 0.203 |
| Berlin-Mitte-Center             | 397   | 0.008  | 0.184 | 0.178 |
| Berlin-Mitte-Prenz-Fried-Center | 974   | 0.003  | 0.164 | 0.159 |
| Berlin-Prenzlauerberg-Center    | 352   | 0.009  | 0.182 | 0.176 |
| Berlin-Tiergarten               | 359   | 0.009  | 0.194 | 0.189 |
| Chicago-Regional                | 12979 | 0.000  | 0.112 | 0.056 |
| Chicago-Sketch                  | 933   | 0.003  | 0.129 | 0.125 |
| Eastern-Massachusetts           | 72    | 0.049  | 0.261 | 0.248 |
| Hessen-Asymmetric               | 4660  | 0.001  | 0.103 | 0.103 |
| Philadelphia                    | 13389 | 0.000  | 0.120 | 0.060 |
| Terrassa-Asymmetric             | 1603  | 0.002  | 0.137 | 0.134 |
| Winnipeg-Asymmetric             | 948   | 0.003  | 0.148 | 0.144 |
| Winnipeg                        | 1040  | 0.003  | 0.154 | 0.150 |

Com essas ferramentas, buscamos caracterizar as topologias de Redes Veiculares V2V, V2I e V2V2I, de acordo com seu comportamento apresentado no plano Complexidade-Entropia. Aliando métricas de Redes Complexas, tais como coeficiente de agrupamento e distância média para garantir a robustez do modelo quanto sua capacidade de caracterizar as redes reais. As demais seções deste trabalho, organizam-se em: Seção 2 que está subdivida em Seção 2.1 que descreve as redes utilizadas no trabalho, Seção 2.2 discute a formalidade para definição das redes e Seção 2.3 e 2.4 apresentando os quantificadores e sua metodologia de utilização. Seção 3 discute os resultados e a Seção 4 encerra com as devidas conclusões.

## 2. Materiais e Métodos

### 2.1. Base de dados

Neste trabalho utilizamos 17 redes reais com dados já pré-processados para caracterização das topologias, todas essas redes estão disponíveis no repositório criado pelo grupo *Research Core Team*<sup>1</sup>. Cada base possui dados sobre o fluxo da rede, viagens individuais de cada veículo e seu grafo agregado no período amostrado. Consideramos que essas bases, por apresentarem os dados individuais dos nós veículos, seguem uma organização V2V. Na tabela 1 estão os detalhes sobre a quantidade de nós, arestas e probabilidade de conexão  $\rho$  de cada rede.

### 2.2. Definições das Redes

Para cada rede consideramos um grafo  $G(V, E)$  representado por uma matriz de adjacência  $\mathbf{A}$  com dimensões  $N \times N$ , sendo  $N$  o número de nós presentes na rede, onde

<sup>1</sup><https://github.com/bstabler/TransportationNetworks>

$A_{ij} = 1$  caso exista uma conexão entre os nós  $i$  e  $j$ , caso contrário,  $A_{ij} = 0$ . Utilizando um grafo sem laços, não direcionado e sem pesos, temos que a diagonal principal da matriz,  $A_{ii} = 0 \forall i, \dots, N$ ,  $\mathbf{A} = \mathbf{A}^T$ . O grau de cada nó  $k_i$  pode ser calculado com  $k_i = \sum_{n=j}^N A_{ij}$ ,  $k_i \leq N - 1$ . Para nossa análise com Entropia e Complexidade de acordo com a abordagem de [Wiedermann et al. 2017], utilizamos uma distribuição uniforme  $P_i$  denominada *Random Walk Betweenness* [Newman 2005], tal que:

$$p_{i \rightarrow j} = \frac{A_{ij}}{k_i} \in \{0, 1/k_i\}, \quad (1)$$

sendo  $p_{i \rightarrow j}$  a probabilidade do nó  $i$  saltar aleatoriamente para o nó  $j$ .

O *Betweenness Centrality* mede a centralidade de um nó na rede, verificando quantos caminhos mínimos entre um par de nós na rede transversam um nó  $i$ . Já o *Random Walk Betweenness* dá ênfase aos caminhos aleatórios que transversam um nó, verificando quão frequente dois outros nós transitam por ele.

### 2.3. Entropia da Rede

A Entropia da Rede  $S$  é calculada com a Entropia clássica de Shannon para distribuições discretas:

$$S(P) = - \sum_k p_k \log p_k. \quad (2)$$

A Entropia clássica de Shannon mede a desordem de um sistema, considerando a probabilidade do sistema apresentar um estado  $k$ . Levando em consideração a distribuição de probabilidade utilizada (1), para cada nó  $i$  temos uma distribuição de probabilidade  $P_i$ , consequentemente teremos uma Entropia para cada nó que será  $S_i$ , dada por

$$S_i(P_i) = - \sum_j \frac{A_{ij}}{k_i} \log \frac{A_{ij}}{k_i} = \log k_i \quad (3)$$

No caso de um nó isolado  $i$  com grau  $k_i = 0$ , define-se que  $S_i = 0$ . Após calculada a Entropia de Shannon para cada nó, encontra-se a média normalizada que caracteriza  $S$ :

$$S(P) = \frac{1}{N \log(N-1)} \sum_i S_i(P_i) \in [0, 1]. \quad (4)$$

A Entropia  $S$  pode ser interpretada de acordo com as características *Random Walk Betweenness*. Para uma rede totalmente conectada, onde  $p_{i \rightarrow j} = \frac{1}{N-1} \forall i \neq j$ , os caminhos tornam-se totalmente aleatórios, assim o nó  $i$  pode saltar para qualquer nó  $j$ , consequentemente a Entropia é maximizada e temos que  $S(P) = 1$ . Em uma rede esparsa, o caminho torna-se mais determinístico por limitar as possibilidades, de maneira que podemos prever com maior frequência para qual nó  $j$  será realizado o salto. Assim, quanto mais esparsa for a rede, menor será a Entropia, assim,  $S \rightarrow 0$ .

## 2.4. Complexidade Estatística

Em busca de compreender a estrutura física do sistema complexo, a Complexidade Estatística  $C$  mede o desequilíbrio do sistema baseado no conteúdo da informação descrita pela distribuição de probabilidade  $P$  e sua Entropia  $S$  [Lopez-Ruiz et al. 1995]. O desequilíbrio do sistema na mecânica estatística é medido através do divergente de *Jenson-Shannon*

$$Q_i(P_i, P_{i,e}) = Q_0 \{S_i(0.5[P_i + P_{i,e}]) - 0.5[S_i(P_i) + S_i(P_{i,e})]\} \quad (5)$$

com  $Q_0 = 1/\log 2$  para garantir  $Q_i \in [0, 1]$ . A distribuição  $P_{i,e}$  denota a mesma que  $P_i$  para uma apropriada referência ou estado de equilíbrio, neste caso, uma rede. Nesta abordagem, assume-se que o estado de equilíbrio do sistema corresponde ao totalmente aleatório.

Para uma rede complexa, utilizaremos o modelo *Erdős-Rényi* como referência para descrever o estado totalmente aleatório. Para satisfazer a equação 5, é necessário comparar as redes reais aos modelos aleatórios, ou seja, calcular  $P_{i,e}$ . Com o objetivo de garantir a aleatoriedade, para cada rede real utilizada são geradas 100 redes usando o modelo de *Erdős-Rényi*, com o mesmo número de nós  $N$  e probabilidade de conexão  $\rho = \sum_i k_i / (N(N - 1))$ , assim, para cada rede de referência criada, computamos  $P_{i,e}$  e  $Q_i$ , a média de  $Q_i$ , e por fim,  $Q$ .

$$Q_i(P_i, P_{i,e}) = \frac{1}{100} \sum_{e=1}^{100} Q_e(P_i, P_{i,e}), \quad (6)$$

$$Q(P, P_e) = \frac{1}{N} \sum_{i=1}^N Q_i(P_i, P_{i,e}). \quad (7)$$

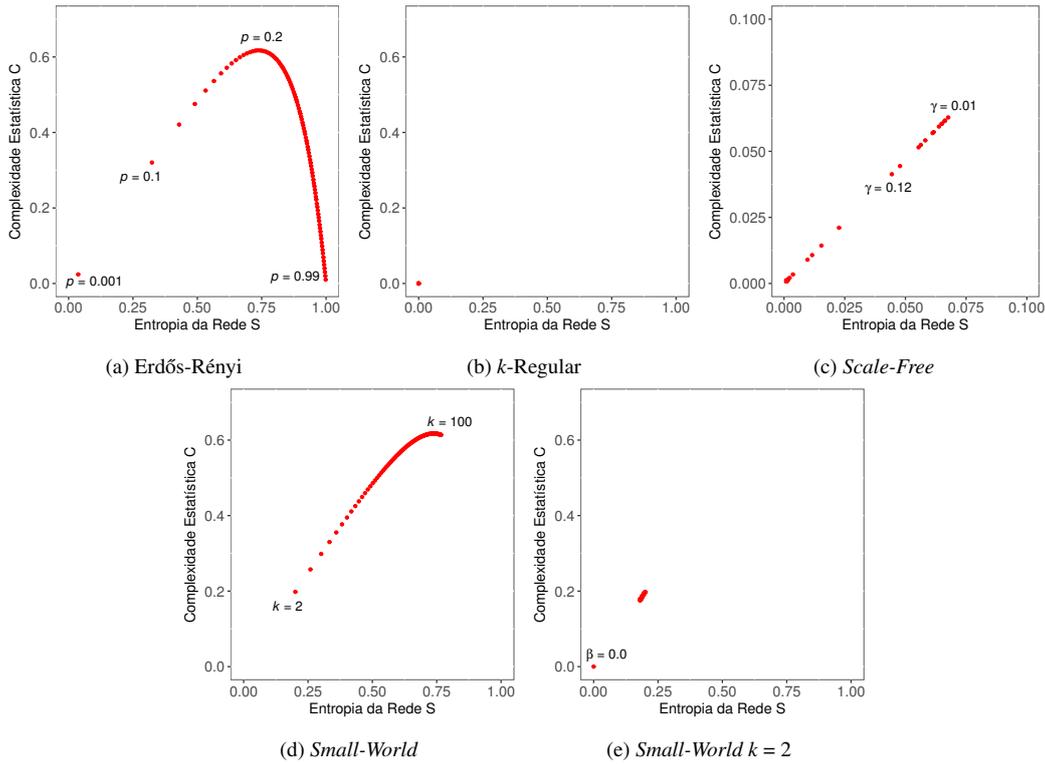
Em conformidade com a rede de referência apropriada, uma rede totalmente conectada implica  $P = P_e$  possuindo um desequilíbrio mínimo e, assim,  $Q = 0$ . Já uma rede esparsa, as probabilidades  $P$  e  $P_e$  diferem, o que indicará a presença de um alto desequilíbrio, daí,  $Q \gg 0$ . Observa-se que não faz sentido atribuir uma alta Complexidade a uma rede muito esparsa ou uma rede totalmente conectada, dessa forma, a utilização do divergente  $Q$  ou da Entropia  $S$  por si só, não é suficiente, com isso, a Complexidade Estatística foi proposta utilizando o produto dessas duas quantidades [Rosso et al. 2007].

$$C(P) = Q(P, P_e)S(P) \in [0, 1]. \quad (8)$$

Observamos que para o caso de uma rede muito esparsa,  $S(P) = 0$  e  $C(P) = 0$ . Já no caso de uma rede totalmente conectada  $S(P) = 1$ , segue-se que  $P = P_e$  que implica  $Q(P, P_e) = 0$ , assim,  $C(P) = 0$ . Com isso, temos a formação do plano Complexidade-Entropia que pode ser utilizado para caracterização de Redes Complexas.

## 3. Resultados e Discussão

Com o propósito de verificar a consistência do modelo em sua capacidade de discernir Redes Complexas de acordo com sua topologia, foram avaliados grafos de Erdős-Rényi [Erdos 1959], *Small World* [Watts and Strogatz 1998], *Scale-Free* [Barabási and Albert 1999] e  $k$ -Regular, como ponto de partida para termos uma referência sólida dos seus comportamentos no plano  $S \times C$ , assim, comparamos aos resultados apresentados pelas redes veiculares, com variações na organização da rede em V2V, V2I e V2V2I.

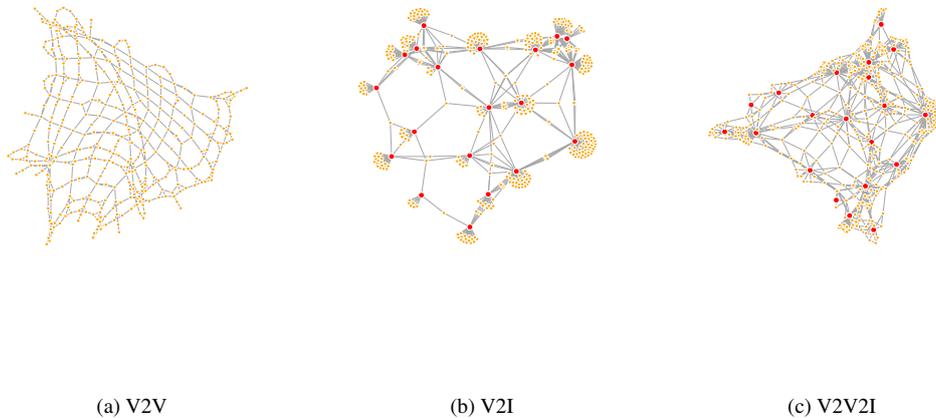


**Figura 1:** Na figura (a), temos os resultados do Modelo Erdős-Rényi com  $N = 1000$  e uma variação a probabilidade de conexão  $\rho$ . Em (b), observamos o comportamento do grafo  $k$ -regular no plano para qualquer  $k$ . A figura (c) apresenta os resultados do modelo *Scale Free* de acordo com a variação de  $\gamma$ . Em (d) e (e), verificamos as características do grafo *Small-World* variando o grau médio  $k$  e a probabilidade de religação  $\beta$ .

### 3.1. Redes Sintéticas

Para gerar as redes de Erdős-Rényi utilizamos  $N = 1000$  e variamos sua probabilidade de conexão  $p$ , tal que,  $p \in [0, 1]$ . Como pode ser visto na Figura 1a, a medida que  $p$  aumenta, a Entropia segue crescendo, no entanto, a Complexidade apresenta um máximo em  $p = 0.2$  e a partir desse ponto vai diminuindo bruscamente. Em  $p = 0.99$ , temos que a Entropia  $S \rightarrow 1$  e a  $C \rightarrow 0$ , o que sugere que temos uma rede muito conectada, porém com uma estrutura aleatória de acordo com os conceitos apresentados na Seção 2.4. Já em um grafo  $k$ -regular,  $S = 0$  e  $C = 0$  para qualquer  $k$ , como ilustrado na Figura 1b, onde foram gerados grafos regulares com  $N = 1000$  e  $k \in [0, N - 1]$  e não houve nenhuma alteração em seus resultados.

As redes *Scale-Free* foram geradas de acordo com o modelo [Barabási and Albert 1999], onde sua distribuição do grau  $P(k)$  segue a lei da potência:  $P(k) \sim k^{-\gamma}$ , tal que  $\gamma \in [0, 10]$  e  $N = 1000$ . A variação de  $\gamma$  determina a velocidade do decaimento da distribuição do grau, de forma que quanto maior, maior a chance de termos grandes componentes conectados a maioria dos nós, aproximando-se de um grafo regular ou geométrico, assim, à medida que  $\gamma \rightarrow \infty$ ,  $S \rightarrow 0$ . Verificando os resultados apresentados na Figura 1c, notamos que, em geral, os grafos correspondentes às redes *Scale-Free* apresentam  $S < 0.07$  e  $C < 0.07$ , com uma transição mais brusca em  $\gamma = 0.12$ .



**Figura 2: Representação da topologia da rede Anaheim com  $N = 416$ .**

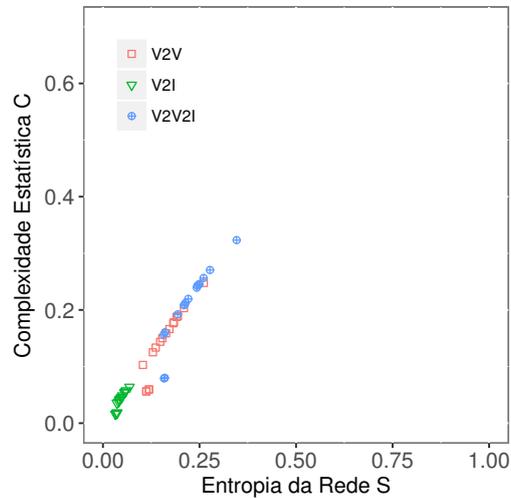
De acordo com os resultados apresentados na Figura 1d, podemos analisar o comportamento transitório das Redes *Small-World*. Com a variação do grau médio da rede inicial  $k \in [2, 100]$ , temos uma mudança de característica na dinâmica da topologia da rede. Levando em consideração o modelo de criação do grafo *Small-World*, onde  $k$  define o grau de cada nó, no grafo  $k$ -regular inicial, de onde partem as religações entre as arestas de acordo com a probabilidade de religação  $\beta$ . Contudo, a influência de  $\beta$  nos resultados, não é tão relevante quanto o grau médio, analisamos  $\beta \in [0, 1]$ , com  $\beta = 0$  e  $k = 2$ , nos resta apenas um grafo regular como apresentado na Figura 1e, já para os demais valores de  $\beta$ , as características da rede *Small-World* no plano de Complexidade e Entropia não apresentam variações tão relevantes em relação aos demais.

### 3.2. Redes Veiculares

Para as 17 redes reais descritas na Seção 2.1, realizamos o pré-processamento na rede de forma a adaptar de acordo com as definições da rede na Seção 2.2, removendo os laços e considerando uma rede não-direcionada. Calculando os quantificadores de Teoria da Informação, mapeamos os resultados no plano Complexidade-Entropia, onde essas redes na sua topologia original são consideradas redes V2V. Observando os resultados de uma rede V2V no plano de acordo com a Figura 3, identificamos um comportamento próximo das redes *Small-World* descrito na Figura 1e, mas com uma tendência de transição, aproximando-se de uma rede *Scale-Free*.

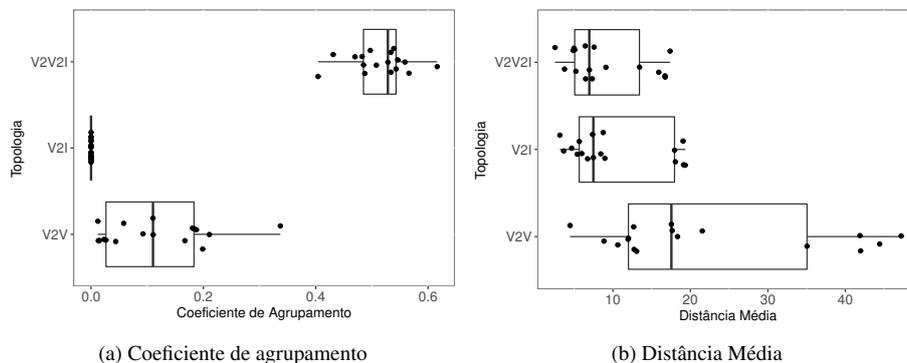
Com propósito de analisar a transição na topologia das redes veiculares, utilizamos os dados reais das redes, que caracterizam uma topologia V2V ilustrada na Figura 2a, para simular topologias V2I e V2V2I, observando a variação das características da rede à medida que modificamos a estrutura da rede real. Para criar uma rede com conexões V2I, foram removidas todas as conexões entre veículos, 5% de nós de foram escolhidos de forma aleatória na rede e estabelecidos como infraestruturas. A determinação de escolher 5% dos nós como infraestruturas foi empírica, baseando-se na suposição que o número de infraestruturas presentes em uma VANET é muito menor que o número de veículos em geral. Então, para cada veículo escolhemos qual infraestrutura encontra-se mais próxima e estabelecemos a conexão V2I. Caso hajam duas infraestruturas com a mesma distância

mínima, permite-se a conexão do veículo às duas, assumindo que em um cenário global, há grandes chances do veículo transitar entre o alcance de ambas.



**Figura 3: Plano Complexidade-Entropia com resultado da análise de todas as redes veiculares na topologia V2V, V2I e V2V2I.**

Seus resultados no plano Complexidade-Entropia estão na Figura 3 e a estrutura da rede está ilustrada na Figura 2b. Em comparação aos resultados apresentados anteriormente, nos indicam uma semelhança das características da rede *Scale Free* com as Redes Veiculares em uma topologia V2I. Considerando o cenário V2V2I, as conexões existentes nas Redes Veiculares entre veículos foram mantidas, apenas adicionando as infraestruturas, do mesmo modo que em V2I (Figura 2c), conectando os nós veículos aos nós mais próximos que foram selecionados aleatoriamente como infraestruturas. Analisando o plano na Figura 3, observamos a transição das características das Redes Veiculares V2V2I com valores mais próximos do modelo *Small-World*, distanciando-se do *Scale-Free*, que em comparação aos resultados da Figura 1d, podemos notar que as redes V2V2I possuem um comportamento próximo das Redes *Small-World* com  $k$  entre 2 e 10.



**Figura 4: Diagrama de caixa descrevendo a variação do Coeficiente de Agrupamento (a) e Distância Média (b) de acordo com a mudança de topologia para cada Rede Veicular.**

Para uma verificação mais detalhada desse comportamento, recorremos a métricas clássicas de redes complexas como **Coefficiente de Agrupamento** e **Distância Média**, onde redes *Watts-Strogatz* apresentam um baixo valor de distância média e um alto coeficiente de agrupamento. Podemos observar nas Figuras 4 e 4a que a mudança de topologia  $V2V \rightarrow V2I$ , remove a presença de grupos, anulando o coeficiente de agrupamento. Da mesma forma, em  $V2V \rightarrow V2V2I$ , observamos que o coeficiente de agrupamento aumenta e a distância média na rede diminui, como ilustrado na Figura 4b, indicando um comportamento *Small-World* na rede. Diante disso, por indução, supomos que o aumento do número de infraestruturas na rede deve aumentar a presença de agrupamentos e diminuir a distância média ainda mais, já que teremos uma maior quantidade de nós intermediários com um alto valor de  $k$  aumentando a conectividade da rede.

#### 4. Conclusão

Na abordagem de Teoria da Informação apresentada, conseguimos utilizar um método bem sucedido em caracterizar diferentes tipos de redes reais, com o propósito de caracterizar Redes Veiculares no contexto dos modelos de Redes Complexas. Assim, observamos que Redes Veiculares com uma topologia  $V2I$ , apresentam estruturas semelhantes ao grafo *Scale-Free*, com coeficiente de agrupamento nulo, onde as infraestruturas possuem um comportamento de *hub*. Já as topologias  $V2V$  e  $V2V2I$ , são melhores descritas pelo modelo *Small-World*, com algumas variações na sua estrutura, tendo em vista que a presença de infraestruturas induzem a existência de *hubs* na rede, consequentemente temos um coeficiente de agrupamento maior em  $V2V2I$  e uma distância média menor.

A identificação das características dessas topologias são relevantes para definição de protocolos de roteamento e disseminação de dados adequadamente, redes com alto coeficiente de agrupamento indicam uma maior tolerância à falhas, por possuir menos chance de apresentar *links* críticos capazes de conectar ou desconectar grandes componentes, e maior probabilidade de um nó encontrar um caminho alternativo a um nó vizinho caso a conexão existente seja extinta. Uma baixa distância média aponta para uma maior facilidade de um nó se comunicar com outro nó qualquer nessa rede, dessa forma, a disseminação de dados na rede pode ocorrer de forma mais rápida. Daí, podemos concluir que a topologia  $V2V2I$  é mais robusta em relação à  $V2V$  e  $V2I$ , sendo mais capaz de lidar com falhas e de distribuir a informação de forma mais eficiente na rede. Para os próximos trabalhos, devemos analisar o impacto mais detalhado do aumento da quantidade de infraestruturas na rede e aprofundar na variação das características ao longo do tempo, com base em dados de deslocamento e velocidade, em busca de uma caracterização mais abrangente utilizando este método em comparação as demais abordagens existentes na literatura.

#### Referências

- [Aquino et al. 2015] Aquino, A. L., Cavalcante, T. S., Almeida, E. S., Frery, A. C., and Rosso, O. A. (2015). Characterization of vehicle behavior with information theory. *The European Physical Journal B*, 88(10):257.
- [Barabási and Albert 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439):509–512.
- [Erdos 1959] Erdos, P. (1959). On random graphs. *Publicationes mathematicae*, 6:290–297.

- [Hajlaoui et al. 2016] Hajlaoui, R., Guyennet, H., and Moulahi, T. (2016). A survey on heuristic-based routing methods in vehicular ad-hoc network: Technical challenges and future trends. *IEEE Sensors Journal*, 16(17):6782–6792.
- [Liang et al. 2015] Liang, W., Li, Z., Zhang, H., Wang, S., and Bie, R. (2015). Vehicular ad hoc networks: architectures, research issues, methodologies, challenges, and trends. *International Journal of Distributed Sensor Networks*, 11(8):745303.
- [Liu et al. 2016] Liu, J., Wan, J., Wang, Q., Deng, P., Zhou, K., and Qiao, Y. (2016). A survey on position-based routing for vehicular ad hoc networks. *Telecommunication Systems*, 62(1):15–30.
- [Lopez-Ruiz et al. 1995] Lopez-Ruiz, R., Mancini, H. L., and Calbet, X. (1995). A statistical measure of complexity. *Physics Letters A*, 209(5-6):321–326.
- [Newman 2005] Newman, M. E. (2005). A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54.
- [Pappalardo et al. 2013] Pappalardo, L., Rinzivillo, S., Qu, Z., Pedreschi, D., and Giannotti, F. (2013). Understanding the patterns of car travel. *The European Physical Journal Special Topics*, 215(1):61–73.
- [Rosso et al. 2007] Rosso, O., Larrondo, H., Martin, M., Plastino, A., and Fuentes, M. (2007). Distinguishing noise from chaos. *Physical review letters*, 99(15):154102.
- [Tang et al. 2016] Tang, J., Liu, F., Zhang, W., Zhang, S., and Wang, Y. (2016). Exploring dynamic property of traffic flow time series in multi-states based on complex networks: Phase space reconstruction versus visibility graph. *Physica A: Statistical Mechanics and its Applications*, 450:635–648.
- [Watts and Strogatz 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440.
- [Wiedermann et al. 2017] Wiedermann, M., Donges, J. F., Kurths, J., and Donner, R. V. (2017). Mapping and discrimination of networks in the complexity-entropy plane. *Physical Review E*, 96(4):042304.
- [Yan et al. 2017] Yan, Y., Zhang, S., Tang, J., and Wang, X. (2017). Understanding characteristics in multivariate traffic flow time series from complex network structure. *Physica A: Statistical Mechanics and its Applications*, 477:149–160.
- [Yousefi et al. 2006] Yousefi, S., Mousavi, M. S., and Fathy, M. (2006). Vehicular ad hoc networks (vanets): challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766. IEEE.
- [Zhang et al. 2013] Zhang, D., Huang, H., Zhou, J., Xia, F., and Chen, Z. (2013). Detecting hot road mobility of vehicular ad hoc networks. *Mobile Networks and Applications*, 18(6):803–813.

# Construção do Estado da Arte em Estratégias Híbridas para Raciocínio de Contexto: uma Abordagem explorando Revisão Sistemática da Literatura

Roger S. Machado<sup>1</sup>, Ricardo B. Almeida<sup>1</sup>, João Ladislau Lopes<sup>2</sup>,  
Ana Marilza Pernas<sup>1</sup>, Adenauer C. Yamin<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Computação (PPGC)  
Universidade Federal de Pelotas (UFPel)

<sup>2</sup>Instituto Federal Sul-rio-grandense (IFSul)

{rdsmachado, rbalmeida, marilza, adenauer}@inf.ufpel.edu.br

joaolopes@cavg.ifsul.edu.br

**Abstract.** *Hybrid proposals for context reasoning stand out in the literature, these proposals allow the use of two or more reasoning strategies. Motivated by this scenario, this paper aims to build the state of the art, systematizing research trends in Context Awareness. A Systematic Literature Review is explored, contributing to the identification of relevant works in the area, as well as the specification of criteria for its selection. In this review, we analyze works published between 2012 and 2017. During the review process, 2952 papers were identified, and at the end, seven works that proposed hybrid strategies were selected. The selected works are presented, discussed, and a comparison between them is performed.*

**Resumo.** *As propostas híbridas para raciocínio de contexto vêm ganhando destaque na literatura, as mesmas permitem a utilização de duas ou mais estratégias de raciocínio. Motivado por este cenário, esse artigo tem o objetivo de construir o estado da arte, sistematizando tendências de pesquisa em Ciência de Contexto. É explorada uma Revisão Sistemática da Literatura, contribuindo tanto pela identificação de trabalhos relevantes na área, bem como, pela especificação de critérios para sua seleção. Nessa revisão são analisados trabalhos publicados entre 2012 e 2017. Durante o processo de revisão foram identificados 2952 trabalhos, sendo selecionados ao final sete trabalhos que propõem estratégias híbridas. Os trabalhos selecionados são apresentados, discutidos, e uma comparação entre eles é realizada.*

## 1. Introdução

Devido ao rápido desenvolvimento da IoT (*Internet of Things*) enquanto uma estratégia de ubiquidade, e sua utilização em diferentes aspectos da vida cotidiana, observa-se um aumento na quantidade de dados de contexto que continuamente são gerados, os quais representam o estado do ambiente ao longo do tempo. Acredita-se que a utilização efetiva deste grande volume de dados de contexto pode introduzir novas possibilidades para o desenvolvimento humano, com base em soluções computacionais que sejam autônomas, responsivas e adaptativas ao contexto [Li et al. 2015].

Desta forma, a ciência de contexto torna-se uma abordagem chave para fornecer serviços adaptáveis, por exemplo, quando os serviços mais adequados são selecionados de acordo com as informações de contexto relevantes para o usuário, ou quando é necessário alterar os parâmetros operacionais dos serviços, em função do contexto, durante sua execução. Deste modo, as aplicações cientes de contexto devem ser capazes de adaptar seus comportamentos em mudança com um mínimo de intervenção humana, e para tanto introduzem desafios diversos para os seus desenvolvedores [Khattak et al. 2014].

Os sistemas cientes de contexto na sua operação envolvem uma grande quantidade de informações de contexto que precisam ser: recuperadas constantemente; efetivamente interpretadas; rapidamente processadas; divulgadas às aplicações interessadas; e mantidas em repositórios de contexto [Bellavista et al. 2012]. Portanto, para minimizar os esforços de desenvolvimento deste tipo de aplicação, precisam ser adotados mecanismos robustos para gerenciamento de contexto, bem como, esquemas de representação de contexto que possibilitem o tratamento da heterogeneidade dos dados envolvidos.

Investigar o gerenciamento de contexto, em termos de abordagens para modelagem de informações de contexto e estratégias de raciocínio para processamento das informações de contextuais, constitui parte de um significativo esforço de pesquisa. Além disso, é necessário investigar o emprego da ciência de contexto para o desenvolvimento de aplicações que são adaptáveis segundo os interesses definidos pelos usuários [Bibri 2015].

Uma série de abordagens de modelagem e estratégias para raciocínio de contexto foram recentemente desenvolvidas, no entanto, as mesmas não se mostram versáteis para atender, individualmente, os requisitos de raciocínio para as diferentes aplicações ubíquas, no cenário da IoT. Assim, identifica-se a necessidade de utilização de diferentes métodos para raciocínio de contexto, constituindo assim, as estratégias híbridas [Li et al. 2015].

O uso de estratégias híbridas para raciocínio é uma área de pesquisa em crescimento na comunidade devido a sua possibilidade de maior precisão preditiva. Ele fornece as abordagens adequadas para a solução baseada em computador considerando problemas complexos, como os que são enfrentados pelas aplicações ubíquas [Subbaraj and Venkataraman 2016].

O objetivo deste artigo é construir o estado da arte na área de ciência de contexto, empregando estratégias híbridas para raciocínio. Para tanto, é realizada uma Revisão Sistemática da Literatura (RSL), cujos procedimentos necessários para sua consecução são discutidos, facultando sua reconstrução com outros critérios. Como produto, foram identificados trabalhos representativos na área no período entre 2012 e 2017, sendo registrados os retornos obtidos nas diferentes bases de indexação. Ao final da RSL, foram selecionados sete trabalhos que propõem estratégias híbridas para raciocínio de contexto. Os trabalhos selecionados são apresentados e discutidos, sendo, ainda, realizada uma comparação entre eles.

Este artigo está dividido da seguinte forma: a Seção 2 discute a RSL realizada; a Seção 3 apresenta os trabalhos selecionados durante revisão, onde são analisadas as principais características de cada trabalho; e, por fim, a Seção 4 discute as considerações finais e os trabalhos futuros.

## 2. Trabalhos que propõem estratégias híbridas para raciocínio de contexto

A RSL é uma metodologia de pesquisa que prevê a execução de uma série de procedimentos para gerar uma revisão de literatura em uma área, ou seja, identificar um conjunto de artigos que traz uma prospecção criteriosa do tema de pesquisa abordado [Kitchenham and Charters 2007]. Uma característica desta metodologia de pesquisa é que todos os seus procedimentos são registrados, permitindo que o estudo realizado seja reproduzível por outros pesquisadores.

### 2.1. Questões de pesquisa

A primeira etapa para o desenvolvimento de uma RSL é a definição das questões de pesquisa que fundamentam a mesma [Petersen et al. 2015]. Desta forma, as questões de pesquisa que fundamentam essa RSL e nortearam o estudo são:

RQ1: Quais estratégias para raciocínio de contexto são utilizadas em conjunto?

RQ2: Em quais domínios são aplicadas as estratégias híbridas para raciocínio?

RQ3: Que metodologia é empregada na avaliação?

### 2.2. Processo de busca

O processo de busca pelos artigos seguiu o fluxo de execução apresentado na figura 1, onde pode ser visualizado que o processo possuiu quatro etapas. A primeira etapa é a aplicação da string de busca nas bases de indexação selecionadas. Na segunda etapa é realizado o primeiro filtro, o qual elimina os trabalhos publicados antes de 2012, analisando assim somente trabalhos dos últimos cinco anos, desta forma, buscando os trabalhos publicados entre 2012 e 2017. A terceira etapa é responsável por realizar um filtro com base na string de busca, analisando o título, resumo e palavra-chave dos trabalhos. A última etapa é responsável por analisar todo o texto, tendo como base os critérios de inclusão e exclusão, sendo selecionados somente os artigos que propõem estratégias híbridas para raciocínio de contexto.

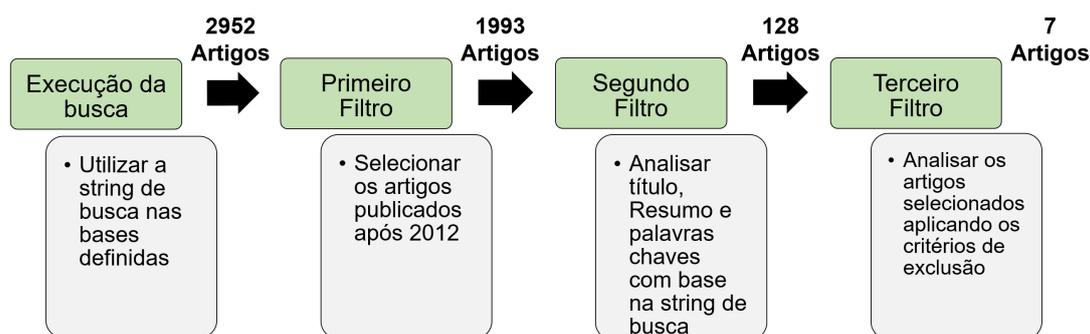


Figura 1. Fluxo do processo de revisão sistemática.

A string de busca utilizada no processo de revisão sistemática da literatura foi: ((“**hybrid reasoning**”) OR (“**hybrid approach**”) OR (“**hybrid framework**”) OR (“**hybrid strategy**”)) AND ( (“**context aware**”) OR (“**context awareness**”)).

A fim de encontrar os estudos relevantes foi aplicada a string de busca em seis bases de indexação: (i) ACM; (ii) IEEE; (iii) Science Direct; (iv) Scopus; (v) Springer; (vi) Web of Science. A escolha das bases de indexação se deu devido a sua importância, bem como, por cobrirem a maioria dos periódicos e artigos de conferências publicados no campo da ciência da computação, com foco em aplicações cientes de contexto.

### 2.3. Critérios de exclusão

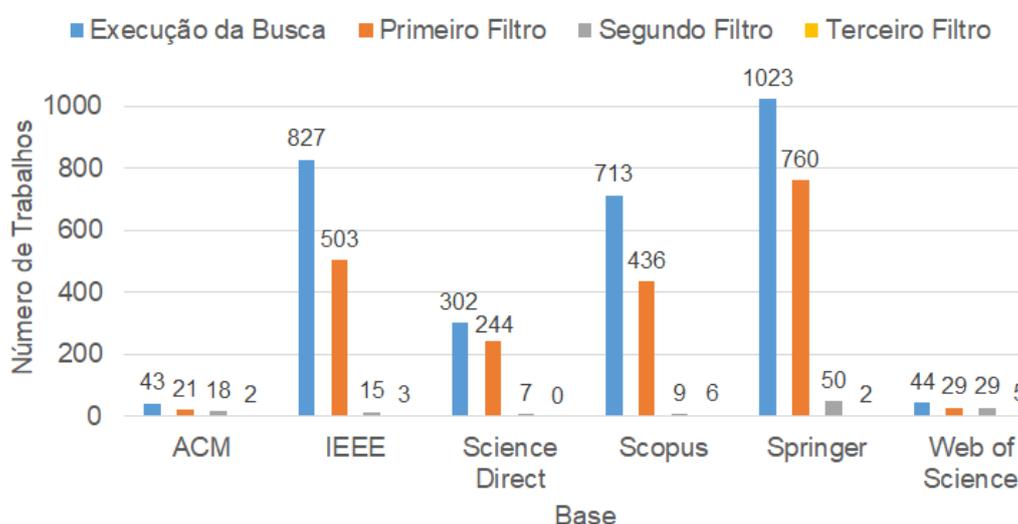
Os filtros aplicados nesta RSL seguem os seguintes critérios de exclusão:

- trabalhos publicados antes de 2012;
- trabalhos que não contêm a string de busca no título, resumo ou palavras-chave;
- trabalhos que não são artigos publicados em conferências ou periódicos;
- trabalhos que não estão escritos em inglês ou português;
- trabalhos que não apresentam estratégias para raciocínio;
- trabalhos que propõem estratégias para raciocínio que não são híbridas;
- trabalhos que registram pequenas diferenças de outro artigo.

O último critério de exclusão foi utilizado para excluir os trabalhos preliminares de uma mesma pesquisa, sendo selecionado somente o trabalho mais atual. Para aplicação desse critério foi analisado todo o texto dos artigos, verificando se os artigos tratavam da mesma pesquisa, registrando apenas pequenas diferenças entre eles.

### 2.4. Resultados da revisão sistemática

A figura 2 apresenta uma visão geral dos artigos retornados por cada base de indexação durante o processo de revisão sistemática empregado. Destaca-se que os artigos duplicados estão contabilizados na figura.



**Figura 2. Artigos retornados em cada base de indexação.**

A tabela 1 apresenta uma visão geral do número de artigos que foram excluídos com base em cada critério de exclusão. Conforme pode ser visualizado, o critério que excluiu o maior número de trabalhos foi de analisar o título, resumo e palavra-chave tendo como base a string de busca, sendo excluídos 1865 artigos.

### 3. Trabalhos selecionados na RSL realizada

Após a realização da RSL apresentada anteriormente, foram selecionados 7 artigos, os quais são apresentados a seguir, sendo explorados aspectos referentes a sua estratégia para raciocínio, bem como, suas principais funcionalidades.

**Tabela 1. Número de artigos excluídos por cada critério de exclusão.**

| Critério de Exclusão   | Número de Artigos Excluídos |
|--|-----------------------------|
| Trabalhos publicados antes de 2012   | 959                         |
| Trabalhos que não contêm a string de busca no título, resumo ou palavras-chave | 1865                        |
| Trabalhos que não são artigos publicados em conferências ou periódicos         | 10                          |
| Trabalhos que não são escritos em inglês ou português                          | 2                           |
| Trabalhos que não apresentam estratégias para raciocínio de contexto           | 24                          |
| Trabalhos que propõem estratégias para raciocínio que não são híbridas         | 11                          |
| Trabalhos que apresentam pequenas modificações de outro artigo                 | 4                           |

### 3.1. Raciocínio baseado em regras e em ontologias

O artigo de [Lopes et al. 2012] propõe uma arquitetura para ciência do contexto, a qual tem como premissa fundamental a possibilidade de interpretação e a composição dinâmica das informações adquiridas por sensores posicionados em diferentes localidades do ambiente ubíquo. Com isso, a arquitetura busca viabilizar a construção de contextos complexos, cujas informações contextuais possam ser obtidas de forma distribuída, bem como ter suas regras de processamento alteradas sem a necessidade de interromper a execução das aplicações.

Para realização do processamento das informações contextuais é proposta uma estratégia híbrida para raciocínio a qual utiliza um raciocínio semântico provido pela ontologia, com base em axiomas e regras para inferir novos conhecimentos baseadas nas instâncias da ontologia. Além deste, é proposta a utilização de um raciocínio baseado em regras do tipo ECA (Evento-Condição-Ação), o qual realiza o raciocínio de informações contextuais providas por um modelo relacional.

### 3.2. Raciocínio baseado em ontologias e em casos

Em [Strobbe et al. 2012] é proposta uma plataforma que tem como principais características ser ciente de contexto e tratar da agregação e da abstração de informações de contexto usando ontologias para representar as mesmas. A plataforma CASP utiliza na etapa de processamento de contexto dois raciocinadores, um baseado em regras ontológicas e o outro em casos.

Para viabilizar o raciocínio híbrido, são exploradas combinações dos raciocinadores baseados em regras e em casos. As informações de contexto de alto nível podem ser derivadas usando regras, casos ou uma combinação de ambos. As regras ontológicas representam conhecimento geral do domínio, enquanto os casos capturam conhecimento específico. O raciocínio baseado em regras avalia as regras fornecidas e adiciona as informações derivadas à base de conhecimento. O raciocínio baseado em casos recupera as situações semelhantes e usa as soluções correspondentes para atualizar as informações na base de conhecimento representado pelo modelo ontológico.

### 3.3. Raciocínio baseado em regras fuzzy e em casos

O artigo [Yuan and Herbert 2014] apresenta um *framework* para raciocínio híbrido desenvolvido para o sistema CARA (*Context-Aware Real-time Assistant*). O *framework* realiza

o monitoramento de um ambiente doméstico inteligente, fornecendo uma fusão de dados cientes de contexto, bem como mecanismos de detecção de anomalias que suportam a atividade de análise e geração de alertas.

O raciocínio híbrido é baseado em regras fuzzy e em casos, permitindo tornar o CARA mais robusto e adaptável a um ambiente sujeito a mudança. Emprega-se o uso do raciocínio baseado em casos para detectar anomalias condicionais para automação residencial e o uso do raciocínio baseado em regras difusas hierárquicas para lidar com exceções, bem como, para recuperação de casos e adaptação de casos sensíveis à consulta.

### **3.4. Raciocínio baseado em aprendizagem de máquina**

Em [Smaaberg et al. 2014] é apresentado um protótipo de um sistema de recomendação de concertos para grupos de usuários. O protótipo é ciente de contexto e leva em conta a localização e a temporalidade do usuário ao dar recomendações. Para realizar o raciocínio das informações contextuais é proposta uma estratégia híbrida a qual implementa dois algoritmos, os quais aproveitam as informações históricas dos usuários, um algoritmo de linhagem colaborativa (*K-Nearest Neighbor*) e um algoritmo de fatoração de matriz. Os algoritmos podem ser utilizados de forma individual ou combinada.

A usabilidade do protótipo foi avaliada usando a Escala de Usabilidade do Sistema e uma avaliação centrada no usuário foi realizada para avaliar a qualidade das recomendações. O objetivo do protótipo era recomendar concertos para um usuário dentro da localização e de um tempo específico, com o intuito de oferecer recomendações que os usuários poderiam ter interesse em participar. Os resultados da avaliação de usabilidade mostram que os usuários ficaram satisfeitos com a usabilidade do protótipo. Os resultados da Avaliação da Qualidade mostram que a abordagem que combina os dois algoritmos obteve o melhor resultado. O protótipo foi testado e os usuários ficaram satisfeitos com a qualidade das recomendações.

### **3.5. Raciocínio baseado em aprendizagem de máquina e lógica probabilística**

No artigo [Baz et al. 2016] é proposto um sistema que classifica os produtos de varejo em uma prateleira de loja. O sistema de classificação proposto combina os pontos fortes de classificadores sem contexto e informações de contexto. O objetivo do sistema é projetar um modelo probabilístico que codifique as relações entre os produtos na prateleira e combine isso com os métodos atuais de classificação de imagem baseada em visão.

São propostos duas estratégias híbridas diferentes, a primeira estratégia híbrida combina SVMs (*Support Vector Machines*) e um modelo gráfico generativo que tenta explicitamente modelar uma distribuição de probabilidade conjunta, com base em HMMs (*Hidden Markov Models*). Na segunda estratégia, os SVMs são combinados com uma abordagem discriminativa baseada em CRFs (*Conditional Random Fields*) tendo como intuito formar um novo contexto. Os modelos probabilísticos são treinados aprendendo com os erros do classificador livre de contexto (SVMs) e as relações vizinhas entre os produtos de varejo.

### **3.6. Raciocínio baseado em ontologia e aprendizagem de máquina**

Em [Razzaq et al. 2017] é proposta uma estratégia híbrida para raciocínio de contexto para um *framework* que oferece suporte personalizado para saúde e bem-estar. Para realizar o raciocínio das informações contextuais é proposta uma nova abordagem conjuntiva,

a qual é baseada em ontologia e aprendizagem de máquina. A estratégia baseada em aprendizagem de máquina suporta a inferência classificando contextos de alto nível com base no conjunto de dados instanciado na ontologia.

Para realização do armazenamento dos dados de contexto providos pela ontologia os autores propõem a utilização do modelo de triplas. Nos testes realizados foi combinada uma estratégia baseada em ontologia com seis diferentes técnicas de aprendizagem de máquina: *Naive Bayes*, *KStar*, *IBK*, *J48*, *RandomForest* e *RandomTree*. A técnica que combinada com a ontologia apresentou o melhor resultado foi a *RandomForest*, enquanto que a *KStar* apresentou o pior resultado.

### 3.7. Raciocínio baseado em ontologias e lógica probabilística

No artigo [Li et al. 2017] é proposta uma estratégia híbrida para o raciocínio de contexto para um *framework* de ciência de contexto. A estratégia proposta é constituída de três técnicas de raciocínio: (i) ontológico, sendo utilizados axiomas internos à ontologia com o intuito de inferir conhecimento relacionado as classes presentes na mesma; (ii) baseado em regras, que consiste de regras que processam os dados contextuais presentes na ontologia, sendo utilizada a linguagem SWRL (*Semantic Web Rule Language*) para a concepção das regras; (iii) *Multi-Entity Bayesian Network*, o qual consiste de um sistema lógico que integra a lógica da primeira ordem com a teoria da probabilidade bayesiana.

Os autores destacam que, com a incorporação das três técnicas, é proporcionado um melhor desempenho para o raciocínio de contexto, aproveitando-se dos pontos positivos de cada técnica, com o intuito de compensar as fraquezas das mesmas. Os autores destacam que a estratégia híbrida proposta é eficaz para fornecer diferentes capacidades de raciocínio e satisfazer as diferentes necessidades das aplicações [Li et al. 2017].

### 3.8. Discussão dos trabalhos selecionados na RSL

A tabela 2 apresenta uma análise comparativa entre os trabalhos selecionados na RSL realizada. Esta análise é realizada com base nos seguintes critérios: (i) os tipos de sensores empregados na etapa de aquisição; (ii) os modelos utilizados para realizar a representação do contexto; (iii) as técnicas presentes para realizar o processamento; (iv) a forma de armazenamento dos dados contextuais; e (v) algumas ferramentas utilizadas para o desenvolvimento do trabalho.

Com base na tabela 2 é constatado que na etapa de aquisição apenas dois trabalhos não deixam claro como fazem a aquisição dos dados de contexto. Enquanto os outros trabalhos analisados possuem suporte para lidar com diferentes tipos de sensores, assim permitindo os mesmos atender à aplicações de diferentes naturezas.

Analisando a etapa de modelagem, pode ser visualizado que quatro trabalhos utilizam o modelo baseado em ontologias, principalmente devido a sua característica de representação semântica. Destaca-se ainda que em [Lopes et al. 2012] foi proposta a utilização de uma modelagem híbrida combinando modelos ontológicos e relacionais.

Em relação ao armazenamento das informações contextuais, nota-se que a maioria dos trabalhos não discute informações referentes a esta funcionalidade, apesar da mesma ser importante para as aplicações cientes de contexto. O trabalho de [Yuan and Herbert 2014] utiliza o modelo baseado em linguagem de marcação, empregando a linguagem XML (*eXtensible Markup Language*), principalmente pela facilidade

**Tabela 2. Comparação entre os trabalhos selecionados.**

| Trabalhos                                  | Aquisição                       | Modelo                          | Processamento  | Armazenamento                   | Ferramentas  |
|--|---------------------------------|---------------------------------|--|---------------------------------|--|
| LOPES et al., 2012                         | Sensores<br>Físicos<br>Lógicos  | Ontológico e<br>Relacional      | Regras e<br>Regras Ontológicas   | Relacional                      | Java, Jena, OSGI,<br>Protégé, Pellet,<br>PostgreSQL    |
| STROBBE et al.,<br>2012                    | Sensores<br>Físicos<br>Lógicos  | Ontológico                      | Regras Ontológicas e<br>Baseado em Casos   | Informação<br>Não<br>Disponível | Java, Jena, OWL API,<br>jCOLIBRI                       |
| YUAN; HERBERT,<br>2014                     | Sensores<br>Físicos<br>Lógicos  | Fuzzy                           | Regras Fuzzy e<br>Baseado em Casos   | XML                             | jCOLIBRI   |
| SMAABERG;<br>SHABIB; KROGSTIE,<br>J., 2014 | Informação<br>Não<br>Disponível | Informação<br>Não<br>Disponível | Aprendizagem de<br>Máquina (KNN e<br>Fatoração de Matriz)  | Informação<br>Não<br>Disponível | JavaScript, HTML5,<br>Durandal.js, Model<br>View       |
| BAZ; YORUK;<br>CETIN, 2016                 | Informação<br>Não<br>Disponível | Informação<br>Não<br>Disponível | Aprendizagem de<br>Máquina e Lógica<br>Probabilística  | Informação<br>Não<br>Disponível | VLFEAT toolbox   |
| RAZZAQ; AMIN;<br>LEE, 2017                 | Sensores<br>Físicos<br>Lógicos  | Ontológico                      | Aprendizagem de<br>Máquina e Axiomas<br>Ontológicos  | Jena TDB (triple<br>store)      | Java, Maven, Jena,<br>Protégé, Pellet,<br>OWL-API Weka |
| LI; MARTÁNEZ;<br>RUBIO, 2017               | Sensores<br>Físicos<br>Lógicos  | Ontológico                      | Regras Ontológicas,<br>Axiomas Ontológicos e<br>Lógica Integrada com<br>Probabilidade<br>Bayesiana | Informação<br>Não<br>Disponível | Pellet   |

da mesma ser utilizada para transmitir as informações entre diferentes componentes. Enquanto o trabalho de [Lopes et al. 2012] emprega um modelo relacional de armazenamento, o qual não é considerado um modelo satisfatório para o armazenamento de dados providos pelo modelo ontológico [Can et al. 2017]. Além destes, o trabalho de [Razzaque et al. 2016] propõe a utilização do modelo de triplas, sendo um modelo mais propício para realizar a persistência das informações providas por ontologias.

Acredita-se serem necessários mecanismos com diferentes características para armazenamento de contexto, já que as aplicações cientes de contexto atuais tendem a lidar com diferentes tipos de dados, níveis semânticos entre outros aspectos. Por sua vez, a eficiência destes mecanismos é um aspecto significativo, considerando a escalabilidade do volume de dados contextuais envolvido, sua constante atualização ou consulta.

No que diz respeito ao raciocínio de contexto, os trabalhos apresentados apesar de utilizarem estratégias híbridas, os mesmos não permitem as diferentes aplicações comporem as estratégias que serão utilizadas em função dos dados contextuais envolvidos. Desta forma, nota-se a ausência de uma abordagem que permita a combinação das diferentes estratégias para o raciocínio de contexto. Onde a criação desta abordagem poderá aumentar a flexibilidade para as aplicações utilizarem os contextos de interesse, como também facilitar a identificação das situações de interesse.

Com base na análise dos artigos selecionados na RSL realizada é possível responder as questões de pesquisa que fundamentaram esta revisão. As questões de pesquisa e as repostas adquiridas são:

**RQ1: Quais estratégias para raciocínio de contexto são utilizadas em conjunto?**

Foi possível observar que não existe um padrão para efetivação das combinações das estratégias, por exemplo, estratégias baseadas em ontologias são usadas com regras, casos e aprendizagem de máquina. Ressalta-se que o trabalho de [Smaaberg et al. 2014] combina três técnicas diferentes, entretanto todas baseadas em aprendizagem de máquina.

**RQ2: Em quais domínios são aplicadas as estratégias híbridas para raciocínio?** Dentre os sete trabalhos selecionados, três deles [Lopes et al. 2012, Strobbé et al. 2012, Yuan and Herbert 2014] se caracterizam por poderem ser aplicados em diferentes domínios. Enquanto os outros trabalhos [Smaaberg et al. 2014, Baz et al. 2016, Razzaq et al. 2017, Li et al. 2017] são aplicados respectivamente em: sistemas de recomendação; sistemas de classificação; paradigma de saúde e bem estar; robôs subaquáticos.

**RQ3: Que metodologia é empregada na avaliação?** Os trabalhos utilizaram uma metodologia baseada em cenários de uso para a sua avaliação. Dentre os trabalhos que propuseram a utilização técnicas de aprendizagem de máquina, dois deles [Smaaberg et al. 2014, Baz et al. 2016] utilizaram uma base de dados para realização da avaliação, enquanto [Razzaq et al. 2017] realizou a coleta de dados de 20 usuários, gerando sua própria base de dados.

#### 4. Considerações finais

Este artigo construiu o estado da arte, sistematizando tendências de pesquisa em Ciência de Contexto empregando estratégias híbridas para raciocínio. Para tanto, é explorada uma Revisão Sistemática da Literatura, contribuindo tanto pela identificação de trabalhos relevantes na área, bem como, pela especificação de critérios para sua seleção. Nesta revisão foram analisados trabalhos publicados nos últimos cinco anos, analisando os trabalhos entre 2012 e 2017. Durante o processo de revisão foram identificados 2952 trabalhos, sendo selecionados ao final sete trabalhos que propõem estratégias híbridas. Os trabalhos selecionados são apresentados, discutidos, e uma comparação entre eles é realizada.

Na comparação realizada foram discutidos os recursos oferecidos pelos trabalhos para tratar a aquisição, modelagem, processamento e armazenamento de informações contextuais, bem como, as ferramentas utilizadas em sua prototipação.

Como trabalho futuro destaca-se a concepção de um *framework* flexível para provimento da ciência de contexto para as aplicações ubíquas. Este *framework* a ser concebido deverá ser flexível no que diz respeito aos recursos oferecidos, disponibilizando uma abordagem para raciocínio de contexto que permita uma composição dinâmica de diferentes estratégias.

#### Agradecimentos

O presente trabalho foi realizado com apoio da CAPES (Programa Nacional de Cooperação Acadêmica - Procad) e da FAPERGS (Programa Pesquisador Gaúcho - PqG). Roger S. Machado é Bolsista FAPERGS/CAPES - BRASIL, nível doutorado.

## Referências

- Baz, I., Yoruk, E., and Cetin, M. (2016). Context-aware hybrid classification system for fine-grained retail product recognition. In *IEEE 12th IVMSPP*, France.
- Bellavista, P., Corradi, A., Fanelli, M., and Foschini, L. (2012). A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.*, 44(4):24:1–24:45.
- Bibri, S. E. (2015). Context modeling, representation, and reasoning: An ontological and hybrid approach. In *The Human Face of Ambient Intelligence*, volume 9 of *Atlantis Ambient and Pervasive Intelligence*, pages 197–257. Atlantis Press, Paris.
- Can, O., Sezer, E., Bursa, O., and Unalir, M. O. (2017). Comparing relational and ontological triple stores in healthcare domain. *Entropy*, 19(1):30.
- Khattak, A. M., Akbar, N., Aazam, M., Ali, T., Khan, A. M., Jeon, S., Hwang, M., and Lee, S. (2014). Context representation and fusion: Advancements and opportunities. *Sensors*, 14(6):9628–9668.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Joint report, Keele University and Durham University.
- Li, X., Eckert, M., Martinez, J.-F., and Rubio, G. (2015). Context aware middleware architectures: Survey and challenges. *Sensors*, 15(8):20570.
- Li, X., Martínez, J.-F., and Rubio, G. (2017). Towards a hybrid approach to context reasoning for underwater robots. *Applied Sciences*, 7(2).
- Lopes, J., Souza, R., Gadotti, G., Gusmão, M., Costa, C., Barbosa, J., Yamin, A., and Geyer, C. (2012). A distributed architecture for dynamic contexts composition in ubi-comp. In *38th CLEI 2012 - Conference Proceedings*, Colombia.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64(Supplement C):1 – 18.
- Razzaq, M. A., Amin, M. B., and Lee, S. (2017). An ontology-based hybrid approach for accurate context reasoning. In *19th APNOMS*, pages 403–406, South Korea.
- Razzaque, M., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016). Middleware for internet of things: A survey. *Internet of Things Journal, IEEE*, 3(1):70–95.
- Smaaberg, S., Shabib, N., and Krogstie, J. (2014). A user-study on context-aware group recommendation for concerts. In *CEUR Workshop Proceedings - SP 2014*, Chile.
- Strobbe, M., Van Laere, O., Dhoedt, B., De Turck, F., and Demeester, P. (2012). Hybrid reasoning technique for improving context-aware applications. *Knowledge and Information Systems*, 31(3):581–616.
- Subbaraj, R. and Venkataraman, N. (2016). Reasoning in context aware computing – a review. *International Journal of Pharmacy and Technology*, 8:5021–5032.
- Yuan, B. and Herbert, J. (2014). Context-aware hybrid reasoning framework for pervasive healthcare. *Personal and Ubiquitous Computing*, 18(4):865–881.

# DETAB: Modelo Colaborativo De Detecção E Alerta De Buracos Em Vias Pavimentadas

Thiago Roberto Lima Lopes<sup>1</sup>, Rodrigo da Rosa Righi<sup>1</sup>, Lucas Pfeiffer Salomão Dias<sup>1</sup>,  
Cristiano André da Costa<sup>1</sup>

<sup>1</sup>Programa de Pos-Graduação em Computação Aplicada (PPGCA)  
da Universidade do Vale do Rio dos Sinos (UNISINOS)  
Av. Unisinos, 950 – Bairro Cristo Rei – São Leopoldo, RS, Brazil

{lolthiago, lucaspfsd}@gmail.com, {cac, rrrighi}@unisinos.br

**Abstract.** *This article presents the Modelo Colaborativo de Detecção e Alerta de Buracos em Vias Pavimentadas (DETAB), which provides a distributed framework that automatically collects through ultrasonic sensors installed in the vehicle, analyzes and shares pothole data among drivers without any kind of human interaction. The applied technologies: HC-SR04 sensors, NodeMCU microcontroller, and an Android App target to be a lower cost solution with high accuracy. The results of both simulated (a toy and in-home lanes) and real (a car in a particular road) scenarios shows that the DETAB was able to map potholes and share the information with other drivers in a satisfactory fashion.*

**Resumo.** *Este artigo apresenta um modelo denominado Modelo Colaborativo de Detecção e Alerta de Buracos em Vias Pavimentadas (DETAB) o qual automaticamente detecta a existência de buracos, através de sensores ultrassônicos instalados no veículo, analisa e compartilha as informações relativas ao buraco com outros motoristas sem a necessidade de nenhum tipo de intervenção humana. As tecnologias envolvidas: sensor HC-SR04, micro controladora NodeMCU e um aplicativo para Android tem por objetivo ser uma solução de baixo custo, mas com alta acurácia. Os resultados obtidos através da avaliação do modelo em ambiente simulado, bem como em ambiente real, mostram que o DETAB foi capaz de identificar buracos na pista e também realizar o compartilhamento dessa informação com outros motoristas de forma satisfatória.*

## 1. Introdução

Historicamente, o poder público não consegue fornecer uma ampliação da malha rodoviária asfaltada e uma manutenção efetiva que garanta qualidade duradoura, portanto, não acompanhando o aumento significativo no número de veículos ocorrido nas últimas décadas. Com o incremento de veículos sobre a mesma estrutura rodoviária, há o surgimento de problemas, sendo as péssimas condições das estradas um potencializador, em especial nos congestionamentos de tráfego e acidentes que findam em perda de vidas humanas. Estes fatos, além de notórios, também se encontram relatados pela Confederação Nacional do Transporte [CNT 2018].

A CNT no seu anuário do Transporte de 2016, mostra uma avaliação de 100.763km de vias brasileiras asfaltadas e conclui que 57,2% estão em condições gerais de regular a péssimo. Este dado torna-se mais impactante quando olha-se para alguns

itens específicos como geometria, onde 77,2% estão de regular a péssimo [CNT 2016]. Novamente a CNT, em 2018 [CNT 2018] mostra dados mais pungentes, onde 61,8% das rodovias pesquisadas estão em condições gerais de regular a péssimo.

Buracos danificam rodas, suspensão, desalinham a direção, colaboram para o aparecimento de ruídos e prejudicam motoristas e passageiros. Eles são responsáveis por milhões de reais em manutenção veicular e execução de apólices de seguro. Na questão de prejuízos, conforme for o órgão responsável pela manutenção da rodovia o estado, o município ou a união podem ser responsabilizados, gerando mais despesas para o setor público. Desta forma, é de muita relevância identificar a localização dos buracos para tomada prévia de ações a fim de evitar acidentes e planejar melhor a rota de tráfego.

Um dos grandes problemas no planejamento da manutenção da rodovia é a falta de confiabilidade das medições manuais, o que pode levar a uma má priorização do trabalho de reparação. Alguns dos benefícios de uma coleta automatizada são a precisão e padronização dos levantamentos das condições e a capacidade de ser ampliado para atuar em qualquer âmbito, ajudando na redução de custos de manutenção em vários níveis.

Visando colaborar com a identificação de buracos, minimizando o risco à vida, este trabalho propõe o Modelo Colaborativo de Detecção e Alerta de Buracos em Vias Pavimentadas (DETAB). Diferentemente dos trabalhos relacionados, o DETAB combina módulos de *hardware*, computação móvel e integra *frameworks* para detectar e compartilhar informações sobre buracos com outros motoristas que trafegam próximos aos buracos existentes sem a necessidade de nenhuma intervenção humana.

O trabalho está organizado da seguinte maneira: na Seção 2 são apresentados os trabalhos relacionados; o modelo proposto e sua arquitetura são descritos em detalhes na Seção 3; as decisões tomadas e a tecnologia utilizada nesta implementação do modelo, são apresentados na Seção 4; a Seções 5 abrange a metodologia de avaliação, descrição dos cenários de testes realizados e os resultados obtidos, e; por fim, na Seção 6 são apresentadas as considerações finais e as expectativas de trabalhos futuros.

## 2. Trabalhos Relacionados

Foram analisados trabalhos relacionados que são o estado da arte e que se propõem a resolver problemas similares. Também fora elaborado uma tabela comparativa (Tabela 1), organizada de forma sequencial, mostrando as principais características.

**1** - [Nienaber et al. 2015] compararam em seu trabalho três abordagens para o cálculo de problema de percepção de profundidade, que é encontrado durante a análise de buracos através de imagem. As abordagens foram: semelhança geométrica de triângulos, razão anarmônica e calibração da matriz de projeção da câmera.

**2** - [Moazzam et al. 2013] apresenta uma alternativa para a análise tridimensional de imagens de problemas em pavimentação usando Microsoft Kinect Sensor. Uma limitação apontada é justamente vinculada ao hardware do Kinect que possui alcance de 4 metros, de acordo com dados oficiais do fabricante, e 3.5 metros de acordo com este trabalho.

**3** - [Rishiwal and Khan 2016] apresentam um aplicativo para Android que realiza a detecção de buracos no pavimento através da vibração e aceleração do dispositivo móvel. Este trabalho apresenta uma taxa de 93.75% de exatidão na detecção de buracos.

4 - [Vupparaboina et al. 2015] apresentam uma abordagem usando uma câmera e um laser fazendo com que seja possível de detectar buracos, secos ou com água, em vias secundárias. A prova de conceito apresentada demonstra uma taxa de assertividade de 98% para buracos secos e 94.7% para buracos com água.

5 - [Madli et al. 2015] propõem um mecanismo de detecção de buracos através de sensor ultrassônico, que efetua alerta ao usuário através de um aplicativo para *smartphone*.

6 - [Murthy and Varaprasad 2014] apresentam uma abordagem baseada em visão 2D com tratamento de imagem feitos em *frames* capturadas de uma câmera do tipo *pin-hole*. Este trabalho possui limitações relacionados as condições de luminosidade.

7 - [Klein and Borges 2016] criaram um aplicativo colaborativo para *smartphone* chamado Buracos Monitor, onde os usuários podem compartilhar informações sobre buracos. Este aplicativo possui um repositório de informações global externo, em um website, que pode ser compartilhado com o poder público.

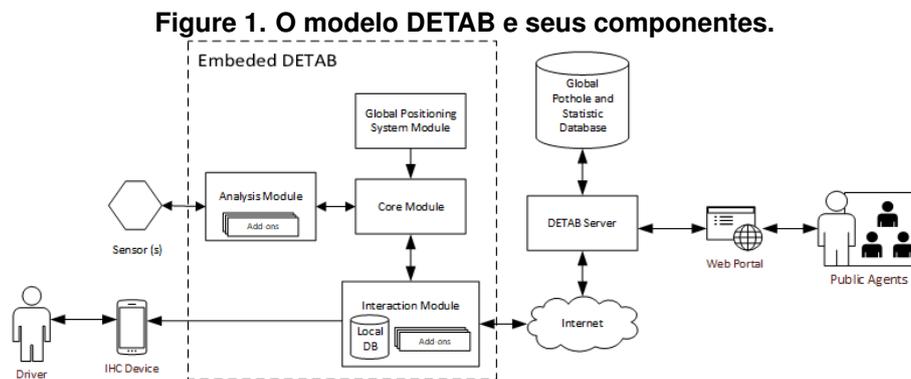
**Table 1. Comparação de trabalhos relacionados**

| Características                  | Trabalhos |          |                      |            |            |          |            | DETAB              |
|----------------------------------|-----------|----------|----------------------|------------|------------|----------|------------|--------------------|
|                                  | 1         | 2        | 3                    | 4          | 5          | 6        | 7          |                    |
| Tipo de detecção                 | Vídeo     | Visão 3D | Vibração/ aceleração | Visão 3D   | Sonar      | Vídeo    | N/A        | Flexível           |
| Dispositivo                      | GoPro     | Kinect   | Android              | Laser/Câm. | Ultrassom  | Câmera   | N/A        | Extensível         |
| Tem notificação ao motorista?    | N/A       | N/A      | N/A                  | N/A        | SMS        | N/A      | Aplicativo | Aplicativo         |
| Distância de alcance             | 30m       | 3.5m     | N/A                  | 20cm       | 4m         | 3.5m     | N/A        | Relativo ao add-on |
| Velocidade                       | 40km/h    | N/A      | N/A                  | N/A        | N/A        | N/A      | N/A        | Relativo ao add-on |
| % precisão - buraco seco         | 98%       | N/A      | 93.75%               | 98%        | N/A        | N/A      | N/A        | Relativo ao add-on |
| % precisão - buraco com água     | N/A       | N/A      | N/A                  | 94.7%      | N/A        | N/A      | N/A        | Relativo ao add-on |
| Detecta profundida / severidade? | Não       | Profund. | Severid.             | Profundid. | Profundid. | Severid. | N/A        | Relativo ao add-on |
| É colaborativo?                  | Não       | Não      | M2M                  | Não        | M2M        | Não      | H2M        | M2M                |
| Notifica motorista?              | N/A       | N/A      | N/A                  | N/A        | SMS        | N/A      | Som/Visual | Som/Visual         |
| Acesso web para autoridades?     | N/A       | Não      | Sim                  | Não        | Não        | Não      | Sim        | Sim                |
| Precisa de internet?             | N/A       | Não      | Não                  | N/A        | Não        | Não      | Sim        | Sim                |
| Salva dados localmente?          | N/A       | N/A      | Sim                  | Não        | Não        | Não      | Sim        | Sim                |
| Multi-sensor?                    | Não       | Não      | Não                  | Não        | Não        | Não      | Não        | Sim                |

É fundamental ressaltar que três pontos importantes diferenciam o modelo proposto dos demais trabalhos relacionados são: a) a capacidade de emitir alertas aos usuários motoristas, de modo a dar-lhes ciência do que está por vir e viabilizar a tomada de medidas cabíveis com antecedência, eliminando ou ao menos reduzindo os riscos de acidentes e danos materiais possíveis de serem causados; b) o modelo tem por objetivo ser uma solução de comunicação M2M, ou seja, não necessitando de nenhuma intervenção humana para que todo o processo ocorra; c) a oferta de um portal Web para autoridades encarregadas da manutenção das vias públicas, de forma a auxiliá-los na organização dos pontos prioritários e que necessitam maior atenção.

### 3. Modelo Colaborativo De Detecção E Alerta De Buracos Em Vias Pavimentadas - DETAB

O modelo DETAB é independente da tecnologia de implementação, sendo possível implementá-lo das mais diversas formas e tecnologias, a Figura 1 ilustra o modelo.



Conforme ilustrado na Figura 1 é possível visualizar a forma de interação no modelo, onde o DETAB trabalha junto aos sensores que são capazes de detectar irregularidades no asfalto. O DETAB ao detectar um buraco, calcula seu posicionamento geográfico (latitude e longitude) usando o sistema GPS e notifica o motorista sobre a(s) incidência(s) de buraco(s) à frente. O modelo proposto possui algumas funcionalidades básicas que garantem sua melhor eficiência e são detalhadas abaixo:

**1- Fornecer uma interface padrão para recepção de informações do módulo de detecção:** esta funcionalidade garante com que o sistema possa trabalhar com diversos tipos de dispositivos e sensores como fonte de informação. Para tanto, são definidos quais dados são esperados e seus formatos, de modo que possam ser desenvolvidos *add-ons* normalizadores específicos para cada tecnologia que se deseja acoplar.

**2- Notificar o usuário durante o trajeto:** esta funcionalidade trata da apresentação para o usuário da informação sobre a existência de buraco(s) à frente. O módulo que implementará esta funcionalidade terá por objetivo ser extensível à diversos dispositivos que tenham interfaceamento com o usuário, pois o foco principal é garantir a ciência do motorista sobre o(s) buraco(s) em seu percurso, independente da tecnologia adotada.

**3- Enviar as informações coletadas para a base de dados global:** esta funcionalidade garante que o modelo seja colaborativo, uma vez que, o modelo executando em um determinado veículo envie e receba informações da base de dados global, que por sua vez são oriundas do modelo executando em outros veículos.

#### 3.1. Arquitetura

A arquitetura do DETAB é dividida em quatro módulos principais. Para que o modelo seja independente de tecnologia, emprega-se o conceito de *add-on*, tendo estes a responsabilidade de comunicação com uma determinada tecnologia, evitando que o modelo demande conhecimento qualificado de um nicho tecnológico. Este trabalho apresenta margem para três grandes grupos de *add-ons*, que também são apresentados a seguir.

**Add-ons de análise específica por tipo de entrada** - Devem ser interpretar sinais produzidos por sensores, e notificar o módulo de análise, respeitando a interface de comunicação que determina um padrão de formato dos dados, quando um buraco for identificado.

**Add-ons de interação com o usuário motorista** - Devem interagir com outros sistemas que fazem comunicação com o usuário motorista, convertendo a informação recebida em um determinado formato para o formato requerido pela aplicação.

**Add-ons de persistência e colaboração entre partes do modelo** - Devem salvar e compartilhar a informação sobre a existência de buraco em determinada posição geográfica. Este satisfaz a parte colaborativa uma vez que os dados compartilhados podem ser consumidos por veículos que também estejam executando o sistema.

**Analysis Module** - Este módulo tem por responsabilidade encapsular múltiplas implementações de algoritmos capazes de lidarem com diversos tipos de entrada de dados, que dependem basicamente do tipo de sensor utilizado. Para este módulo não encontrar-se atrelado a um determinado tipo de sensor ou tecnologia, ele atua como um *dispatcher*, executando *add-ons* que realizam a análise especializada para cada tipo de sensor. Este módulo, após a detecção realizada por um ou mais *add-ons*, coleta as informações fornecidas e invoca o Core Module, para dar seguimento ao processo.

**Core Module** - Este modulo atua como orquestrador que interliga todos os demais módulos do sistema. Sua atuação inicia em forma de reação a uma notificação do Analysis Module. Ao ser notificado, obtendo as informações provenientes do(s) sensor(s) este módulo usa o módulo de posicionamento para obter suas coordenadas geográficas (latitude e longitude) de forma a compor uma informação completa. Estas informações serão redirecionadas para o módulo de interação para dar seguimento ao processo.

**Interaction Module** - Este módulo é responsável por fazer a interação com o meio externo através de lançamento de eventos, onde um evento trata-se da identificação de um buraco. Um dos objetivos que este módulo visa atender é a funcionalidade acima descrita de notificar o usuário durante o trajeto, bem como enviar as informações coletadas para a base de dados global. Para este módulo não atrelar-se a tecnologia ele também atua com um *dispatcher*, que faz a propagação de eventos para *add-ons* registrados. Neste contexto, a ação a ser tomada e os detalhes de como fazê-la, ao lançar-se um evento são definidos pelo *add-ons* que está registrado, que podem ser nenhum ou muitos.

**Global Positioning System Module** - Este módulo é responsável por fornecer ao módulo requisitante, neste caso o módulo principal, as informações de posicionamento geográfico (latitude e longitude). Estas informações podem ser obtidas através do dispositivo GPS (Sistema de posicionamento global) presente nos *smartphones* ou através da central multimídia em veículos que possuem um sistema GPS de fábrica.

**Web Portal** - O portal *web* visa ser uma forma fácil e eficaz de colaborar com o poder público, oferecendo uma ferramenta onde seja possível verificar-se a ocorrência de buracos e sua severidade de forma a guiar a priorização das ações realizadas para manutenção das vias públicas. Este portal *web* oferece mecanismos para os agentes do poder público assinalarem que determinado(s) buraco(s) foi(ram) concertado(s).

**DETAB Server** - O Detab Server visa prover uma interface de consulta e

alimentação de dados online de forma a ser um *hub* dos sistema rodando nos veículos, bem como para o portal *web*. Este servidor também tem por responsabilidade filtrar e eliminar mapeamentos duplicados, ou seja, o mesmo buraco mapeado por outros motoristas, bem como encapsular o acesso aos bancos de dados.

#### 4. Decisões de Tecnologia e Implementação

Conforme o modelo proposto, foi desenvolvido um protótipo que implementa: O sistema embarcado, o aplicativo de interação com o motorista, além do lado *server-side*. Este trabalho parte da premissa de que os sensores e os algoritmos empregados para análise e detecção de buracos já existam, então dentro das possibilidades, optou-se pelo sensor ultrassônico HC-SR04 utilizado em [Madli et al. 2015], conectado a uma placa controladora ESP8266-WiFi-SOC da Espressif com o *firmware* NodeMCU 1.0. Existem diferentes tipos de sensores ultrassônicos com diferentes faixas de transmissão e ângulos de detecção, todavia, o HC-SR04 se demonstrou em particular atrativo devido sua popularidade, baixo custo e estar presente em outros trabalhos relacionados.

O *add-on* de interação com o motorista mostra os alertas de buracos através de um aplicativo Android. Neste aplicativo também foram imputadas as responsabilidades do Core Module, Global Positioning System Module e Interaction Module. O dispositivo utilizado nos testes de avaliação foi o Motorola Moto X Style, com Android versão 6.0, 3GB de memória RAM e processadores: Dual-core 1.8GHz + Quad-core 1.44GHz. Este aparelho tem uma precisão de GPS que varia de 3 à 5 metros e utiliza os sistemas A-GPS e GLONASS combinados para aumentar a precisão. Para a implementação do Detab Server foi escolhido a tecnologia de ASPNET Web API, de modo a fornecer dados ao aplicativo, questões de autenticação e inserção de novos buracos mapeados.

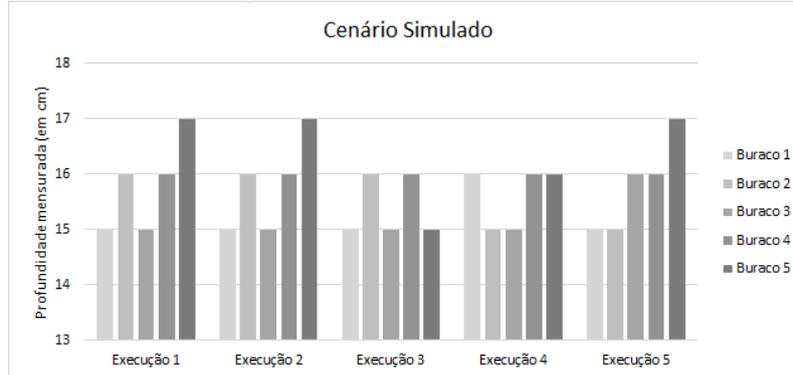
#### 5. Metodologia de Avaliação: Experimentos e Resultados Obtidos

A elaboração de uma forma de análise do sistema com viés quantitativo resultou na criação de uma linha de avaliação que contemple testes de funcionalidade e de integração do sistema, mais especificamente nas questões de: A) detecção de buracos por parte do sensor; B) envio dessas informações para o aplicativo Android; C) pela emissão de alerta pelo aplicativo para o motorista, e por fim, a submissão dos buracos ao DETAB Server.

##### 5.1. Cenário: Teste com ambiente e buracos simulados

A fim de formar um método de teste exequível repetitivamente, criou-se um cenário de pequena escala onde foi fixado na parte frontal de um carro de controle remoto (que tem 15 cm de altura), o sensor HC-SR04, apontando para baixo, portanto, formando ângulo de 90 graus em relação ao solo. Este sensor ficou disposto a uma altura de 12 cm em relação ao solo, sendo o sistema pré-configurado para uma tolerância de 2 cm, ficando assim, definido como buraco valores lidos que excedam o limite de 15 cm de altura.

Para simular o ambiente foi desenvolvido um percurso usando 4 folhas de isopor de 1 m x 33 cm x 20 mm, totalizando 4 metros de pista, onde foram dispostos 5 buracos de aproximadamente 13 cm x 8 cm em espaçamento aleatório. Nestas folhas de isopor também foram inseridas cavidades que não são considerados buracos devido sua profundidade ser inferior a tolerância, de modo a testarmos se este o sistema não iriam emitir falso positivo. A Figura 2 é uma montagem de fotos mostrando a criação da pista de teste usando isopor, bem como a instalação e fixação do sensor no carro de controle remoto.

**Figure 2. Fotos tiradas durante os testes do cenário simulado.****Figure 3. Buracos mapeados durante o teste em ambiente simulado.**

Assim, o experimento proposto se baseou em realizar a simulação em uma bateria de 5 execuções, e com os resultados coletados em cada uma delas efetuar um comparativo com a quantidade efetiva de buracos, além de verificar se houve a notificação de mapeamento pelo aplicativo.

O gráfico apresentado na Figura 3 apresenta as profundidades mensuradas para cada um dos cinco buracos em cada uma das cinco execuções. Evidencia-se que todos os buracos foram mapeados corretamente e notificados ao aplicativo Android, apenas considerando-se uma variação entre as medidas atribuída à precisão do sensor utilizado.

## 5.2. Cenário: teste em ambiente real

De modo a levar o experimento a um nível mais concreto, instalou-se o sensor HC-SR04 em um veículo de passeio, VW Voyage, cujo qual tem um altura padrão em relação ao solo de 16 cm. O sensor foi disposto da mesma forma dos testes realizados no ambiente simulado, ou seja, instalado na parte frontal, apontando para baixo. Todavia, neste cenário o sensor ficou disposto a uma altura de 30 cm do solo. A placa controladora foi instalada no interior do veículo e alimentada através de uma tomada de 12 volts do próprio carro.

Para execução dos testes foi escolhido a rua Manoel Osório Ourique, Girassol, Gravataí/RS, dadas suas características: ser asfaltada, reta, plana e possuir buracos de variados formatos, tamanhos e profundidades. O cenário busca aferir se os resultados

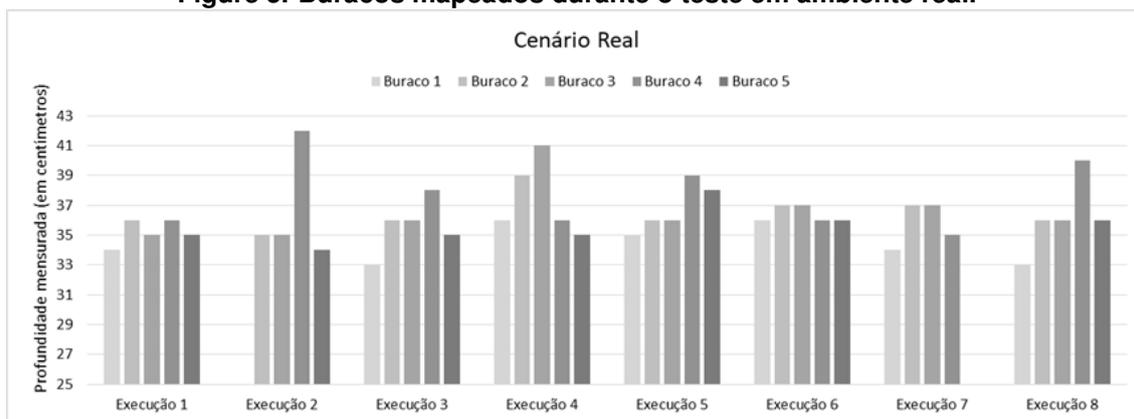
obtidos pelo sistema condizem com a realidade. Para tal, manualmente contou-se os buracos em uma determinada trajetória dentro da via, e após percorreu-se por 8 vezes o mesmo percurso mapeado.

**Figure 4. Fotos tiradas durante os testes do cenário real.**



A Figura 4 mostra 3 fotos que exemplificam a instalação do sistema no veículo. A primeira mostrando a posição da instalação do sensor na parte frontal, a segunda mostrando o *case* criado e montado manualmente que envolve o sensor para apoiá-lo e protegê-lo, e a terceira a instalação da placa controladora no interior do veículo.

**Figure 5. Buracos mapeados durante o teste em ambiente real.**



Durante a execução dos testes no ambiente real, o sistema apresentou um comportamento anormal, detectando buracos onde não havia. De forma a tentar calibrar o sistema, manipulou-se algumas variáveis de ajuste do algoritmo de análise. Dentre essas variáveis, o tempo de aguardo entre os ciclos de leitura do sensor e a tolerância dos valores lidos. Os testes demonstraram que não era uma questão de trabalhar com a tolerância, mas sim verificar a imprecisão das leituras fornecidas pelo sensor HC-SR04 que não se apresentava nos testes no cenário simulado. Neste quesito notou-se a deficiência deste sensor em efetuar leituras de determinadas superfícies não completamente sólidas, como é o caso da manta asfáltica usada na pavimentação das vias brasileiras.

O gráfico apresentado na Figura 5 apresenta as profundidades mensuradas (em centímetros) para cada um dos cinco buracos em cada uma das oito execuções dos testes. Evidencia-se que neste cenário nem todos os buracos foram mapeados (buraco 1 x execução 2 e buraco 5 x execução 7), todavia, todos aqueles que foram detectados foram corretamente enviados ao aplicativo Android. Também é possível observar as oscilações que ocorrem entre as execuções para o mesmo buraco, corroborando com a indicação de que o sensor HC-SR04 não opera adequadamente em determinadas superfícies.

Entretanto, a leitura por parte do sensor é apenas um ponto técnico e específico relacionado a uma instancia dos *add-ons*, e seu mal funcionamento não implica na invalidação do modelo. O modelo mostra-se válido visto que a imperfeição na detecção de buracos deste *add-on* pode ser corrigida substituindo por outra tecnologia.

### 5.3. Cenário: teste de notificação ao motorista pelo aplicativo

Este cenário prático de teste visa demonstrar e aferir o funcionamento da funcionalidade de notificação na forma visual e sonora ao motorista durante o trajeto, dado a existência de um(s) buraco(s) à frente. Para tal, foram pré-inseridas no sistema incidências de buracos dispostos aleatoriamente ao longo de duas localidades no município de Gravataí/RS e percorreu-se três vezes cada trecho que continha os buracos. Com o aplicativo DETAB aberto durante a condução de um veículo trafegando a aproximadamente 40km/h, observou-se que o aplicativo notificou corretamente todas as incidências de buracos, durante todas as seis execuções, indicando que o módulo de notificação do usuário está comunicando-se adequadamente com os demais e operando plenamente sua função.



A Figura 6 mostra três *screenshots* da tela do *smartphone* durante a exibição de notificações sobre buracos à frente (sonora e visual), a onde o sistema foi previamente configurado para alertar o motorista quando a distância do veículo em relação ao buraco for igual ou inferior a 15 metros, mas este valor pode ser configurado com outros valores para haver um tempo hábil maior para tomada de decisões. Devido a precisão do sistema GPS não é possível afirmar exatamente em qual faixa da via o buraco está presente, sendo notificado inclusive buracos que não necessariamente estão na rota frontal do veículo.

## 6. Conclusão e Trabalhos Futuros

Este trabalho apresenta o DETAB, um modelo de detecção de buracos e alertas à motoristas sobre a incidência de buracos, prevenindo acidentes e fornecendo dados à órgãos que planejam e executam ações de reparo nem vias asfaltadas. Neste trabalho, priorizou-se o baixo custo, valendo-se de um sensor ultrassônico simples e um aplicativo para Android, presente na maior parte dos dispositivos moveis brasileiros [WorldPanel 2016].

Com base nos testes, constate-se evidências do funcionamento do DETAB e da capacidade de identificar buracos, ser colaborativo ao enviar informações coletadas para a API, e por distribuir essa informação a outros motoristas. Entretanto, considerando-se

que os testes exitosos foram em sua maioria no ambiente simulado, é importante salientar que os resultados obtidos não podem ser generalizados para qualquer tipo de via.

Como trabalhos futuros, espera-se testar outros sensores que proporcionem maior precisão ao sistema. Além disto, a escolha de outros sensores e formas de mapear o solo permitirá antever os buracos até mesmo para o primeiro motorista a cursar um trajeto. Uma integração com o sistema *On Board Diagnostics* (ODB-II) dos veículos pode fornecer parâmetros úteis a serem usados para alertar o motorista viabilizando uma aproximação e manobra de desvio cada vez mais segura.

## References

- CNT, C. N. d. T. (2016). Anuário cnt do transporte - estatísticas consolidadas. Disponível em: <http://anuariodotransporte.cnt.org.br/File/PrincipaisDados.pdf>. Acesso em: 09 Junho 2017.
- CNT, C. N. d. T. (2018). O estudo transporte rodoviário: Rodovias esquecidas do brasil. Disponível em: [http://cms.cnt.org.br/ImagensCNT/PDFsCNT/Estudo – RodoviasEsquecidas/rodovias – esquecidas – brasil.pdf](http://cms.cnt.org.br/ImagensCNT/PDFsCNT/Estudo-RodoviasEsquecidas/rodovias-esquecidas-brasil.pdf). Acesso em: 30 Março 2018.
- Klein, B. and Borges, C. E. L. (2016). Buracos monitor. Disponível em: [http://www.buracosmonitor.com.br/blog/2016/08/quem – somos/](http://www.buracosmonitor.com.br/blog/2016/08/quem-somos/). Acesso em: 09 Junho 2017.
- Madli, R., Hebbar, S., Pattar, P., and Golla, V. (2015). Automatic detection and notification of potholes and humps on roads to aid drivers. *IEEE Sensors Journal*, 15(8):4313–4318.
- Moazzam, I., Kamal, K., Mathavan, S., Usman, S., and Rahman, M. (2013). Metrology and visualization of potholes using the microsoft kinect sensor. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1284–1291. IEEE, IEEE.
- Murthy, S. B. S. and Varaprasad, G. (2014). Detection of potholes in autonomous vehicle. *IET Intelligent Transport Systems*, 8(6):543–549.
- Nienaber, S., Kroon, R. S., and Booyesen, M. J. (2015). A comparison of low-cost monocular vision techniques for pothole distance estimation. In *Computational Intelligence, 2015 IEEE Symposium Series on*, pages 419–426. IEEE.
- Rishiwal, V. and Khan, H. (2016). Automatic pothole and speed breaker detection using android system. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1270–1273. IEEE.
- Vupparaboina, K. K., Tamboli, R. R., Shenu, P. M., and Jana, S. (2015). Laser-based detection and depth estimation of dry and water-filled potholes: A geometric approach. In *Communications (NCC), 2015 Twenty First National Conference on*, pages 1–6. IEEE.
- WorldPanel, K. (2016). Smartphone os market share. Disponível em: [http://www.kantarworldpanel.com/global/smartphone – os – market – share/](http://www.kantarworldpanel.com/global/smartphone-os-market-share/). Acesso em: 09 Junho 2017.

# EXEHDA-RR: Uma proposta para tratar incertezas e otimizar o processo de classificação de recursos na IoT\*

Renato Dilli, Amanda Argou, Ana Pernas, Renata Reiser, Adenauer Yamin

<sup>1</sup> Universidade Federal de Pelotas (UFPel)

Rua Gomes Carneiro, 1 - Centro - CEP 96010-610 – Pelotas – RS – Brasil

{renato.dilli, aacardozo, marilza, reiser, adenauer}@inf.ufpel.edu.br

**Abstract.** *The Internet of Things (IoT) is characterized by a broad range of resources connected to the Internet, requesting and providing services simultaneously. Given this scenario, suitably selecting the resources that best meet users' demands has been a relevant and current research challenge. This paper presents the EXEHDA-RR, a proposal to classify and select the most appropriate resource, applying fuzzy logic to solve uncertainties in the definition of ideal weights for QoS attributes, and adding machine learning to the pre-classification of resources in order to reduce the computational cost generated by the MCDA algorithms. The experimental results of the pre-classification show the efficiency of the proposed model.*

**Resumo.** *A Internet das Coisas (IoT) é caracterizada por uma ampla gama de recursos conectados à Internet, solicitando e fornecendo serviços simultaneamente. Diante desse cenário, selecionar adequadamente os recursos que melhor atendam às demandas dos usuários tem sido um desafio de pesquisa relevante e atual. Este artigo apresenta o EXEHDA-RR, uma proposta para classificar e selecionar o recurso mais apropriado, aplicando lógica fuzzy para resolver incertezas na definição de pesos ideais para os atributos de QoS, e agregando aprendizado de máquina à pré-classificação dos recursos visando reduzir o custo computacional gerado pelos algoritmos MCDA. Os resultados experimentais da pré-classificação mostram a eficiência do modelo proposto.*

## 1. Introdução

O cenário atual contabiliza aproximadamente 27 bilhões de coisas conectadas à Internet disponibilizando serviços aos clientes e previsão de mais de 125 bilhões até 2030 [IHS Markit 2017]. O número de serviços será maior à medida que as coisas físicas poderão fornecer uma ou mais funcionalidades.

Um desafio a ser vencido após a descoberta dos recursos é classificar os serviços para selecionar o que melhor atende a solicitação do usuário [García 2012], sendo esta uma tarefa difícil e demorada. Os processos de classificação vem se concentrando nas preferências do usuário, que frequentemente estabelecem uma ordem baseada na Qualidade de Serviço (QoS) ou propriedades não-funcionais (NFPs - *Non-Functional Properties*) [Schröpfer et al. 2006].

---

\*O presente trabalho foi realizado com apoio da CAPES (Programa Nacional de Cooperação Acadêmica - Procad) e da FAPERGS (Programa Pesquisador Gaúcho - PqG).

Os recursos, geralmente disponibilizados por meio de serviços, possuem atributos de QoS com valores apresentando unidades de medidas heterogêneas e faixas amplas nos valores mínimos e máximos. Para selecionar o recurso mais oportuno para a requisição do cliente, é essencial a correta especificação destes valores.

Os responsáveis por definir estes valores devem ter conhecimento de todas as características dos atributos de qualidade, e desta forma avaliar os valores para qualificar os recursos. Considerando que em cada domínio a definição destes valores pode ser feita por especialistas com percepções diferentes, surge a incerteza quando da definição dos valores a serem adotados como atributos de QoS.

Com a premissa de atender estes desafios, o objetivo geral deste trabalho é a concepção de um modelo para a classificação de recursos, denominado EXEHDA-RR (*Resource Ranking*). O trabalho tem como premissa contribuir com a arquitetura de software existente no serviço de descoberta do *middleware* EXEHDA.

O diferencial dessa proposta está no uso combinado de lógica fuzzy e aprendizado de máquina no reconhecimento de padrões para classificação de recursos. Os recursos são inicialmente classificados por algoritmo de Análise de Decisão Multicritério (MCDA). Em cada classificação, um novo treinamento do algoritmo de aprendizado de máquina é realizado para pré-classificar novos recursos à medida que eles entram na infraestrutura computacional. Esse processo reduz a necessidade de processar todos os cálculos do algoritmo MCDA em cada solicitação do cliente. O cenário de avaliação do EXEHDA-RR mostrou resultados satisfatórios pela combinação da lógica fuzzy com o aprendizado de máquina. Considerando a revisão de literatura e as demandas do grupo de pesquisa, os resultados alcançados são oportunos para uso no *middleware* EXEHDA.

Este artigo está organizado da seguinte forma: na Seção 2 é apresentada a arquitetura e as funcionalidades do serviço de descoberta e ranqueamento de recursos EXEHDA-RR, na Seção 3 o tratamento da incerteza na definição dos pesos de QoS é apresentado, na Seção 4 é apresentada a avaliação da acurácia da pré-classificação de recursos do EXEHDA-RR, na Seção 5 são discutidos os Trabalhos Relacionados e na Seção 6 estão as Considerações Finais.

## 2. EXEHDA-RR: Arquitetura e Funcionalidades

O EXEHDA [Davet 2015] é o *middleware*, foco deste trabalho, responsável por prover a infraestrutura computacional básica para a IoT. Este ambiente computacional é constituído por células de execução, nas quais os dispositivos computacionais são distribuídos. Cada célula é constituída dos seguintes componentes: (i) EXEHDABase, o elemento central da célula, sendo responsável por todos serviços básicos e constituindo referência para os demais elementos; (ii) o EXEHDANodo que corresponde aos dispositivos computacionais responsáveis pela execução das aplicações; (iii) o EXEHDANodo móvel, um subcaso do anterior, que corresponde aos dispositivos tipicamente móveis que podem se deslocar entre as células do ambiente ubíquo; (iv) o EXEHDABorda, responsável por fazer a interoperação entre os serviços do *middleware* e os diversos tipos de *gateways*; e (v) o EXEHDAGateway, que 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 EXEHDA-RR (*Resource Ranking*) é um serviço para classificação de recursos

proposto para o *middleware* EXEHDA. Sua arquitetura de software foi modelada considerando a dinamicidade em que os recursos entram e saem do ambiente, portanto, controla a presença de dispositivos, inclusive com baixo poder computacional, conectados à gateways. O aumento da expressividade na descrição e consulta por recursos foi provida através de tecnologias de Web Semântica com intuito de potencializar a descoberta de recursos no ambiente. A arquitetura de software do EXEHDA-RR (Figura 1) possui três componentes distintos: (CD) Componente Diretório; (CR) Componente Recurso; e (CC) Componente Cliente. O CD, localizado no EXHDABase, contém os módulos responsáveis pela descoberta e ranqueamento de recursos.

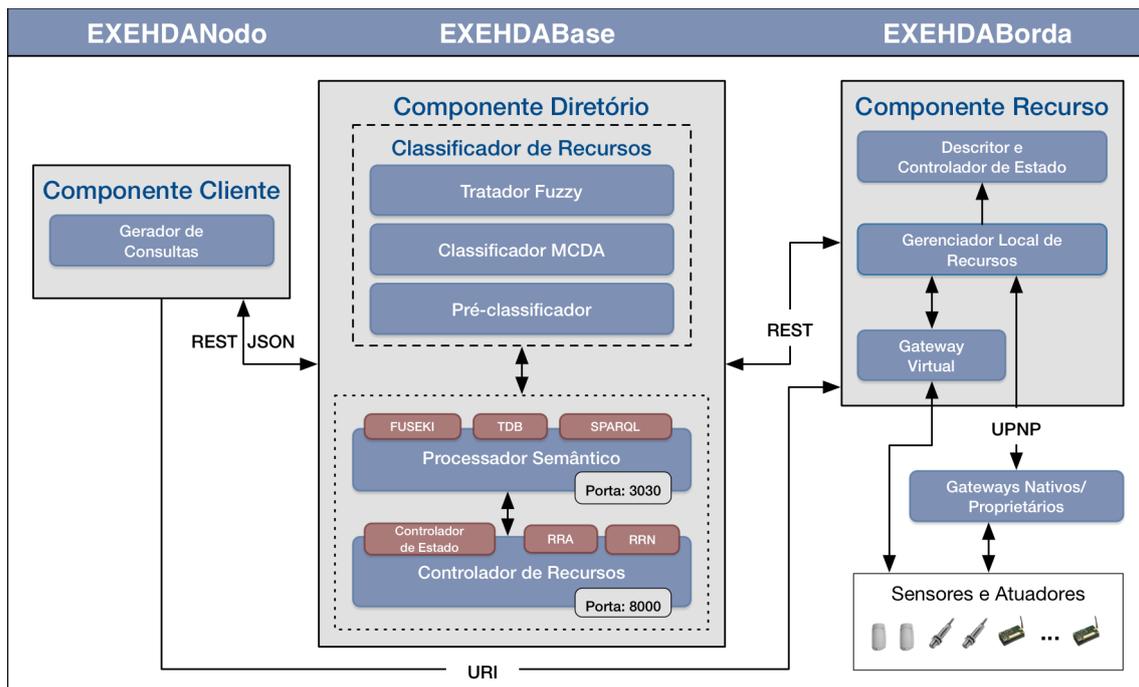


Figura 1. Sub-módulos do Classificador de Recursos

O processo de classificação dos recursos já descobertos é promovido pelo módulo “Classificador de Recursos”. Este módulo contém os seguintes sub-módulos:

- **Tratador Fuzzy.** Tem por objetivo resolver as divergências entre vários especialistas na atribuição dos pesos de importância de cada atributo de QoS. As funções de pertinência empregadas e o método de defuzzificação estão detalhados em [Argou et al. 2017].
- **Classificador MCDA.** Responsável por classificar todos os recursos descobertos através do algoritmo MCDA proposto, tendo como entrada os pesos definidos pelo Tratador Fuzzy. O algoritmo MCDA proposto para classificação dos recursos é detalhado em [Dilli et al. 2017].
- **Pré-classificador.** Os novos recursos recém descobertos são pré-classificados através de algoritmo de aprendizagem de máquina, visando reduzir o custo computacional gerado pelo algoritmo MCDA.

### 3. Tratamento da incerteza na definição dos pesos de QoS

O EXEHDA-RR utiliza ontologias para descrição dos recursos e seus atributos de qualidade (QoS), avalia os requisitos não-funcionais dos recursos por meio de algoritmo

MCDA, considera as preferências do cliente, trata a incerteza na definição dos pesos dos atributos através de lógica fuzzy e realiza a pré-classificação de recursos através do algoritmo de aprendizagem de máquina LMT.

No processo de classificação de recursos, que pode 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 usuário e administradores da infraestrutura computacional é uma atividade que depende da experiência e do conhecimento de cada um. O tratamento da incerteza introduzida por essas divergências é uma das contribuições deste trabalho.

Para avaliar o emprego da lógica fuzzy na especificação dos atributos de QoS ideais, definidos pelos especialistas do ambiente computacional, foi utilizado o *dataset* QWS versão 2.0 disponibilizado por [Al-Masri and Mahmoud 2007], com 2.505 recursos e 9 atributos de qualidade. A Tabela 1 descreve os 5 atributos utilizados.

**Tabela 1. Atributos do dataset QWS**

| Atributo      | Descrição   | Unid |
|---------------|---|------|
| Response Time | Tempo do envio de uma requisição e receber sua resposta | ms   |
| Availability  | Número de invocações corretas/total invocações          | %    |
| Throughput    | Total de invocações por um determinada período          | %    |
| Reliability   | Taxa de erro no total de mensagens                      | %    |
| Latency       | Tempo que o servidor leva para processar uma requisição | ms   |

A Figura 2 e a Tabela 2 exibem os Termos de Linguísticos e os pesos atribuídos.

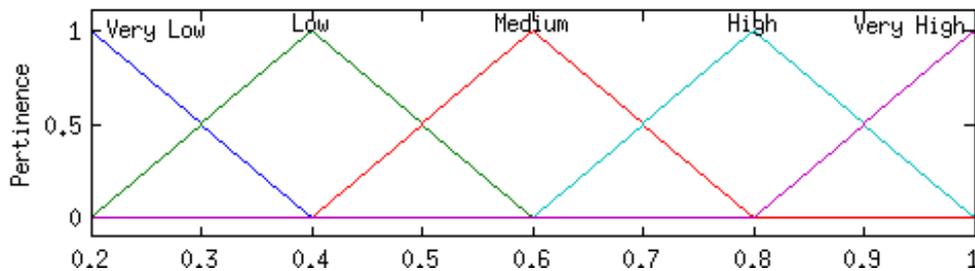


Figura 2. Representação dos conjuntos Fuzzy

**Tabela 2. Grau de Importância**

| Termo Linguístico | Valor         |
|-------------------|---------------|
| VH - Very High    | (0.8,1.0,1.0) |
| H - High          | (0.6,0.8,1.0) |
| M - Medium        | (0.4,0.6,0.8) |
| L - Low           | (0.2,0.4,0.6) |
| VL - Very Low     | (0.0,0.2,0.4) |

**Tabela 3. Avaliação dos especialistas**

| Atrib | Esp1 | Esp2 | Esp3 | Esp4 |
|-------|------|------|------|------|
| RT    | M    | H    | L    | H    |
| AV    | VH   | VH   | H    | VH   |
| TH    | M    | M    | H    | H    |
| RE    | H    | H    | VH   | H    |
| LA    | VH   | H    | VL   | H    |

A Tabela 3 simula a definição do grau de importância de cada atributo de qualidade usando os Termos Linguísticos definidos por cinco especialistas. Esta avaliação visa

definir os pesos a serem atribuídos a cada atributo, fornecendo os recursos mais adequados à solicitação do usuário.

Na Tabela 4 é apresentada a conversão dos Termos Linguísticos, atribuídos pelos especialistas, em números fuzzy triangulares.

A média difusa entre os especialistas e a defuzzificação usando o método da média ponderada também é apresentada. No final, os valores dos atributos são normalizados.

**Tabela 4. Cálculo dos pesos de QoS**

| Atrib | Esp1          | Esp2          | Esp3          | Esp4          | Méd.Fuzzy        | Def  | Norm |
|-------|---------------|---------------|---------------|---------------|------------------|------|------|
| RT    | (0.4,0.6,0.8) | (0.6,0.8,1.0) | (0.2,0.4,0.6) | (0.6,0.8,1.0) | (0.45,0.65,0.85) | 0.65 | 0.70 |
| AV    | (0.9,1.0,1.0) | (0.9,1.0,1.0) | (0.6,0.8,1.0) | (0.9,1.0,1.0) | (0.83,0.95,1.00) | 0.93 | 1.00 |
| TH    | (0.4,0.6,0.8) | (0.4,0.6,0.8) | (0.6,0.8,1.0) | (0.6,0.8,1.0) | (0.50,0.70,0.90) | 0.70 | 0.75 |
| RE    | (0.6,0.8,1.0) | (0.6,0.8,1.0) | (0.9,1.0,1.0) | (0.6,0.8,1.0) | (0.68,0.85,1.00) | 0.87 | 0.94 |
| LA    | (0.9,1.0,1.0) | (0.6,0.8,1.0) | (0.0,0.2,0.4) | (0.6,0.8,1.0) | (0.53,0.70,0.85) | 0.70 | 0.75 |

Os valores normalizados resultantes serão aplicados no processo de classificação e seleção de recursos do algoritmo MCDA.

#### 4. Avaliação da acurácia na pré-classificação

Nesta seção é avaliada a acurácia na classificação realizada pelo algoritmo de aprendizagem de máquina no processo de pré-classificação de novos recursos.

Para a realização dos testes foi utilizado o *dataset* QWS [Al-Masri and Mahmoud 2007], com os cinco atributos já apresentados na Tabela 1.

Para prototipação dos algoritmos de árvores de decisão, foi utilizado o MOA - *Massive Online Analysis*. O MOA é um framework para mineração de dados. As bibliotecas do MOA e do WEKA foram desenvolvidas em java, portanto foi utilizado o Jpype para que o Python possa manipular as bibliotecas em Java.

##### 4.1. Explorando MCDA na construção do ranking de referência

A aplicação do algoritmo MCDA é demonstrada na classificação de um *dataset* contendo 5 atributos de qualidade para 10 recursos. Cada linha representa um recurso e cada coluna um atributo de qualidade (Tabela 5). Os valores dos atributos Resp.Time e Latency são melhores se forem baixos e os valores dos atributos Availability, Throughput e Reliability são melhores se forem altos.

A primeira etapa da classificação é a normalização dos dados. Para tanto, consideramos os vetores  $N=\{0,1,1,1,0\}$  e  $C=\{6,2,3,2,50\}$  e  $w=\{0.70,1.00,0.75,0.94,0.75\}$ . Os atributos Resp.Time e Latency qualificam o recurso com valores baixos e Availability, Throughput e Reability qualificam com valores altos. Todos os atributos são normalizados com valor máximo definido em “C”, o resultado é apresentado na Tabela 6.

A Tabela 7 apresenta os valores dos atributos após dividir o valor normalizado da Tabela 6 pelo maior valor normalizado de cada coluna. Após é somado todos os valores de atributos em cada linha. A seguir o recurso recebe um valor que vai de 0 a 100. Será dado o valor 100 para o melhor recurso do *dataset*. A classificação de 1 a 4 é atribuída através da regra ilustrada na Figura 3.

**Tabela 5. Exemplo do dataset QWS**

| Resp.Time | Availability | Throughput | Reliability | Latency |
|-----------|--------------|------------|-------------|---------|
| 302.75    | 89           | 7.1        | 73          | 187.75  |
| 482       | 85           | 16         | 73          | 1       |
| 3321.4    | 89           | 1.4        | 73          | 2.6     |
| 126.17    | 98           | 12         | 67          | 22.77   |
| 107       | 87           | 1.9        | 73          | 58.33   |
| 107.57    | 80           | 1.7        | 67          | 18.21   |
| 255       | 98           | 1.3        | 67          | 40.8    |
| 136.71    | 76           | 2.8        | 60          | 11.57   |
| 102.62    | 91           | 15.3       | 67          | 0.93    |
| 200       | 40           | 13.5       | 67          | 41.66   |

**Tabela 6. Atrib. Normalizados**

| RT   | AV   | TH   | RE   | LA    |
|------|------|------|------|-------|
| 1.70 | 1.07 | 0.97 | 1.06 | 0.21  |
| 1.07 | 1.02 | 2.19 | 1.06 | 38.56 |
| 0.15 | 1.07 | 0.19 | 1.06 | 14.83 |
| 4.07 | 1.18 | 1.64 | 0.98 | 1.69  |
| 4.80 | 1.04 | 0.26 | 1.06 | 0.66  |
| 4.78 | 0.96 | 0.23 | 0.98 | 2.12  |
| 2.02 | 1.18 | 0.18 | 0.98 | 0.95  |
| 3.76 | 0.91 | 0.38 | 0.87 | 3.33  |
| 5.01 | 1.09 | 2.10 | 0.98 | 41.46 |
| 2.57 | 0.48 | 1.85 | 0.98 | 0.93  |

**Tabela 7. Resultado da Classificação**

| RT   | AV   | TH   | RE   | LA   | MCDA  | Classif |
|------|------|------|------|------|-------|---------|
| 0.24 | 0.91 | 0.33 | 0.94 | 0.00 | 61.18 | 2       |
| 0.15 | 0.87 | 0.75 | 0.94 | 0.70 | 85.99 | 1       |
| 0.02 | 0.91 | 0.07 | 0.94 | 0.27 | 55.67 | 3       |
| 0.57 | 1.00 | 0.56 | 0.86 | 0.03 | 76.42 | 1       |
| 0.67 | 0.89 | 0.09 | 0.94 | 0.01 | 65.68 | 2       |
| 0.67 | 0.82 | 0.08 | 0.86 | 0.04 | 62.27 | 2       |
| 0.28 | 1.00 | 0.06 | 0.86 | 0.02 | 56.14 | 3       |
| 0.53 | 0.78 | 0.13 | 0.77 | 0.06 | 57.22 | 3       |
| 0.70 | 0.93 | 0.72 | 0.86 | 0.75 | 100   | 1       |
| 0.36 | 0.41 | 0.63 | 0.86 | 0.02 | 57.59 | 3       |

O algoritmo MCDA apresentado nesta seção foi utilizado para classificar o *dataset* contendo o repositório inicial de 100, 200 ou 300 recursos. Após 1000 recursos foram classificados e agrupados de diferentes formas para avaliar o comportamento do algoritmo de aprendizagem de máquina.

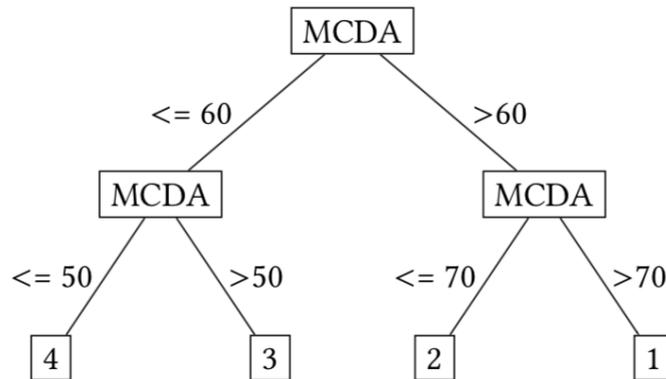
#### 4.2. Explorando aprendizagem de máquina na pré-classificação de recursos

Esta seção apresenta o resultado de 4 avaliações realizadas na pré-classificação de recursos através do algoritmo de aprendizagem de máquina LMT. Foi escolhido o algoritmo LMT por ter obtido a melhor acurácia dentre os algoritmos de árvore de decisão analisados [Dilli et al. 2017]. Em cada avaliação foram classificados 1000 recursos com valores de 1 a 4. Os recursos foram agrupados em lotes de 50, 100, 200 e 500 recursos.

Todos os recursos foram classificados através do algoritmo MCDA e reclassificados de 1 a 4. Essa reclassificação foi usada para treinar o algoritmo de aprendizado de máquina.

Em cada avaliação, o grupo de recursos classificados através do algoritmo MCDA é retirado do conjunto de dados de treinamento e usado no conjunto de dados de teste do algoritmo de aprendizado de máquina.

As avaliações foram realizadas com os seguintes repositórios iniciais para trei-



**Figura 3. Reclassificação do algoritmo MCDA**

namento do algoritmo LMT: (a) - repositório inicial com 100 recursos; (b) - repositório inicial com 200 recursos e (c) - repositório inicial com 300 recursos.

A Tabela 8 apresenta as classificações incorretas e o percentual de acurácia de cada uma das 4 avaliações:

**Tabela 8. Avaliação da Acurácia**

| Qty              | 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  |
| <b>TOTAL</b>     | 232  | 223  | 180 | 228  | 216  | 195  | 258  | 266  | 258  | 322  | 299  | 464  |
| <b>Accuracy%</b> | 76.8 | 77.7 | 82  | 77.2 | 78.4 | 80.5 | 74.2 | 73.4 | 74.2 | 67.8 | 70.1 | 53.6 |

A primeira avaliação (E1) considerou grupos de 50 recursos, totalizando 20 grupos. A segunda avaliação (E2) considerou grupos de 100 recursos, totalizando 10 grupos. A terceira avaliação (E3) considerou grupos de 200 recursos, totalizando 5 grupos. A quarta avaliação (E4) considerou grupos de 500 recursos, totalizando 2 grupos. Para cada grupo avaliado, o algoritmo MCDA reclassifica recursos e treina o algoritmo de aprendizado de máquina.

Ao analisar a acurácia obtida pelo pré-ranqueamento do algoritmo de aprendizado de máquina LMT, podemos observar que a avaliação E1c obteve a acurácia de 82%. O repositório inicial foi definido com 300 recursos que foram usados para treinamento e lotes de 50 recursos para novos treinamentos.

As avaliações E2b e E2c também se mostram satisfatórias, com acurácia de 78.4% e 80.5%. Estas 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 da aprendizagem de máquina mostrou-se muito grande, acarretando a perda de acurácia.

## 5. Trabalhos Relacionados

A Tabela 9 apresenta uma comparação dos principais Trabalhos Relacionados ao EXEHDA-RR, empregando os seguintes critérios: (i) expressividade na representação de recursos; (ii) utilização de algoritmos MCDA; (iii) aplicação das Preferências do Cliente; (iv) emprego de QoS; e (v) uso de lógica fuzzy e (vi) emprego de Aprendizagem de Máquina (AM). A presença do critério está representada pelo caracter “+” e a ausência por “-”.

**Tabela 9. Comparação dos Trabalhos Relacionados**

| Trabalhos | Expr. | MCDA | Pref. | QoS | Fuzzy | AM |
|-----------|-------|------|-------|-----|-------|----|
| Maheswari | +     | +    | +     | +   | +     | -  |
| Salah     | -     | +    | +     | +   | +     | -  |
| Perera    | +     | +    | +     | -   | -     | -  |
| Gomes     | +     | -    | -     | -   | -     | -  |
| Almulla   | -     | +    | +     | +   | +     | -  |
| Nunes     | -     | +    | +     | -   | -     | -  |
| Suchithra | -     | +    | +     | +   | -     | -  |
| Vaadaala  | -     | +    | +     | +   | -     | +  |
| EXEHDA-RR | +     | +    | +     | +   | +     | +  |

A especificação de recursos através de linguagens de alta expressividade é realizada nos trabalhos de [Maheswari and Karpagam 2015], [Perera et al. 2013] e [Gomes et al. 2016].

Somente o trabalho de [Gomes et al. 2016] não considera as preferências do cliente, pois trabalha com dados de contexto.

Critérios de Qualidade de Serviço (QoS) são avaliados nos trabalhos

de [Maheswari and Karpagam 2015], [Salah and Saadi 2016], [Almulla et al. 2015], [Suchithra and Ramakrishnan 2015] e [Vaadaala 2013].

A lógica fuzzy é discutida nos trabalhos [Maheswari and Karpagam 2015], [Salah and Saadi 2016] e [Almulla et al. 2015]. Nenhum deles aplica a lógica fuzzy para resolver a incerteza entre os especialistas.

Pode-se observar que apenas o trabalho de [Vaadaala 2013] utiliza aprendizagem de máquina. Mas com o objetivo de mensurar a acurácia de um único atributo de qualidade com o algoritmo J48. Os autores consideraram os resultados satisfatórios e obtiveram uma acurácia de 63%.

## 6. Considerações Finais

Este artigo apresenta um modelo para classificação de recursos na IoT, denominado EXEHDA-RR. Como principal contribuição do trabalho desenvolvido até o momento, podemos destacar: (i) o uso da lógica fuzzy na definição dos pesos dos atributos e seu uso no algoritmo MCDA para classificar os recursos adequados à solicitação do cliente; e (ii) o uso do algoritmo de árvore de decisão na pré-classificação de recursos, considerando diferentes intervalos de treinamento.

Ao aplicar a lógica fuzzy na resolução de conflitos entre os especialistas na definição de graus de importância para cada atributo de QoS, foi possível promover a adequação dos pesos utilizados no algoritmo MCDA, considerando assim os recursos de melhor qualidade como resultado dos pedidos dos clientes.

Além disso, usando o aprendizado de máquina supervisionado, a partir de um conjunto de dados, classificado pelo algoritmo MCDA, foi possível classificar recursos com considerável acurácia. O algoritmo LMT foi adequado, com até 82% de acurácia na pré-classificação.

O modelo de classificação de recursos proposto considera a dinamicidade da infraestrutura computacional fornecida pela IoT, com um grande número de consumidores e provedores de recursos.

O uso de algoritmos de aprendizado de máquina na pré-classificação de recursos promove uma redução relevante do esforço computacional dos algoritmos MCDA na classificação de recursos para cada solicitação dos clientes. Esta técnica foi modelada empregando tecnologias web semânticas para a especificação dos recursos e seus atributos de qualidade.

A perspectiva com a proposição do EXEHDA-RR é potencializar o processo de classificação de recursos do *middleware* EXEHDA, capacitando o mesmo para um cenário de elevada escalabilidade e dinamicidade quando da composição do ambiente computacional, situação esta típica da infraestrutura provida pela IoT.

Destacamos como trabalhos futuros a participação do usuário na definição dos pesos de importância dos atributos de QoS e a avaliação da acurácia na pré-classificação com *datasets* contendo um número maior de recursos.

## Referências

- Al-Masri, E. and Mahmoud, Q. H. (2007). QoS-based discovery and ranking of Web services. In *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, pages 529–534.
- Almulla, M., Yahyaoui, H., and Al-Matori, K. (2015). A new fuzzy hybrid technique for ranking real world Web services. *Knowledge-Based Systems*, 77:1–15.
- Argou, A., Dilli, R., Reiser, R., and Yamin, A. (2017). Lógica Fuzzy na Definição da Importância dos Atributos de QoS na Seleção de Recursos da IoT. In *IV Workshop-Escola de Informática Teórica*, pages 173–180.
- Davet, P. T. (2015). Uma Contribuição à Inclusão Digital dos Métodos de Pesquisa Agropecuário no Cenário da IoT.
- Dilli, R., Filho, H. K., Pernas, A. M., and Yamin, A. (2017). EXEHDA-RR: Machine Learning and MCDA with Semantic Web in IoT Resources Classification. *WebMedia '17, October 17–20, 2017, Gramado, Brazil*.
- García, J. M. (2012). *Improving Semantic Web Services Discovery and Ranking*. PhD thesis, University of Seville.
- Gomes, P., Cavalcante, E., Batista, T., Taconet, C., Chabridon, S., Conan, D., Delicato, F., and Pires, P. (2016). A QoC-Aware Discovery Service for the Internet of Things. *Ubiquitous Computing and Ambient Intelligence*, 7656:344–355.
- IHS Markit (2017). The Internet of Things : a movement , not a market. Technical report.
- Maheswari, S. and Karpagam, G. R. (2015). Comparative analysis of semantic web service selection methods.
- Perera, C., Zaslavsky, A., Christen, P., Compton, M., and Georgakopoulos, D. (2013). Context-aware sensor search, selection and ranking model for internet of things middleware. In *Proceedings - IEEE International Conference on Mobile Data Management*, volume 1, pages 314–322.
- Salah, N. B. and Saadi, I. B. (2016). 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*, pages 171–179.
- Schröpfer, C., Schönherr, M., Offermann, P., and Ahrens, M. (2006). A Flexible Approach to Service Management-Related Service Description in SOAs. In *CEUR Workshop Proceedings*, volume 234.
- Suchithra, M. and Ramakrishnan, M. (2015). A survey on different web service discovery techniques.
- Vaadaala, V. (2013). Classification of Web Services Using JForty Eight. *A National Conference on Recent Trends in Computer Science and Technology International Journal of Electronics Communication and Computer Engineering*, 4(6):181–184.

# EXEHDA-SO: Uma Abordagem Ontológica para Ciência de Situação Aplicada a Segurança da Informação

Diórgenes Y. L. da Rosa<sup>1</sup>, Roger S. Machado<sup>1</sup>, Ricardo B. Almeida<sup>1</sup>,  
Adenauer C. Yamin<sup>1</sup>, Ana Marilza Pernas<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Computação  
Universidade Federal de Pelotas (UFPEL), Pelotas – RS – Brasil

{diorgenes, rdsmachado, rbalmeida, adenauer, marilza}@inf.ufpel.edu.br

**Abstract.** *The typical infrastructures of Ubiquitous Computing assume characteristics of flexibility regarding the connectivity in the environments. Aiming security in these scenarios, several solutions are deployed in its own syntax languages, providing events in different formats. In this sense, Situation Awareness, as a strategy capable of integrating events from different sources, becomes a requirement for the controls implementation. This work proposes an ontological approach to SA applied to the information security domain, called EXEHDA-SO. The proposal was evaluated based on a real infrastructure, showing itself capable of handling heterogeneous events from different contextual source.*

**Resumo.** *As infraestruturas típicas da Computação Ubíqua pressupõem características de flexibilidade quanto à conectividade nos ambientes. Visando segurança nestes cenários, diversas soluções são implantadas em linguagens de sintaxe própria, provendo eventos em formatos distintos. Neste sentido, a Ciência de Situação (CS), enquanto estratégia capaz de integrar eventos de diferentes fontes, torna-se requisito para a implementação de controles. Este trabalho propõe uma abordagem ontológica para CS aplicada ao domínio de segurança da informação, denominada EXEHDA-SO. A proposta foi avaliada com base em uma infraestrutura real, se mostrando capaz de tratar eventos heterogêneos provenientes de diferentes fontes contextuais.*

## 1. Introdução

Para atender as prerrogativas de Computação Ubíqua [Weiser 1991], tarefas e funcionalidades cotidianas geralmente contam com algum tipo de conectividade, impondo um ambiente flexível e atento a constantes mudanças. Em se tratando de redes de computadores, o foco nestas premissas remete a uma infraestrutura permissiva, isto é, com o menor número possível de bloqueios e controles. Entretanto, reduzir o número de controles pode expor as informações que trafegam em rede. Estes riscos potencializam-se pelo aumento de conexões simultâneas e conseguinte elevação do tráfego de rede, bem como devido ao aumento na complexidade das conexões com a utilização de diversas portas de conexão, expondo o ambiente a ocorrência de ciberataques.

Em atenção a estes problemas verifica-se um uso recorrente de soluções de propósito específico no domínio de Segurança da Informação (SI), utilizando diferentes formatos para eventos. O emprego destas diferentes soluções de SI trazem empecilhos

quanto à integração de diferentes contextos, dificultando uma visão unificada do ambiente. E as atuais soluções usadas para correlação de eventos em SI costumam utilizar linguagens com sintaxes próprias, o que dificulta sua reutilização e não propicia aproveitamento compartilhado das constantes evoluções nas regras de correlação.

Considerando este panorama, a Ciência de Situação (CS) traz consigo estratégias para tratamento deste problema, pois é construída sobre uma visão ampla a respeito do contexto do ambiente. A CS fomenta a identificação de conjunturas complexas, determinando assim auxílio importante aos analistas de segurança nas tomadas de decisão para proteção da infraestrutura. Dey (1999) afirma que um sistema é ciente de contexto se este utiliza contexto para fornecer informações ou serviços relevantes para o usuário, onde a relevância depende da tarefa do usuário.

Posto este alinhamento, o presente trabalho propõe uma solução para CS por intermédio de uma abordagem baseada no uso de ontologias. Estas estruturas proveem entendimento compartilhado e processável sobre um domínio de conhecimento, podendo especificar relações semânticas entre diferentes situações e identificando cenários de alto nível. O modelo ontológico concebido, denominado EXEHDA-SO *Execution Environment for Highly Distributed Applications - Security Ontology*, contempla conceitos de SI e informações sobre a arquitetura e os ativos que suportam as funcionalidades de um ambiente típico em UbiComp. Este foi testado e avaliado em ambiente simulado, alusivo à infraestrutura computacional da Universidade Federal de Pelotas. No desenvolvimento, foram integrados conceitos já consolidados pelos trabalhos prévios do grupo de pesquisa LUPS (Laboratory of Ubiquitous and Parallel Systems) [Lopes et al. 2014] e [Almeida 2016].

A próxima seção detalha a proposta em seus fragmentos ontológicos, a seção 3 trás um cenário de uso no qual a proposta foi aplicada, a seção 4 aborda alguns trabalhos relacionados e por fim apresentam-se as conclusões na seção 5

## 2. EXEHDA-SO

A EXEHDA-SO consiste em uma abordagem ontológica, capaz de considerar informações de diversas fontes no intuito de gerar inferências que possam auxiliar na detecção de situações de interesse para aprimoramento da segurança do ambiente ubíquo monitorado. Este trabalho constitui colaboração ao *middleware* EXEHDA [Lopes et al. 2014] e derivado trabalho voltado à SI desenvolvido em [Almeida 2016], integrando às funcionalidades de identificação de situações de interesse, focada em ampliar as possibilidades da camada de compreensão de CS. A proposta EXEHDA-SO é apresentada em sua totalidade em [da Rosa 2017].

A Figura 1 apresenta uma visão abrangente da solução proposta. Nesta, trabalhos prévios do grupo estão refletidos, sendo proposto o tratamento de contexto em um Repositório Híbrido de Informações Contextuais [Machado et al. 2017] e o processamento de eventos por meio de regras [Almeida 2016]. O EXEHDA-SO propõe o Processamento Híbrido de Eventos, a partir do desenvolvimento de um modelo ontológico capaz de prover semântica, interoperação e flexibilidade ao método previamente proposto.

Para concepção da EXEHDA-SO foram definidos três fragmentos ontológicos (Core, Scope Analiser e InterCell Analiser), cujas principais classes podem ser visualizadas na Figura 2. As próximas subseções descrevem cada um dos fragmentos.

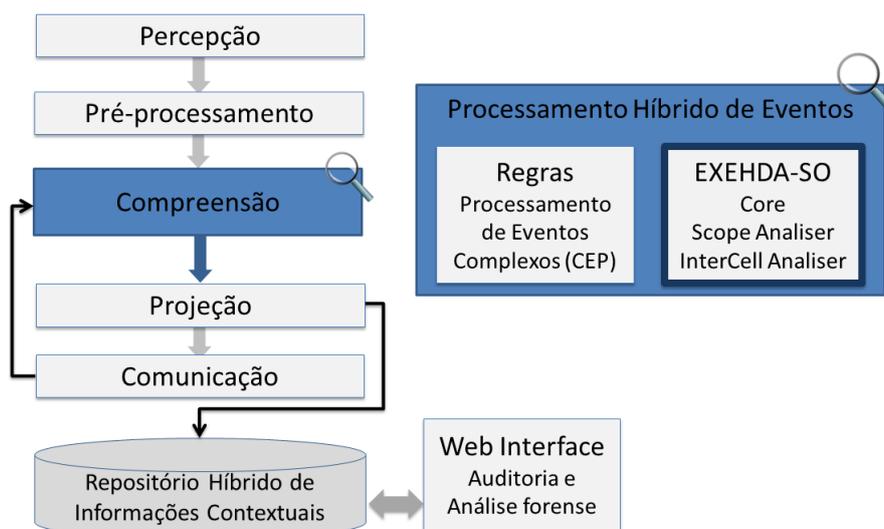


Figura 1. EXEHDA-SO integrando a etapa de Compreensão em CS

## 2.1. EXEHDA-SO: Core

O fragmento ontológico denominado *Core* visa representar conceitos gerais de SI e do próprio *middleware* EXEHDA, no sentido de transversalizar uma base de conhecimento em apoio as funcionalidades de CS tratadas ao longo do modelo como um todo. Este fragmento, bem como os demais, consiste em uma ontologia tipificada como Leve. Não busca-se representar a completude de conceitos estabelecidos em bibliotecas do domínio de SI, refletindo sim apenas a abstração necessária para as demandas do projeto.

Este fragmento conta com a classe *Fundamentals* que remetem aos 5 aspectos teóricos centrais de SI destacados pela ISO 27002<sup>1</sup>: *Confidentiality, Integrity, Availability, Authenticity, Non-repudiation*. Por intermédio da propriedade *aimstoEnsure* pontua-se a classe *Control* que visa assegurar os fundamentos de SI. De forma geral, os cenários de interesse são caracterizados no momento em que confirma-se ou existe a suspeita de que estão ocorrendo eventos voltados à perda de algum destes aspectos de SI.

Os ativos que compõem o ambiente ubíquo, desde o mais básico sensor até servidores de aplicação, são passíveis de ataques e são representados no modelo pela classe *EXEHDA Node*. Como subclasses desta representação tem-se (i) *EXEHDA mob Node* englobando dispositivos móveis, equipamentos específicos de IoT ou sensores diversos, (ii) *Station* para as estações de trabalho não móveis, (iii) *Border Server* para servidores distinguindo instalações nativas e virtuais. Por intermédio das propriedades de dados das instâncias da classe *EXEHDA Node* é possível estabelecer um perfil de execução que dita as regras pelas quais aquele ativo é submetido em termos de conectividade com a rede interna e externa, sistema operacional, hardware disponível, entre outros.

A classe *Control* representa toda a contramedida que pode ser adotada em prevenção ou correção a um determinado cenário de ataque identificado. Esta classe descreve requisições de mudança no ambiente como sendo (i) *Preventive*, ao estabelecer controle voltado a alguma atuação que possivelmente pode ocorrer no ambiente observando comportamentos passados; (ii) *Reductive*, quando mesmo sem garantir a não

<sup>1</sup>ISO 27002: boas práticas para a gestão de segurança da informação

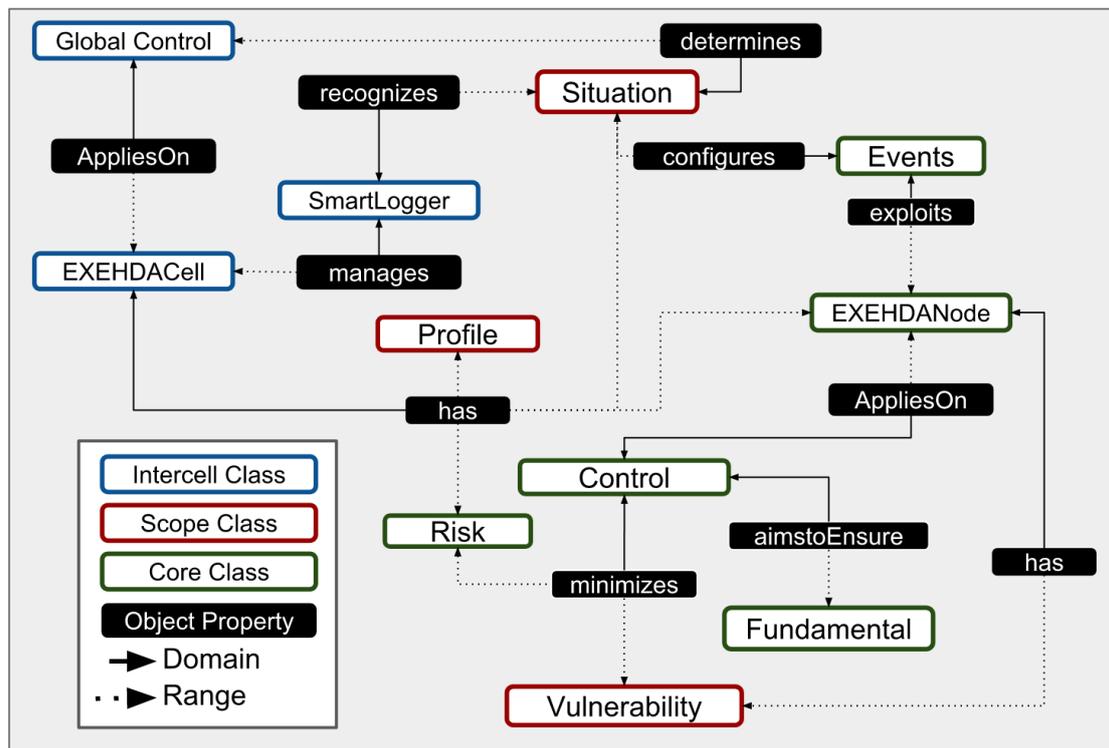


Figura 2. Principais classes da EXEHDA-SO

ocorrência de determinada ação há uma diminuição das situações, minimizando as falhas decorrentes; (iii) *Repressive*, que são medidas que neutralizam as ocorrências maliciosas e (iv) *Corrective* quando são necessários ajustes estratégicos para a recuperação de falhas. Considerando que a aplicação destes controles pode impactar o ambiente de alguma maneira, são estipuladas ações ativas e passivas. Por ação ativa entende-se toda a ação voltada à proteção ao ambiente executada de forma automática, quando a criticidade do cenário impõe isto. O bloqueio de uma porta de comunicação por intermédio de um *script* personalizado que interage com o *Firewall* exemplifica um controle ativo. Já os controles passivos são aqueles que apenas instrumentalizam a análise de um gestor de redes. O envio de um e-mail é considerado uma ação passiva, pois não interage diretamente com os perfis de execução do ambiente de forma automática.

Para a classe *Risk* são avaliados tanto o impacto que a exploração de determinado ativo pode representar quanto a probabilidade com que esta exploração ocorra. Cenários percebidos tem seus riscos determinados por dados contextuais que podem ser incluídos inclusive na fase de pré-processamento, como por exemplo (i) severidade da situação (ii) prioridade do evento e (iii) criticidade do ativo, sendo tratados como propriedades de dados. Como subclasses de *Risk* foram definidas: *Human Actions*, *System and Technology Failures*, *External Factors*.

Todo o evento percebido que possa impactar algum dos fundamentos básicos de SI estabelecidos no modelo é representado pela classe denominada *Events*. Esta é uma das classes principais do modelo pois a partir dos eventos, geralmente estruturados em logs de serviços ou soluções de segurança, encaminha posterior identificação de situações de interesse. Cada instância desta classe trás consigo uma série de predicados, materi-

alizando dados contextuais que definem o evento segundo aspectos técnicos relevantes, como: endereçamentos IP's e *HostName* de origem e destino do evento, o serviço alvo, por exemplo. Por intermédio dos predicados das instâncias desta classe é possível verificar o método utilizado pelos atacantes, sua estratégia e intenção. Identifica-se assim se o evento refere-se a, por exemplo, um escaneamento de portas (atividade tipicamente preliminar a ataques), tentativa de entrega de *malware's*, tentativas de acesso ilegítimo, entre outros. Conforme definido pela relação *exploits*, os eventos são ações maliciosas destinadas a *EXEHDA Node's*, sendo um vocábulo estipulado para ataques ou ações tipicamente preliminares à ataques.

## 2.2. EXEHDA-SO: *Scope Analyzer*

A concepção do fragmento ontológico *Scope Analyzer* define as representações necessárias às observações das situações de interesse do ambiente ubíquo no que diz respeito às primeiras fases de processamento ontológico. O seu foco central está no reconhecimento de contextos específicos de segurança na célula de execução atribuída a ele.

O processamento dos cenários já nas células aproxima a correção dos nodos envolvidos no que tange à conectividade, isto é, a correção é aplicada sem percorrer outros níveis da arquitetura. Esta alternativa também auxilia a questão de escalabilidade no tratamento dos eventos ao distribuir o processamento entre setores distintos do ambiente.

As relações entre as demais classes auxiliam a definição de vulnerabilidades, representada pela classe *Vulnerability* do modelo. Neste sentido destaca-se que um controle (classe *Control*) tem por intenção a minimização de uma vulnerabilidade, podendo inclusive eliminá-la protegendo assim os *EXEHDA Node's* alvo que eventualmente possuam a vulnerabilidade em questão.

Relaciona-se com *EXEHDA Node* a classe denominada *Profile*. Esta representa as configurações vigentes empregadas aos recursos computacionais, caracterizando regras de conectividade na subclasse *Network* e especificando quais serviços estão ativados em *Services*. Mais especificamente, a subclasse *Network* disponibiliza padrões permissivos ou restritivos, observada a relevância e criticidade do equipamento em questão.

A classe *Events*, descrita no fragmento *Core*, possui considerável relevância neste fragmento, uma vez que são aceitos dados oriundos de fontes distribuídas e heterogêneas. Com isso, ao identificar padrões de interesse com base nas propriedades de dados são então configuradas as situações, representadas pela classe *Situation*, as quais por sua vez determinam o uso de um determinado controle. A classe *Situation* pode ser compreendida como um repositório de cenários de interesse.

## 2.3. EXEHDA-SO: *InterCell Analyzer*

Observando a larga conectividade e distribuição dos ambientes ubíquos pode-se considerar que uma análise de eventos individualizada das células de execução não é capaz de gerar, por si só, uma visão global do ambiente quanto a situações de SI. Ou seja, é necessário avaliar cenários de múltiplos níveis da arquitetura valendo-se assim do cruzamento de informações entre células distintas. Desta forma, a intenção do fragmento *InterCell Analyzer* é prover uma visão unificada do ambiente, recebendo ocorrências situacionais de múltiplas células e inferindo controles, considerando que a ação pode ser

efetuada em todo o ambiente. No caso, o *InterCell* recebe os dados tratados no fragmento *Scope Analyzer* adicionando alguns conceitos necessários a ações entre células.

A classe *EXEHDACell* é responsável pelo conhecimento referente às células que compõem o ambiente ubíquo, em seus diversos níveis. Esta classe é instanciada por informações sobre os ativos presentes naquele parque computacional por intermédio da sua relação com a classe *EXEHDANode*. É então obtido um panorama geral do funcionamento das células por intermédio dos equipamentos que regem as conexões. Já na subclasse *level*, específica desta classe, é delimitada a abrangência daquela célula no que se refere a questões organizacionais e até mesmo disposição geográfica.

O *SmartLogger*, componente arquitetural responsável pela primeira frase de processamento ontológico, é representado por uma classe de mesma nomenclatura. Assim a *InterCell* reconhece a gestão deste componente considerando subclasses que trazem dados sobre as técnicas de coleta utilizadas, quais são os sensores disponíveis, quais são os *scripts* personalizados existentes, entre outros. Esta classe também indica as *EXEHDACell's* nas quais o *SmartLogger* efetua o gerenciamento, podendo ser em um ou mais níveis da arquitetura.

Neste fragmento novamente a classe *Situation* está presente, contudo desta vez incorporando informações contextuais sobre a localização onde a situação ocorreu. Observar as situações instanciadas por fragmentos ontológicos dispostos em setores periféricos da arquitetura no *InterCell Analyzer* serve como gatilho preventivo a situações identificadas em outras células. Desta forma, identifica-se a propriedade de objetos *determines*, ligando estas situações aos *Global Controls*. Diferente da classe *Control* do fragmento *Scope*, esta classe provê contramedidas observando todas as células. Com esta ação transversalizada no ambiente, protegem-se serviços e funcionalidades passíveis dos mesmos cenários, mesmo que as ocorrências ainda não tenham explorado àquele ambiente.

### 3. Cenário de Uso

Proteger o ambiente a partir da análise de origem e destino dos eventos detectados é um dos eixos centrais desse cenário de uso. Verificar se os eventos de ataque são oriundos de apenas um ou mais *hosts* trás indícios da abrangência necessária das contramedidas. Isto é, se a origem dos eventos coletados, verificada na ontologia pela propriedade *eventSrcIP*, for recorrente para um ou mais destinos é interessante que o analista considere especificamente o bloqueio daquele agente seguindo indicação de contramedida documentada na classe *Controls* da ontologia. Por outro lado, se os eventos refletem a ação de muitas origens (novamente com base no *eventSrcIP*) em um só destino (considerando o campo *eventDstIP*), a contramedida pode considerar o sugestionamento ao analista de segurança para avaliar a possibilidade de diminuir as possibilidades de ataques no servidor de borda em questão (desativando serviços, limitando o acesso aos mesmos apenas pela rede interna, entre outras alternativas que objetivem a mitigação do risco).

Para demonstrar esta identificação, primeiramente é realizada a coleta do log oriundo do HIDS (*Host-based Intrusion Detection System*) OSSEC (*Open Source Security*) que está configurado em modo *Standalone* em um servidor, nomeado *webserver2*. Observa-se que um evento similar é registrado em um segundo servidor, *webserver1*, diferindo apenas o IP e *hostname* do servidor de destino e o momento da ocorrência.

Este evento reflete um ataque de Força Bruta, que consiste em uma ação maliciosa

na qual é realizada a tentativa de descoberta de credenciais de um determinado sistema ou serviço, geralmente por intermédio de um dicionário (*wordlists*) ou por combinação de caracteres. Para execução do ataque em questão foi utilizada a ferramenta THC-Hydra<sup>2</sup> disponível nativamente na distribuição linux Kali<sup>3</sup>.

Os *logs* que caracterizam os eventos de Força Bruta são então tratados pelo componente *Collector* dos servidores. Este estágio de pré-processamento vale-se inicialmente do Filebeat que lê *logs* oriundos de diversos sensores do ambiente, inclusive do HIDS OSSEC. O Filebeat encaminha então os *logs* recebidos ao Logstash, controlando o fluxo deste encaminhamento para que não ocorra sobrecarga, provendo métodos de retomada do encaminhamento em caso de falha do Logstash e criptografando esta comunicação. No Logstash estes dados são normalizados utilizando expressão regular.

Após executadas estas tarefas, os eventos são instanciados na classe *Events* da ontologia. Observa-se que nem todos os campos são encaminhados para a ontologia, sendo utilizados apenas os dados relevantes para o processamento ontológico deste caso. Para efetuar o encaminhamento dos eventos na ontologia foi utilizada a API Java Jena. Neste ponto, com os eventos já instanciados na classe estipulada *Events*, tem-se o conhecimento de diversos fatores fundamentais de uma ação maliciosa como: (i) quais foram os endereçamentos de origem e destino do evento; (ii) o serviço alvo; (iii) o usuário que está sendo utilizado pelas tentativas de acesso; e (iv) a prioridade do evento.

A abordagem para esta identificação de cenário primeiramente separa os eventos por intermédio do predicado destino denominado *eventDstIP* que informa o endereço IP alvo. Assim, na simulação realizada, todos os eventos destinados para o webserver1 são encaminhados para a classe que o representa na hierarquia de ativos (*EXEHDA\_Node*). Esta regra para um servidor web pode ser visualizada na Figura 3. A regra repete-se para cada servidor previsto no ambiente.

```
exehdaso:Events(?x) ^ exehdaso:eventDstIP(?x, ?a) ^ swrlb:equal(?a,
"192.168.0.1") ^ exehdaso:eventName(?x, ?b) ^ swrlb:equal(?b,
"bruteforce") -> exehdaso:BF_WS_1(?x)
```

**Figura 3. Regra de encaminhamento dos eventos para a classe que representa o ativo alvo**

A tarefa de perceber que muitos eventos apresentam uma mesma origem para mais de um destino no ambiente é executada pela regra da Figura 4. Nesta regra SWRL, a origem dos eventos identificados nos dois servidores web é comparada e apresentaram a situação de interesse. Estes eventos são então instanciados na classe *Situation*, a qual por sua vez determina os controles a serem identificados por meio da classe *Control*.

A partir deste momento, a classe *Situation* é instanciada com os eventos cujas propriedades de dados informam quais endereços IP's estão envolvidos no cenário e a ação, que implica a necessidade de adaptação do ambiente ubíquo, neste caso, promovendo o bloqueio do IP do atacante. Neste momento, é inferido que a recorrência deste atacante

<sup>2</sup><https://www.thc.org/thc-hydra/>

<sup>3</sup><https://www.kali.org/>

```

exehdaso:BF_WS_1(?x) ^ exehdaso:BF_WS_2(?y) ^ exehdaso:eventSrcIP
(?x, ?a) ^ exehdaso:eventSrcIP(?y, ?b) ^ swrlb:equal(?a, ?b) ^
exehdaso:eventDstIP(?x, ?c) ^ exehdaso:eventDstIP(?y, ?d) ^
swrlb:notEqual(?c, ?d) -> OSND(?x) ^ OSND(?y)

```

Figura 4. Regra para identificação de origens únicas em muitos destinos

no ambiente indica a possibilidade do mesmo realizar novas tentativas de acesso indevido nesta célula. Scripts personalizados são executados para efetuar a ação na fase Projeção de CS.

A Figura 5 mostra as classes do modelo ontológico aplicadas neste cenário. A completez dos testes e cenários abordados pode ser vista em [da Rosa 2017].

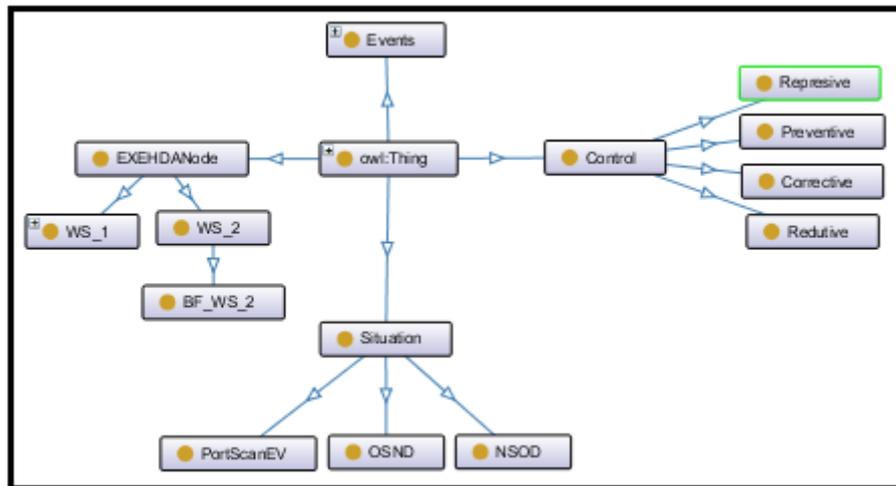


Figura 5. Classes envolvidas no processo de análise de endereçamento

#### 4. Trabalhos Relacionados

Nesta seção estão destacados trabalhos que utilizam ontologias visando proporcionar algum nível de ciência de contexto, limitando o escopo do levantamento a trabalhos que tenham a SI como artifício ou finalidade.

No trabalho [Ficco and Romano 2011] é apresentado um sistema com foco no diagnóstico e na detecção de intrusão, implementando uma correlação de processos híbrida e hierárquica para a detecção de cenários de intrusão. A capacidade de correlação de indícios de ataques é dirigida por uma base de conhecimento ontológica, capturando a relação causal entre as atividades maliciosas preliminarmente detectadas. A ontologia é composta por diversas entidades voltadas ao Monitoramento de Ataques, à Correlação, à Detecção de Intrusão e à Recuperação. Com as informações instanciadas na ontologia, o processamento dos cenários de ataques é efetuado por CEP, correlacionando as situações intermediárias encontradas. Uma vez detectados os cenários de ataque são então encaminhados alertas ao chamado agente remediador, vocábulo alusivo ao analista de sistemas. Os registros utilizados na prototipação são oriundos do IDS *Prelude*.

O trabalho [Bhandari and Gujral 2014] pontua a proposta voltada para a etapa de percepção de situações, sendo concebida uma ontologia. Esta ontologia considera os conceitos estabelecidos pelas taxonomias CVE (*Common Vulnerability Enumeration*), CWE (*Common Weakness Enumeration*), e CAPEC (*Common Attack Pattern Enumeration and Classification*), caracterizando o reuso da proposta. Na ontologia estão representadas as classes: Ator; Ataque; Rede; Vulnerabilidade. Ao receber como entrada as vulnerabilidades oriundas de uma ferramenta externa, a ontologia determina ou percebe o estado da rede, que pode ser “seguro”, “vulnerável” ou “inseguro”.

No trabalho [Azkia et al. 2014] é proposta a utilização do *middleware Adapter* que tem por finalidade efetuar o mapeamento entre os dados de log armazenados e um modelo ontológico, bem como prover uma integração fácil para com a estrutura de logs por intermédio de consultas SPARQL. A ontologia é composta basicamente por 3 elementos: (i) uma terminologia usada no domínio de *Healthcare*; (ii) uma estrutura de logs oriundos dos sistemas; e (iii) uma política de segurança. A terminologia de *Healthcare* mune a ontologia de classes referentes a conceitos da área médica como “Relatórios de Consulta”, “Dados de pacientes”, etc. Quanto à estrutura de logs, o estudo de caso trabalha com logs ATNA (*Audit Trail and Node authentication*) especificado pelo padrão IHE (Integrating the Health care Enterprise) como entrada. O modelo de política de segurança, um terceiro elemento que compõe a ontologia, é adaptado de OrBAC [Cuppens-Bouahia et al. 2008] e de RBCA [Ferraiolo and Kuhn 1992], modelos genéricos para controle de acesso.

Com intuito de comparar a EXEHDA-SO com os trabalhos relacionados descritos nesta seção, apresenta-se a tabela 1. Nos quesitos “formalismo”, “consultas” e “interação” foi mantido um alinhamento com os trabalhos relacionados. Entretanto, destaca-se que a EXEHDA-SO não limita-se a instâncias de uma determinada solução, como pode-se ver na coluna Eventos. Neste quesito observa-se que alguns trabalhos relacionados não informam as fontes de seus eventos enquanto outras utilizam apenas uma. A EXEHDA-SO propõe o uso de regras SWRL, as quais são mencionadas em apenas um trabalho relacionado. Registra-se ainda que o uso de CEP explorado no EXEHDA é compatível com a premissa operacional da EXEHDA-SO.

**Tabela 1. Comparação entre trabalhos relacionados e a EXEHDA-SO**

|                        | Formalismo | Consultas | Inferências | Regras     | Repositório     | Eventos  | Interação |
|------------------------|------------|-----------|-------------|------------|-----------------|----------|-----------|
| FICCO; ROMANO, 2011    | -          | -         | -           | CEP        | -               | Prelude  | -         |
| BHANDARI; GUJRAL, 2014 | OWL        | -         | Hermit      | SWRL       | -               | -        | -         |
| AZKIA et al., 2014     | RDF/OWL    | SparQL    | -           | -          | -               | IHE-ATNA | -         |
| EXEHDA-SO              | OWL        | SparQL    | Pellet      | CEP e SWRL | RHIC (Virtuoso) | Diversos | Java Jena |

## 5. Considerações Finais

A adequação a cenários amplamente dinâmicos requer mecanismos voltados a CS, onde contribuições podem ser direcionadas a distribuição da computabilidade, a diversidade de formatos nos eventos de SI e a multiplicidade de situações que podem ser extraídas na combinação de eventos distintos. Por intermédio das ontologias, a EXEHDA-SO contribui para a estratégia de Compreensão de CS por meio do provimento de um vocabulário

que recebe instâncias de dados contextuais de diversas fontes, processando-as de forma descentralizada, unificando formatos e provendo indicações de contra-medidas passivas ou promovendo adaptação do ambiente ubíquo por intermédio de ações ativas.

A continuidade da pesquisa é identificada nos seguintes aspectos: (i) ampliação do número de cenários de uso; (ii) aumento do número de regras aplicadas; (iii) desenvolvimento de um módulo voltado à adaptação e manutenção da ontologia; e (iv) exploração da interação entre as técnicas de CEP e Ontologias.

### Agradecimentos

O presente trabalho foi realizado com apoio da CAPES (Programa Nacional de Cooperação Acadêmica - Procad) e da FAPERGS (Programa Pesquisador Gaúcho - PqG).

### Referências

- Almeida, R. B. (2016). EXEHDA-USM: Uma arquitetura hierárquica multinível consciente de situação aplicada a segurança da informação. Dissertação de mestrado em ciência da computação, Programa de Pós-Graduação em Computação/UFPel.
- Azkiá, H., Cuppens-Bouahia, N., Cuppens, F., and Coatrieux, G. (2014). Log content extraction engine based on ontology for the purpose of a posteriori access control. *IJKL*, 9(1/2):23–42.
- Bhandari, P. and Gujral, M. (2014). Ontology based approach for perception of network security state. In *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*, pages 1–6.
- Cuppens-Bouahia, N., Cuppens, F., de Vergara, J., Vazquez, E., Guerra, J., and Debar, H. (2008). An ontology-based approach to react to network attacks. In *Risks and Security of Internet and Systems, 2008. CRiSIS '08. Third International Conference on*, pages 27–35.
- da Rosa, D. Y. L. (2017). EXEHDA-SO: Uma abordagem ontológica para ciência de situação aplicada ao domínio de segurança da informação. Dissertação de mestrado em ciência da computação, Programa de Pós-Graduação em Computação/UFPel.
- Dey, A. K. and Abowd, G. D. (1999). Towards a better understanding of context and context-awareness. In *HUC '99*, pages 304–307. Springer-Verlag.
- Ferraiolo, D. and Kuhn, R. (1992). Role-based access control. In *In 15th NIST-NCSC National Computer Security Conference*, pages 554–563.
- Ficco, M. and Romano, L. (2011). A generic intrusion detection and diagnoser system based on complex event processing. In *Data Compression, Communications and Processing (CCP), 2011 First International Conference on*, pages 275–284.
- Lopes, J., Souza, R., Gadotti, G., Pernas, A., Yamin, A., and Geyer, C. (2014). An architectural model for situation awareness in ubiquitous computing. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 12(6):1113–1119.
- Machado, R. S., Almeida, R. B., da Rosa, D. Y. L., Lopes, J. L. B., Pernas, A. M., and Yamin, A. C. (2017). Exehda-hm: A compositional approach to explore contextual information on hybrid models. *Future Generation Computer Systems*, 73:1 – 12.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):66–75.

# **Framework de Suporte ao Desenvolvimento e Evolução de Aplicações Auto-Adaptativas em IoT**

**Belmondo R. A. Junior<sup>1,\*</sup>, Tales P. Nogueira<sup>1,†</sup>,  
Marcio E. F. Maia<sup>2</sup>, Rossana M. C. Andrade<sup>1,‡</sup>**

<sup>1</sup>Grupo de Redes, Engenharia de Software e Sistemas (GREat)  
Mestrado e Doutorado em Ciência da Computação (MDCC)  
Universidade Federal do Ceará (UFC) – Fortaleza – Brasil

<sup>2</sup>Universidade Federal do Ceará (UFC)  
Campus Quixadá - Quixadá - Brasil

{belmondorodrigues, tales, marcio, rossana}@great.ufc.br

**Abstract.** *The traditional development approach based on design-time decisions is inadequate for IoT applications considering (i) the heterogeneity of devices and technologies, and (ii) the inability to handle unforeseen scenarios at development time. Therefore, the use of practices that allow adapting the behavior and structure of applications in a more flexible way is required. In this context, we propose a decoupled support framework that allows IoT applications to achieve self-adaptation. The proposed framework was evaluated in relation to its performance and the results show that this approach can perform adaptations with a large number of devices in a timely manner.*

**Resumo.** *O desenvolvimento baseado em decisões em tempo de design é inadequada para aplicações IoT considerando (i) a heterogeneidade de dispositivos e tecnologias e (ii) a incapacidade de lidar com cenários imprevistos em tempo de design. Portanto, práticas que permitam adaptar o comportamento e a estrutura da aplicação de modo flexível são necessárias. Nesse contexto, este trabalho propõe um framework de suporte desacoplado para permitir que as aplicações IoT atinjam a auto-adaptação. O framework proposto foi avaliado em relação ao seu desempenho e os resultados mostram que esta abordagem pode realizar adaptações considerando um grande número de dispositivos em tempo hábil.*

## **1. Introdução**

A crescente miniaturização das tecnologias de computação e pesquisas em áreas como a computação ubíqua permitiram o surgimento da Internet das Coisas (do inglês *Internet of Things* – IoT). Este paradigma consiste em um cenário no qual dispositivos dotados de

---

\*Aluno de mestrado do MDCC/UFC, financiado pela CAPES até 03/2018 e agora financiado pela Fundação Cearense de Pesquisa e Cultura - FCPC

†Bolsista de Pós-Doutorado CAPES

‡Bolsista do CNPq de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora (DT) com o número de processo 314021/2009-4

sensores funcionam em colaboração com atuadores de forma a atingir um determinado objetivo de modo transparente aos usuários [Pires et al. 2015].

Na IoT, os objetivos são atingidos através da execução de ações no ambiente utilizando sensores e atuadores. Para isso, os desenvolvedores utilizam informações de contexto sobre os usuários e sistemas buscando adaptar o comportamento da aplicação para otimizar seu funcionamento ou tratar condições adversas. Nas abordagens tradicionais de desenvolvimento, as ações executadas no ambiente e os estados da aplicação que disparam adaptações (regras de adaptação) são definidos em tempo de desenvolvimento [Lee 2015].

Essa abordagem baseada em decisões em tempo de desenvolvimento é inadequada para a IoT por dois motivos [Baresi and Ghezzi 2010]: i) grande quantidade de dispositivos, tecnologias de suporte e estados da aplicação; e ii) incapacidade de tratar, em tempo de execução, os cenários não previstos em tempo de desenvolvimento. Assim, a criação de aplicações para a IoT requer o uso de práticas que consigam adaptar o comportamento e a estrutura da aplicação de forma mais flexível.

Para tratar cenários com nível de complexidade elevado, abordagens baseadas em auto-adaptação têm sido utilizadas a fim de permitir a flexibilidade do comportamento e estrutura com um mínimo de intervenção do usuário ou desenvolvedor [Kramer and Magee 2007]. Deste modo, a auto-adaptação é baseada na criação de modelos que descrevem o comportamento e a estrutura do sistema e que são traduzidos em módulos de execução que implementam os requisitos de aplicação. Por exemplo, para alterar o comportamento da aplicação, os modelos que descrevem o estado do sistema são analisados, regras de adaptação geradas em tempo de execução e ações de otimização ou correção são implantadas através da substituição de módulos inadequados.

Com o objetivo de utilizar em IoT os conceitos propostos para os sistemas auto-adaptativos, este trabalho propõe um *framework* que permite o desenvolvimento de aplicações para cenários IoT considerando a heterogeneidade e o dinamismo, bem como as possibilidades de adaptação. Para atingir este objetivo, o *framework* orquestra as possíveis adaptações nos níveis de ambiente e de aplicação através de sequências de ações, chamadas de *workflows* neste trabalho.

## 2. Fundamentação Teórica

### 2.1. Internet das Coisas

Internet das Coisas é um cenário no qual dispositivos com capacidade de sensoriamento e atuação – *smart-objects* – produzem e consomem informação e comunicam-se através de redes sem fio com o propósito de atingir determinado objetivo. A literatura define uma série de requisitos para aplicações IoT que serão descritas a seguir.

A **descrição** dos dispositivos consiste em informações que representam as características funcionais e não funcionais de um dispositivo, e os serviços provindos destes [Li et al. 2014]. A **descoberta** prevê que os objetos inteligentes e serviços se tornem disponíveis em tempo de execução [Barreto et al. 2017].

A **segurança** está associada à privacidade, autenticação e controle de acesso. A **privacidade** trata de como os dados presentes no ambiente estarão dispostos para as aplicações. A **autenticação** está relacionada à comunicação segura entre objetos e serviços. Por fim, o **controle de acesso** é um requisito que visa garantir permissões sobre

dispositivos, sensores e serviços [Ng and Wakenshaw 2017]. O **sensoriamento** diz respeito às capacidades dos *smart-objects* de perceberem as variáveis presentes, internas ou externas. Já a **atuação** diz respeito à capacidade dos *smart-objects* exercerem ações no mundo físico [Ng and Wakenshaw 2017]. O **tratamento de exceção** é um requisito para garantir o equilíbrio do sistema de maneira que a execução seja contínua, mesmo diante da indisponibilidade dos *smart-objects* ou serviços envolvidos [Ng and Wakenshaw 2017].

## 2.2. Sistemas Auto-Adaptativos

Os sistemas auto-adaptativos são aqueles capazes de modificar seu comportamento, atributos ou artefatos em resposta a mudanças internas ou externas. O processo de adaptação, conhecido como MAPE-K Loop, consiste de cinco fases: Monitoria (M), Análise (A), Planejamento (P) e Execução (E) e o Conhecimento (K) [Kephart and Chess 2003].

A primeira fase (M), é responsável por perceber e coletar os estímulos e mudanças dos recursos do sistema. Os estímulos são agregados, correlacionados e filtrados até que sejam caracterizados como um fenômeno que necessite ser analisado. A segunda fase (A), é responsável por analisar e inferir a partir das informações fornecidas pela fase anterior, sendo influenciado por conhecimento prévio armazenado. Caso seja percebido que, para o fenômeno detectado, demanda-se alterações, uma solicitação de mudança é passada para a camada seguinte. A terceira fase (P), tem como função estruturar as ações necessárias a fim de atingir um determinado objetivo. Essas ações são criadas ou selecionadas a partir de um conhecimento prévio para iniciar uma alteração desejada no recurso onde o fenômeno foi detectado. Essas alterações podem variar desde uma única ação até um fluxo de execuções. A quarta fase (E) altera o comportamento do recurso com base nas ações recomendadas pela fase anterior. O Conhecimento (K) diz respeito ao conhecimento compartilhado entre as outras fases, como *logs* históricos e sintomas de possíveis problemas, por exemplo.

## 3. Framework Proposto

Como uma alternativa para a problemática apresentada, este trabalho propõe um *framework* que visa apoiar o desenvolvimento de aplicações considerando a dinamicidade de execução em ambientes auto-adaptativos, como ambientes IoT. Para atingir este objetivo, este *framework* foi desenvolvido levando em consideração os requisitos de descrição, descoberta, sensoriamento, atuação e tratamento de exceção detalhados na seção 2.1. Deste modo, este trabalho atua na fase de execução do MAPE-K Loop, especificando ações, estratégias e regras de adaptação.

### 3.1. Visão Geral

A principal característica deste *framework* é o gerenciamento da execução de *workflows* constituídos de ações e seus modos de execução (sequencial ou em paralelo). Ações especificam adaptações a serem executadas, tanto a nível de sistema quanto a nível de ambiente.

A instanciação dos *workflows* pode ocorrer de duas maneiras: (i) simples, quando o *workflow* é executado a partir de uma solicitação feita por um elemento do sistema; ou (ii) baseado em regras, que permite a execução a partir de um determinado estado da aplicação. Na segunda maneira, o desenvolvedor precisa especificar o estado da aplicação

e as ações a serem executadas quando esse estado for detectado. Assim, a execução baseada em regras é a maneira mais flexível de executar um *workflow*, permitindo a auto-adaptação do aplicativo em tempo de execução. Desse modo, um *workflow* pode ser facilmente trocado em tempo de execução, considerando uma mudança de estado ou uma situação não prevista. Essa troca de *workflows* é um passo do MAPE-K Loop descrito na Seção 2.2. Uma camada de inteligência ou um outro sistema devem estar constantemente analisando o estado do sistema e decidindo se o *workflow* atual vai levar o sistema para um próximo estado válido. Essa camada não faz parte do *framework* proposto.

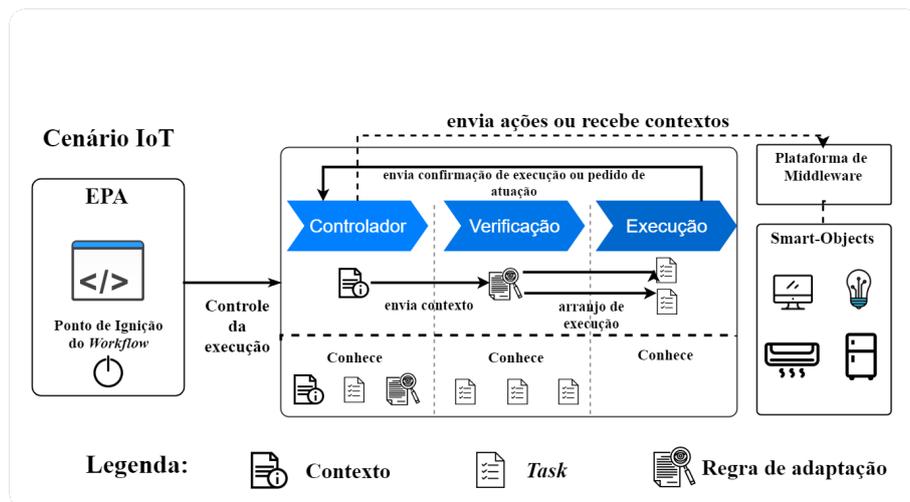


Figura 1. Visão geral do *framework* proposto por este trabalho

O *framework* proposto segue uma ordem que consiste em 3 etapas, conforme ilustrado na Figura 1: (1) Gerenciamento (feito pelo **Controlador**), (2) **Verificação de Regras**, e (3) **Execução**. O **Controlador** conhece os três componentes básicos de execução: estado da aplicação, regras de adaptação e ações. O Controlador monitora continuamente o estado da aplicação e o passa para o segundo módulo. Se qualquer regra for satisfeita, o controlador é responsável pela execução da adaptação associada. O segundo módulo é responsável por verificar se uma ou mais regras estão satisfeitas. Cada regra define um estado da aplicação e as ações que ela deve ativar. Por exemplo, se esta etapa detectar que o usuário está dirigindo, alguma adaptação específica deverá ser executada.

### 3.2. Arquitetura

O *framework* proposto foi projetado em uma arquitetura extensível baseada em componentes, visando melhorar o desacoplamento e o reuso de código. A Figura 2 ilustra as camadas que fazem parte de sua arquitetura: Camada de Desenvolvimento e Camada de Integração e Execução.

A **Camada de Desenvolvimento** contém os módulos de Definição de *Workflow* e o *Workflow Builder*. O Módulo de Definição de *Workflow* é responsável por fornecer as estruturas necessárias para definir o fluxo de adaptações, que podem ser: regras de adaptação, adaptações e o modo de execução (e.g. paralelas, sequenciais ou exclusivas). O módulo *Builder* é responsável por armazenar as estruturas definidas na camada acima e associar as adaptações com suas regras e modos de execução.

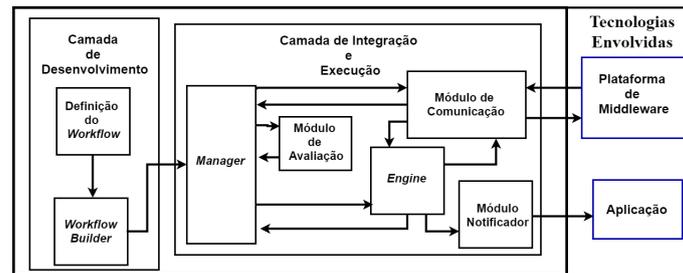


Figura 2. Arquitetura do *framework* proposto neste trabalho

A **Camada de Integração e Execução** é responsável por organizar a execução do *workflow* definido, verificando regras e iniciando adaptações. Ao atingir essa camada, o **Módulo Manager** organiza a execução e distribui as responsabilidades, uma vez que conhece todas as estruturas envolvidas na execução (e.g. regras de adaptação e *gateways*). Ele informa ao **Módulo de Avaliação** os estados da aplicação e ao **Módulo Engine** quais adaptações devem ser executadas e seus modos.

O **Módulo de Comunicação** foi desenvolvido para fornecer uma comunicação para quando for necessário sentir ou agir no ambiente. Portanto, é responsável pela comunicação com quaisquer soluções externas, como plataformas de *middleware*. O **Módulo Engine** é responsável pelo processamento das adaptações exigidas. Além de receber comandos do **Manager**, esta camada é responsável por informar ao **Módulo de Comunicação** sobre as adaptações no ambiente. É, ainda, responsável por enviar informações para o **Módulo Notificador**. A partir daí, o **Módulo Notificador** é responsável por passar as informações para a interface da aplicação ou esperar interação do usuário.

### 3.3. Prova de Conceito

Um Assistente Virtual (AV) Android foi implementado como prova de conceito (PoC) (Figura 3). No cenário proposto, ao chegar ao local de trabalho, o usuário recebe um e-mail de confirmação e o ar condicionado de sua sala é ligado. Quando o usuário entra no prédio e se aproxima da sala, as luzes recebem uma ordem para acendimento. O *framework* adapta a aplicação partindo de regras definidas, como mostra a Figura 4. Para esta PoC, foram definidas duas regras de adaptação: uma baseada na localização do usuário e uma baseada na proximidade da sala. A regra baseada em localização avalia a proximidade geográfica do local de trabalho do usuário, considerando os dados do GPS. A regra baseada na proximidade da sala avalia a distância do usuário e de sua sala, com base em um sinal de *beacon*.

Nesta PoC, foram definidas três ações: a ação responsável por acender as luzes, a ação que confirma a presença do usuário e a ação responsável por ligar o ar-condicionado. Essas ações fazem a transição para o estado de execução apenas quando as regras percebem que é necessário iniciar uma adaptação. Desse modo, as ações permanecem inativas enquanto a regra responsável pela adaptação não seja satisfeita. Os estímulos das regras são as informações contextuais, sejam estas providas do ambiente (físico e lógico) ou do próprio sistema. Como resposta das regras, alguma adaptação pode ocorrer. Na implementação dessa PoC, o *middleware* LoCCAM [Maia et al. 2013] foi utilizado como plataforma de suporte à interação com os *smart-objects* dispostos no ambiente físico.

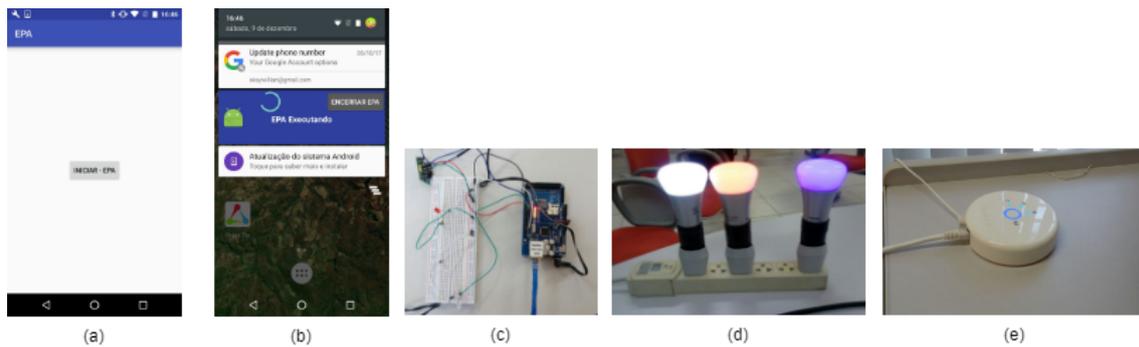


Figura 3. Aplicação e Hardware utilizados na prova de conceito

O fluxo de execução do AV é modificado em tempo de execução (Figura 4). No início, o fluxo permanece inativo, pois as informações contextuais recebidas não atendem às regras definidas (a). Quando uma das regras é satisfeita (b), o fluxo de adaptação relacionado é acionado. Após a sua conclusão, o fluxo retorna ao estado inativo (a). Quando a regra de proximidade é satisfeita (c), o fluxo de adaptações referente torna-se ativo.

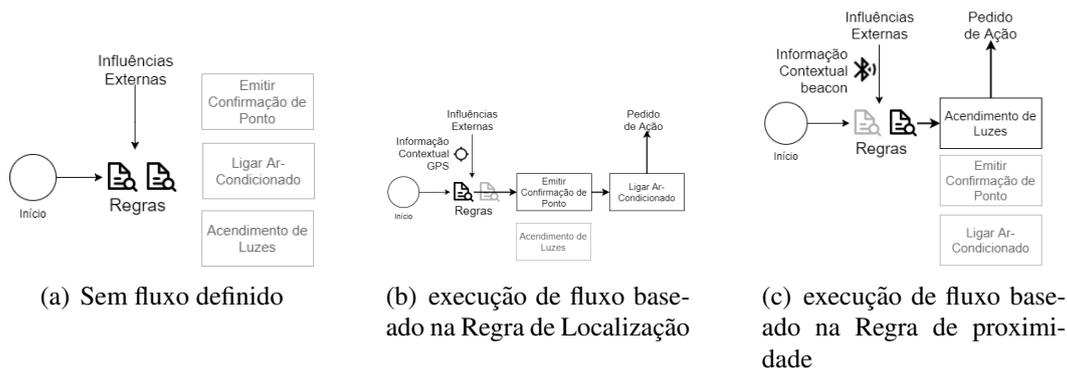
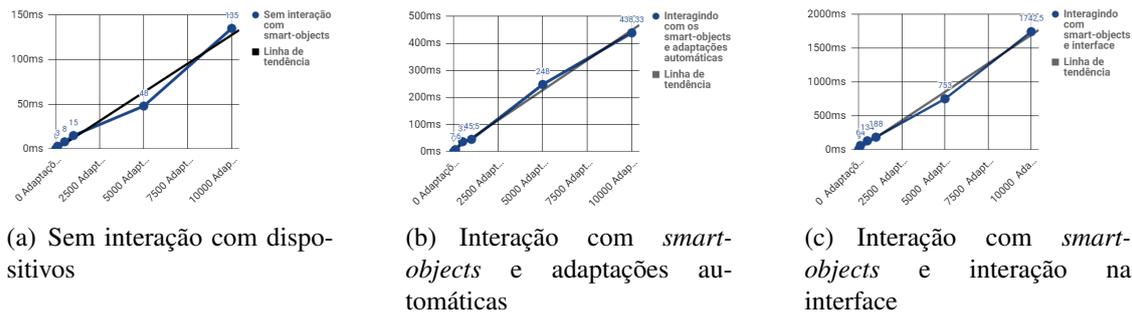


Figura 4. Representação do *workflow* da PoC

#### 4. Avaliação de Desempenho

O *framework* foi avaliado de acordo com o desempenho, a fim de verificar se o mesmo é capaz de realizar adaptações em um cenário heterogêneo com um grande número de estados, regras e possíveis interações com objetos inteligentes. O objetivo desta avaliação foi verificar o comportamento do *framework* proposto em relação ao seu tempo de execução, adaptações instanciadas e completude das adaptações. Para isso, os experimentos foram definidos em três cenários distintos: (a) sem interação com objetos inteligentes, (b) com adaptações automáticas e sem interação com o usuário e (c) com interação e atualização da interface do usuário. Os três cenários foram elaborados para entender o comportamento do *framework* e avaliar se há algum cenário ou número de adaptações que inviabilize o uso desta solução.

Para conduzir a avaliação, foi utilizado um dispositivo Android, versão 6.0 com 1854 MB de memória RAM total e processador Qualcomm 2,27 GHz com 4 núcleos. Como plataforma de *middleware*, foi escolhido o LoCCAM [Maia et al. 2013]. Ainda,



**Figura 5. Avaliações do *framework* proposto neste trabalho**

foi desenvolvida uma biblioteca Java <sup>1</sup> capaz de simular a existência de *smart-objects* e de instanciar adaptações a partir de um número desejado.

O desempenho foi analisado com base no tempo gasto no número de adaptações executadas. Para isso, a avaliação consistiu em cenários com 1, 10, 100, 500, 1.000, 5.000 e 10.000 adaptações. Cada cenário foi executado 10 vezes e foram calculadas as médias de execução em milissegundos. A Tabela 1 e a Figura 5 contêm os resultados desta avaliação.

**Tabela 1. Tabela com os tempos comparativos de execução**

| Nº Adaptações | 1A  | 10As | 100As | 500As | 1.000As | 5.000As  | 10.000As |
|---------------|-----|------|-------|-------|---------|----------|----------|
| Execução (a)  | 0ms | 0ms  | 3ms   | 8ms   | 18ms    | 48ms     | 135ms    |
| Execução (b)  | 0ms | 0ms  | 7,6ms | 37ms  | 65,33ms | 248,33ms | 438,33ms |
| Execução (c)  | 1ms | 9ms  | 64ms  | 134ms | 188ms   | 753ms    | 1742,5ms |

O primeiro cenário (a) foi planejado para entender o tempo de execução basal da solução proposta. Para isso, a simulação consistiu na criação de adaptações "vazias" que geravam mensagens em *logs* para verificação. O segundo cenário (b) foi desenvolvido considerando um ambiente onde a comunicação com sensores e atuadores ocorre de forma transparente para o usuário. O terceiro cenário (c) foi criado para representar um ambiente em que as adaptações interagem com o usuário por meio da interface do dispositivo, como mensagens de confirmação e escolha de opções. A Figura 5 mostra os gráficos para cada experimento. O comportamento do *framework* é representado pela linha azul e a linha preta representa a tendência de execução assintótica.

A principal disparidade ocorreu quando o número de adaptações com interações visuais aumentou. Enquanto o tempo máximo para 10.000 dispositivos foi de 135ms e 438ms nas duas primeiras avaliações, no terceiro cenário esse valor atingiu 1742ms. Isso se deve à arquitetura do sistema operacional Android, já que há apenas uma execução de *thread* principal (a *thread* da interface do usuário). Se é de interesse atualizar a interface do usuário, é necessário que o processo entre na fila de execução da interface do sistema operacional [Google 2017].

Apesar disso, os dados analisados mostram que o *framework* proposto tende a se comportar com uma complexidade linear quando executado em um cenário em que são

<sup>1</sup> <https://github.com/Belmondo/GenerateObjects/>

necessárias um grande número de adaptações. Essa tendência de comportamento sugere que a solução é eficiente se considerarmos que o processamento das adaptações depende do contexto ou do estado da aplicação, conhecendo-os apenas em tempo de execução.

## 5. Trabalhos Relacionados

Nesta seção, são discutidos trabalhos relacionados que propõem algum mecanismo ou infraestrutura de especificação e execução de adaptações que preveja, em sua arquitetura ou meios externos, as características levantadas na seção 2.1.

O *framework* LateVa [Murguzur et al. 2015] trata a variabilidade dentro de linhas de produtos de software dinâmicas (LPSD). Para tal, um modelo base deve ser descrito com as comunalidades e variabilidades de cada família de LPSD, bem como de possíveis pontos de variação. Assim, dependendo da associação do contexto com a descrição da variação, o LateVa seleciona os fragmentos e os executa. Contudo, por seu escopo ser voltado para LSPD, esse *framework* não prevê em seu funcionamento a descoberta de *smart-objects*.

O Presto [Giner et al. 2010] é uma arquitetura desenvolvida para permitir aos programadores sistematizar o desenvolvimento de *workflows* implementados em dispositivos móveis considerando a existência de sensores. Contudo, é necessário descrever cada componente (e.g. dispositivo inteligente) e indicar suas capacidades.

O PROtEUS [Seiger et al. 2015] é um sistema integrado composto por um motor de execução, um mecanismo de processamento de eventos, uma plataforma de serviço, um *caller* de serviço e um servidor. O *core* do PROtEUS consiste de um meta-modelo que descreve a estrutura de um processo e uma *engine* responsável por instanciar esses meta-modelos. O sistema utiliza uma forma hierárquica de representar informações contextuais para capturar e perceber os objetos no ambiente. Contudo, é necessário que o objeto esteja instalado e atribuído no ambiente através da interface de controle e monitoramento.

O SitOPT [Wieland et al. 2015] é um sistema de gerenciamento de *workflows* para sistemas ubíquos. Sua arquitetura é dividida em 3 camadas: (i) *sensing*, (ii) *situation recognition* e (iii) *situation-aware workflow*. Contudo, sua arquitetura não prevê descoberta de objetos e, para possibilitar a modelagem das situações, é necessário descrever cada componente e indicar suas capacidades.

O GET+UNICORN [Baumgraß et al. 2015] é uma abordagem para conectar eventos de várias fontes heterogêneas e execução com foco em transportes. O projeto é dividido em dois sistemas básicos: (i) *Unicorn* e (ii) *GET Controller*. O *Unicorn* é responsável por coletar, processar e distribuir os eventos logísticos relevantes provindos de diversas fontes, enquanto que o *GET* executa e monitora os processos considerando os eventos provindos do *Unicorn*. Contudo, apenas atuação é prevista na abordagem dos autores.

No trabalho de [Montagut and Molva 2005], uma arquitetura de suporte à execução distribuída de *workflows* em cenários pervasivos é proposta: *Enabling Pervasive Workflow Execution* ou EPEW. Mesmo tendo os dispositivos como centro de cada execução, o trabalho não considera as capacidades de atuação.

O *framework* proposto é responsável por orquestrar as possibilidades de adaptação, de ambiente e de aplicação em tempo de execução, considerando os requi-

sitos descritos na seção 2.1. Para isso, tais requisitos são considerados dentro das camadas da arquitetura. O Módulo de Comunicação é responsável pela comunicação com a plataforma de *middleware* desejada pelos desenvolvedores. Portanto, os requisitos de descrição, descoberta, sensoriamento e atuação são considerados aqui. Deste modo, esses requisitos ficam transparentes para as demais camadas. Caso a adaptação esteja associada à aplicação, esta é feita pelo Módulo Notificador. Para o tratamento de exceções em tempo de execução, o Módulo Engine observa as possibilidades de adaptação e, caso uma exceção seja lançada, o fluxo de execução é alterado ou um tratamento pré-definido pelos desenvolvedores é executado. Para este trabalho, não foram considerados os requisitos relacionados à segurança. A Tabela 2 sumariza a discussão apresentada nesta seção.

**Tabela 2. Trabalhos relacionados. Legendas: X - Possui, (X) - Possui Parcialmente e O - Não Possui/Não Descreve**

|                    | Descrição | Descoberta | Relacionados à Segurança | Sensoriamento & Atuação | Exceção |
|--------------------|-----------|------------|--------------------------|-------------------------|---------|
| LateVa             | X         | O          | O                        | (X)                     | O       |
| Presto             | X         | O          | (X)                      | O                       | O       |
| PROtEUS            | X         | (X)        | O                        | X                       | X       |
| SitOPT             | X         | O          | O                        | X                       | X       |
| Get+Unicorn        | O         | O          | O                        | (X)                     | X       |
| EPEW               | O         | X          | (X)                      | (X)                     | O       |
| Framework Proposto | X         | X          | O                        | X                       | X       |

## 6. Considerações Finais

As capacidades de adaptação e evolução em tempo de execução de uma aplicação IoT podem ser comprometidas se considerarmos o modo tradicional de desenvolvimento de aplicações. Visando tratar esse problema, este trabalho apresentou um *framework* para descrição e execução de adaptações baseadas em *workflows* para cenários IoT. O *framework* proposto fornece um conjunto de estruturas de modelagem para especificar ações, estratégias e regras de adaptação. Além disso, permite integrar estes componentes, provendo a auto-adaptação de aplicações em tempo de execução.

O *framework* foi avaliado através de uma PoC e medidas de desempenho. Nesta análise, os resultados mostraram que a solução é vantajosa e pode ser usada em um cenário real. Além disso, esta abordagem se mostrou capaz de realizar e suportar, em tempo hábil, um grande número de adaptações em tempo de execução. Como perspectiva futura, avaliações específicas para sistemas auto-adaptativos devem ser executadas.

## Agradecimentos

Este trabalho foi parcialmente apoiado pelo INCT INES (CNPq/465614/2014-0).

## Referências

- Baresi, L. and Ghezzi, C. (2010). The disappearing boundary between development-time and run-time. In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, FoSER '10, pages 17–22. ACM.
- Barreto, F. M., Maia, M. E., Andrade, R. M., and Viana, W. (2017). Coap-ctx: Extensão sensível ao contexto para descoberta de objetos inteligentes em internet das coisas. *XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC2017*.

- Baumgraß, A., Ciccio, C. D., Dijkman, R. M., Hewelt, M., Mendling, J., Meyer, A., Pourmirza, S., Weske, M., and Wong, T. Y. (2015). Get controller and unicorn: Event-driven process execution and monitoring in logistics. In *BPM*.
- Giner, P., Cetina, C., Fons, J., and Pelechano, V. (2010). Developing mobile workflow support in the internet of things. *IEEE Pervasive Computing*, 9(2):18–26.
- Google (2017). Process and threads. <https://developer.android.com/guide/components/processes-and-threads.html>.
- Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.
- Kramer, J. and Magee, J. (2007). Self-managed systems: an architectural challenge. In *Future of Software Engineering, 2007. FOSE '07*, pages 259–268.
- Lee, E. A. (2015). The past, present and future of cyber-physical systems: A focus on models. *Sensors*, 15(3):4837–4869.
- Li, Q.-s., Chu, H., Xue, B.-y., and Zhang, C. (2014). Semantic-based dynamic positioning mechanism for problem solving in multi-agent systems. *Journal of Central South University*, 21(2):618–628.
- Maia, M. E. F., Fonteles, A., Neto, B., Gadelha, R., Viana, W., and Andrade, R. M. C. (2013). Locom - loosely coupled context acquisition middleware. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 534–541. ACM.
- Montagut, F. and Molva, R. (2005). Enabling pervasive execution of workflows. In *2005 International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 10 pp.–.
- Murguzur, A., Trujillo, S., Truong, H. L., Dustdar, S., Ortiz, , and Sagardui, G. (2015). Run-time variability for context-aware smart workflows. *IEEE Software*, 32(3):52–60.
- Ng, I. C. and Wakenshaw, S. Y. (2017). The internet-of-things: Review and research directions. *International Journal of Research in Marketing*, 34(1):3 – 21.
- Pires, P., Delicato, F., Batista, T., Avila, T., Cavalcante, E., and Pitanga, M. (2015). Plataformas para a internet das coisas. In *Minicursos do XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, chapter Multimodal Interfaces, pages 119–178. Sociedade Brasileira de Computação – SBC.
- Seiger, R., Huber, S., and Schlegel, T. (2015). *PROtEUS: An Integrated System for Process Execution in Cyber-Physical Systems*, pages 265–280. Springer International Publishing.
- Wieland, M., Schwarz, H., Breitenbücher, U., and Leymann, F. (2015). Towards situation-aware adaptive workflows: Sitopt x2014; a general purpose situation-aware workflow management system. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 32–37.

# IMAM - Uma ferramenta para monitoramento inteligente de sistemas e dispositivos em infraestruturas críticas de IoT

David Coelho dos Santos<sup>1</sup>, João Carlos Xavier Júnior<sup>1</sup>,  
Gibeon Soares de Aquino Júnior<sup>2</sup>

<sup>1</sup>Instituto Metrópole Digital (IMD)

<sup>2</sup>Departamento de Informática e Matemática Aplicada (DIMAp)  
Universidade Federal do Rio Grande do Norte (UFRN)  
59078-970 – Natal – RN – Brasil

david.coelho, jcxavier@imd.ufrn.br, gibeon@dimap.ufrn.br

**Abstract.** *Information systems are critical systems therefore they need to be reliable and always available. In this sense, periods of inactivity can be costly, and also have quite significant consequences. Most of the monitoring tools are able to identify faults but not able to predict them nor to support proactive actions. Aiming to provide fault prediction in critical systems, this paper introduces IMAM, an intelligent system monitoring tool capable of applying Machine Learning techniques to extract knowledge from system and devices fault logs in an IoT infrastructure.*

**Resumo.** *É fundamental que sistemas de informação estejam sempre disponíveis e sejam confiáveis. Períodos de inatividade podem custar caro e ter consequências bastante significativas. A maioria das ferramentas de monitoramento conseguem identificar falhas, mas não são capazes de prevêê-las e nem de realizar ações de suporte proativas. Por essa razão, este artigo apresenta o IMAM, uma ferramenta de monitoramento inteligente de sistemas capaz de utilizar técnicas de aprendizado de máquina para extrair conhecimento a partir de logs de falhas de sistemas e dispositivos em uma infraestrutura de IoT.*

## 1. Introdução

Atualmente, devido ao aumento da importância e uso frequente de sistemas informatizados, cresce a preocupação com a manutenção da infraestrutura computacional das organizações, já que este é o principal meio para a disponibilização de acesso aos sistemas e serviços ofertados aos usuários [Silva and Torres 2010]. Nesses ambientes, que geralmente possuem infraestrutura crítica, é fundamental que os sistemas de informação estejam sempre disponíveis e sejam confiáveis. Períodos de inatividade desses sistemas podem custar caro e ter consequências bastante significativas nos mais diversos contextos. Em um sistema de cuidados médicos em saúde (*health care system*), por exemplo, em que os pacientes dependem diretamente do funcionamento adequado e contínuo de sistemas e equipamentos, falhas podem significar vidas. Nesses casos (sistemas críticos), falhas devem ser identificadas e tratadas de maneira rápida e eficiente.

Diante da complexidade envolvida nesses ambientes, falhas podem ocorrer devido aos mais diversos motivos. Por isso, muitas ferramentas de monitoramento são utilizadas

para identificar problemas, e acima de tudo, evitar a indisponibilidade desses sistemas. Contudo, detectar, analisar, e sobretudo obter conhecimento a partir de dados coletados de aplicações em situação de falha, tornam-se muito mais importante à medida que deseja-se prever possíveis novos problemas e realizar ações de suporte proativas visando prover tolerância a falhas em sistemas.

O termo “tolerância a falhas” foi apresentado originalmente por Avizienis [Avizienis 1998] em 1967. Vários autores apresentam suas próprias classificações para as técnicas de tolerância a falhas porém, a mais comum é a classificação em 4 fases de aplicação [Anderson and Lee 1981]: detecção, confinamento, recuperação e tratamento.

Nesse contexto, um sistema de monitoramento e detecção de falhas deve ser confiável e possuir métodos capazes de capturar, armazenar, gerenciar, processar e analisar uma grande quantidade de dados. Visando atacar os desafios dessa área, o principal objetivo desse artigo é apresentar o IMAM, uma ferramenta de monitoramento inteligente de sistemas capaz de utilizar técnicas descritivas de aprendizado de máquina para extrair conhecimento a partir de *logs* de falhas de sistemas em uma infraestrutura de IoT.

O restante deste artigo está dividido em 7 seções. A seção II descreve alguns conceitos teóricos importantes. A seção III relata os trabalhos relacionados. A seção IV ilustra a arquitetura proposta para o IMAM, enquanto que a seção V apresenta o protótipo desenvolvido para validar esta proposta. A metodologia utilizada no experimento preliminar do presente trabalho e uma análise dos resultados são discutidas na Seção VI. Finalmente, a seção VII apresenta as considerações finais deste trabalho e os trabalhos futuros.

## 2. Conceitos Relacionados

Nesta seção são abordados conceitos sobre disponibilidade, ferramentas de monitoramento e técnicas de aprendizado de máquina que são utilizados na concepção, especificação e modelagem do IMAM.

### 2.1. Táticas de Disponibilidade

Antes que qualquer sistema possa agir em relação a uma falha, esta deve ser detectada ou antecipada. As táticas de disponibilidade, portanto, são projetadas para permitir que um sistema preveja a possibilidade de falhas, de modo que um serviço seja entregue e permaneça em conformidade com suas especificações. Assim, cada tática depende essencialmente da detecção de sinais de vida de vários componentes [Bass et al. 2003]. Algumas táticas para detecção de falhas podem ser definidas como:

- **Monitor de Sistema:** é um componente que é usado para monitorar o estado da saúde de várias outras partes do sistema: processadores, processos, memória, entre outros.
- **Ping/echo:** refere-se a um par de mensagens de requisição e resposta (*request/response*) assíncronas trocadas entre os nós. O teste de *ping* é frequentemente enviado ou executado por um monitor de sistema.
- **Heartbeat:** é um mecanismo de detecção de falhas que emprega uma troca periódica de mensagens entre um monitor de sistema e um processo que está sendo monitorado.

- *Timestamp*: é uma tática para detectar sequências de eventos incorretas, principalmente em sistemas distribuídos de transmissão de mensagens.
- *Sanity*: esta tática verifica a validade das operações ou saídas específicas de um componente do sistema. Baseia-se em um prévio conhecimento do design interno e/ou da natureza da informação. É frequentemente empregado em interfaces para examinar um fluxo da informação.

## 2.2. Aprendizado de Máquina - AM

Há uma grande quantidade de algoritmos de aprendizagem de máquinas na literatura e para vários tipos de domínios de problemas. Em geral, esses algoritmos são escolhidos de acordo com o tipo de problema (classificação ou agrupamento). Classificação e Agrupamento (*Clustering*) são tarefas bem conhecidas de aprendizagem de máquinas.

A classificação de dados é o processo de criação de um modelo de previsão a partir de um algoritmo de aprendizagem. O objetivo deste modelo é prever o valor do atributo classe nas instâncias de teste. Por outro lado, Agrupamento procura explorar ou descrever um conjunto de dados. Em geral, não há rótulos pré estabelecidos nos dados, como é caso dos dados oriundos de falhas de sistemas.

Para as tarefas descritivas, serão analisados, com relação à homogeneidade e compacidade, os seguintes algoritmos de Agrupamento (*Clustering*): Expectation-Maximization (EM) [Xu and Wunsch 2005], k-Means [Faceli et al. 2011] e Hierarchical Aglomerativo [Faceli et al. 2011].

Para as tarefas preditivas, serão analisados, com relação à acurácia, os seguintes algoritmos de Classificação: k-NN (k-Nearest Neighbor) [Faceli et al. 2011], MLP (Rede Neural) [Faceli et al. 2011], J48 (Árvore de Decisão) [Faceli et al. 2011], Naive Bayes [Faceli et al. 2011] e SVM (Máquina de Vetores de Suporte) [Mitchell 1997].

## 2.3. Índices de Validação

A qualidade dos grupos ou *clusters* gerados a partir da utilização dos algoritmos de agrupamentos precisa ser validada sob alguns aspectos importante, tais como: homogeneidade e separação. Há vários índices de validação na literatura. Contudo, os mais comumente utilizados são Davies-Bouldin (DB) e Silhueta [Halkidi et al. 2002].

## 3. Trabalhos relacionados

Esta seção apresenta trabalhos relacionados a abordagens, métodos e ferramentas de monitoramento de infraestrutura de IoT que utilizam técnicas de Aprendizado de Máquina (AM).

### 3.1. Soluções de monitoramento em ambientes IoT

Os trabalhos relacionados revelaram que, na maioria dos ambientes baseados em IoT, a disponibilidade das aplicações e dispositivos é fundamental, assim como a necessidade de ferramentas que possam realizar monitoramento e gerenciamento dessa infraestrutura. Portanto, tais ferramentas realizam, basicamente, a coleta de dados relacionados aos recursos computacionais tais como: CPU, RAM e armazenamento. Além disso, são coletadas informações sobre o fluxo de dados na rede de comunicação para fins de detecção de ataques à segurança.

Nesse contexto, Tokuda [Tokuda et al. 2014] propõe o projeto D-Case que realiza monitoramento de redes de sensores ubíquos com detecção rápida de falhas e a recuperação de serviços. Já Yaseen [Yaseen et al. 2017] apresenta um modelo de monitoramento para detecção de ataques de colusão (*collusion attacks*) a partir de experimentos utilizando uma rede de sensores baseada em IoT. A solução utiliza a arquitetura baseada em *fog computing*, que estende a capacidade computacional para a borda da rede.

Através da ferramenta IoT-PIC, Ferrera [Ferrera et al. 2017] apresenta uma solução que realiza o monitoramento, gerência e configuração em tempo real de dispositivos IoT, tais como: gateways, sensores e atuadores. O protocolo XMPP é utilizado como alternativa ao SNMP na gerência e na configuração desses dispositivos. Além disso, o IoT-PIC também permite a descoberta automática de novos componentes e a troca de informações entre objetos IoT.

### 3.2. AM em sistemas de monitoramento

No trabalho de Chen [Chen et al. 2015], os autores propõem um sistema de monitoramento e detecção de ameaças em rede que utiliza tecnologias como Hadoop, MapReduce e Spark para auxiliar o processamento dos fluxos de dados em conjunto com os algoritmos k-Means (agrupamento) e Naíve Bayes (seleção de atributos). De acordo com os autores, o algoritmo k-Means é utilizado para separar os dados (*logs*) em grupos distintos, e dessa forma, poder identificar comportamentos anômalos na rede. Mas, nenhum mecanismo de tolerância a falhas foi apresentado no mesmo.

Suthaharan [Suthaharan 2014], por sua vez, discute os problemas e desafios em *Big Data* e na predição de intrusões em rede de computadores através do uso de técnicas de aprendizado de máquina. De acordo com o autor, através do uso de *logs* e de técnicas de Aprendizado de Máquina é possível prever possíveis ataques às redes, mesmo em se tratando de grandes volumes de dados. Porém, para o autor, a escolha do modelo deve ser muito cautelosa. Novamente, nenhum mecanismo de tolerância a falhas foi apresentado no mesmo.

Por último, no trabalho de Soysal [Soysal and Schmidt 2010], os autores empregam três algoritmos de aprendizado de máquina supervisionados, redes bayesianas, árvores de decisão e Perceptrons multicamadas para a classificação de seis tipos diferentes de tráfego de Internet. Embora aplique algoritmos para predição, o mesmo não faz predição de falhas e nem apresenta algum mecanismo de tolerância a falhas foi apresentado no mesmo.

De forma geral, a maioria dos trabalhos faz uso de *logs* decorrentes do monitoramento executado. Contudo, somente a minoria utiliza tais *logs* como subsídio para algum mecanismo que possa deixar o ambiente mais seguro. Dentre os trabalhos que utilizam os *logs* para esse propósito (segurança), apenas um faz uso de algoritmos descritivos para identificar comportamentos anômalos na rede, enquanto que um outro utiliza algoritmos preditivos para prever possíveis ataques às redes. Logo, diferentemente dos trabalhos que utilizam *logs* e técnicas de Aprendizado de Máquina, este trabalho propõe a utilização de ambas as técnicas descritiva e preditiva para a detecção e recuperação de falhas em sistemas e dispositivos em ambiente de IoT.

## 4. Arquitetura do Sistema

A ferramenta IMAM é um *middleware* capaz de realizar análises em registros de logs de monitoramento de aplicações e dispositivos em ambientes IoT. É composto por 4 componentes principais, são eles: Coletor de Dados de Monitoramento (CDM), Driver, Analisador de Cenários de Monitoramento (ACM) e um Web Dashboard para oferecer ao usuário acesso a configurações, dados e análises, conforme mostrado na Figura 1.

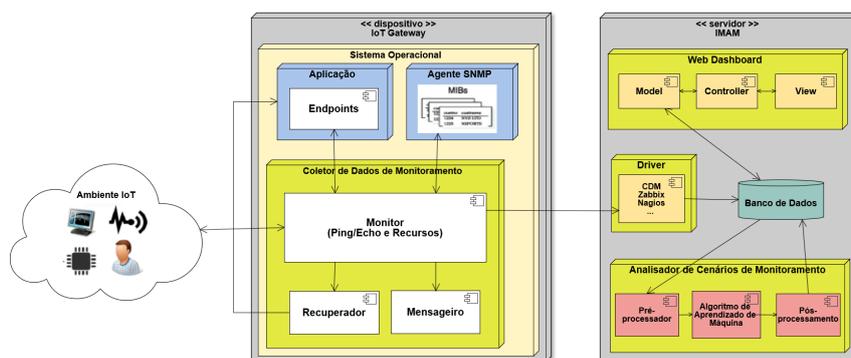


Figura 1. Arquitetura do IMAM.

Além dos componentes principais, o IMAM faz uso de um banco de dados para realizar a persistência dos dados coletados e processados.

### 4.1. Coletor de Dados de Monitoramento - CDM

A função do CDM é coletar dados, de forma periódica, de sistemas e equipamentos a serem monitorados através de *endpoints* e o protocolo SNMP. Após a coleta, os dados são persistidos no banco de dados. Esse componente também envia mensagens aos usuários e possui função de recuperação de serviços.

### 4.2. Driver

Tem como objetivo possibilitar a compatibilização do IMAM com as bases de dados de outras aplicações de monitoramento, tais como: Zabbix, Nagios, entre outras. Basicamente, esse componente age como um adaptador, que mapeia elementos contidos no banco de dados de terceiros para o padrão de persistência definido no banco de dados do IMAM.

### 4.3. Interface Web

O IMAM disponibiliza uma interface Web para permitir que o usuário realize configurações e visualize detalhes relacionados aos dados coletados, processados e analisados através dos demais componentes que compõem o sistema.

### 4.4. Analisador de Cenários de Monitoramento - ACM

O ACM é o principal componente do IMAM. Ele é responsável por realizar as análises dos *logs* de monitoramento através da utilização de técnicas de aprendizado de máquina. A partir de técnicas de agrupamento o ACM implementa um processo que gera vários grupos, baseado na proximidade das informações contidas a partir dos dados de *logs* recuperados da base de dados. Alertas poderão ser construídos e enviados aos usuários



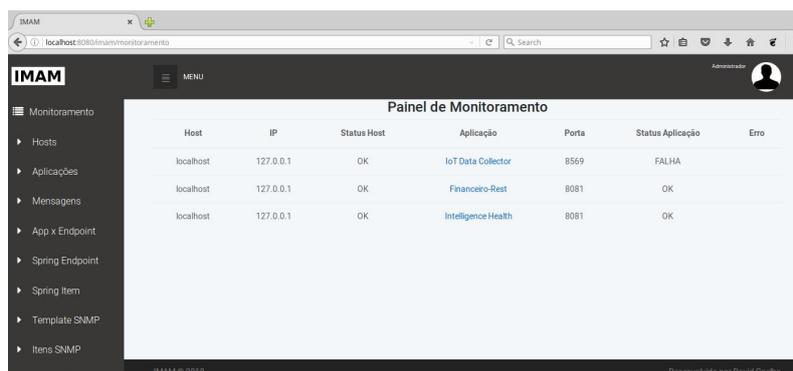


Figura 3. Protótipo do IMAM.

## 6. Experimento Preliminar

Como ainda não foi possível armazenar uma quantidade razoável de dados de monitoramento de sistemas e equipamentos através da utilização da ferramenta IMAM, optou-se por utilizar uma pequena base de dados de monitoramento de equipamentos de climatização do Datacenter do Instituto Metrópole Digital/UFRN, que foram coletados através do protocolo SNMP e ferramenta Zabbix.

A base de dados, que representa um subconjunto da base principal, é composta por 10 atributos e 8168 registros. Na fase de pré-processamento, procurou-se eliminar as inconsistências e redundâncias nos valores dos atributos, assim como, foi efetuada a normalização de todos os atributos numéricos, mostrados a partir da Tabela 1.

| Item                      | Unidade | Tipo     |
|---------------------------|---------|----------|
| Equipamento               | -       | Nominal  |
| Temperatura de sucção     | °C      | Numérico |
| Pressão de sucção         | PSI     | Numérico |
| Temperatura de Entrada    | °C      | Numérico |
| Temperatura de Retorno    | °C      | Numérico |
| Temperatura de Suprimento | °C      | Numérico |
| Velocidade do Ventilador  | %       | Numérico |
| Fluxo de Ar               | L/s     | Numérico |
| Demanda de Resfriamento   | W       | Numérico |
| Status                    | -       | Numérico |

Tabela 1. Atributos.

O objetivo desse primeiro experimento foi entender melhor o comportamento desses equipamentos (ar-condicionado) e poder realizar configurações otimizadas, no intuito de obter uma maior eficiência considerando a relação entre a demanda de refrigeração e distribuição da carga em cada célula de racks do Datacenter.

### 6.1. Resultados

Por não se saber a quantidade ideal de grupos existentes na base a priori, optou-se por usar uma técnica de agrupamento onde o parâmetro  $k$  (quantidade de grupos) não fosse determinado, deixando, dessa forma, que o próprio algoritmo encontre esse valor através da distribuição normal dos valores dos próprios atributos.

Devido a esse fato, foi utilizado o algoritmo EM que gerou uma partição com 8 (oito) grupos. A partir dessa descoberta, decidiu-se por avaliar as partições geradas

por outros dois algoritmos de agrupamento (k-Means e Hierárquico Aglomerativo), assim como o próprio EM, variando para tal a quantidade de grupos ( $k$ ) de 2 até 8. Os algoritmos utilizados estão disponíveis na plataforma de Aprendizado de Máquina Weka [Witten et al. 1999].

Por se tratarem de algoritmos probabilísticos (k-Means e EM), 5 (cinco) execuções diferentes (seeds) foram executadas para cada valor de  $k$ , perfazendo um total de 77 (setenta e sete) execuções (k-Means = 35, EM = 35 e Hierárquico = 7). Para cada partição gerada foram utilizadas duas medidas de avaliação (índices internos) que avaliam a compatibilidade das mesmas. Dessa forma, os resultados dos índices DB e Silhouette foram analisados, visando a escolha da partição mais adequada.

## 6.2. Validação dos Grupos

Os índices utilizados avaliam como os grupos estão separados, mas possuem características distintas. Porém, é importante esclarecer que o índice DB privilegia valores baixos, ou seja, quanto menor, melhor. Já o índice Silhouette pode gerar valores entre -1 e 1, sendo os valores mais próximos de 1 os valores para os grupos mais bem separados.

Para efeito de melhor entendimento, foram consideradas somente as três melhores partições de cada algoritmo. Dessa forma, como pode ser visualizado na Figura 4, o algoritmo Hierárquico (HR-3K) gerou a melhor partição (três grupos) para ambos os índices (0,79 - DB e 0,61 - Silhouette).

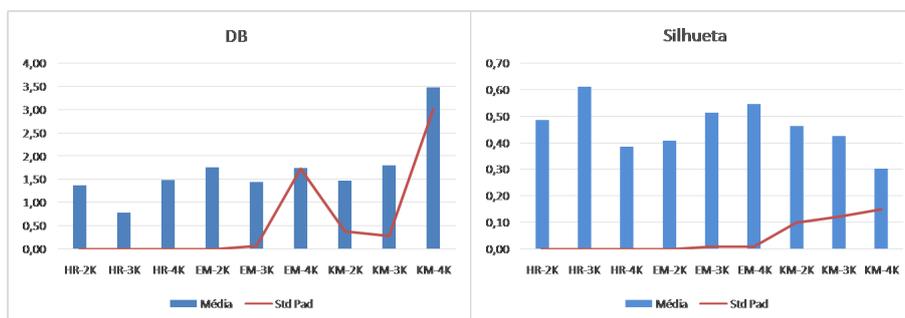
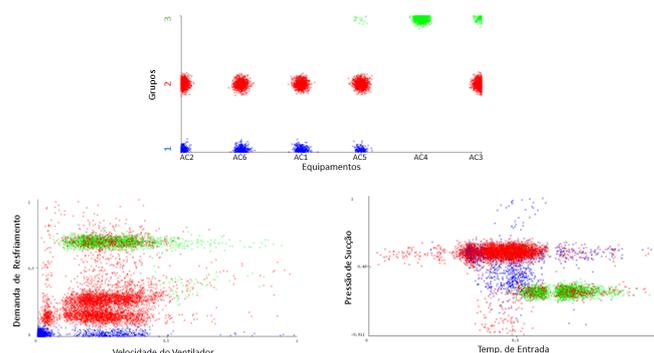


Figura 4. Índices de validação DB e Silhouette.

O gráfico superior central da Figura 5 exibe a relação entre os 3 grupos gerados (HR-3K) e os equipamentos de ar-condicionado. Já o gráfico inferior esquerdo relaciona a demanda de resfriamento com a velocidade dos ventiladores, possibilitando perceber que a demanda de resfriamento é sempre alta para o grupo 3, representado pela cor verde. Por último, no gráfico inferior direito é exibida a relação entre pressão de sucção e temperatura de entrada. Nele, é possível observar que, para o grupo 3, a temperatura de entrada é sempre acima de 50% e a pressão de sucção está constantemente abaixo de 50% quando comparado aos outros grupos.

Em linhas gerais, conclui-se que os equipamentos, representados pelos grupos 1 (azul) e 2 (vermelho), estão trabalhando de forma correta, estável e sem sobrecarga. Os grupos 1 (azul) e 2 (vermelho) representam demandas de refrigeração baixa e média, respectivamente. Por outro lado, o grupo 3 (verde) representa alta demanda por refrigeração, permitindo, dessa forma, que pode haver a ocorrência de problemas em equipamentos e que podem comprometer o bom funcionamento do ambiente do datacenter.



**Figura 5. Grupos formados.**

A etapa de experimento seguinte seria testar várias técnicas de classificação (Seção 2.2) para a escolha do modelo de classificação mais adequado, que irá compor o componente ACM. Porém, como ainda não temos dados suficientemente gerados pela ferramenta IMAM, decidiu-se executar tal fase um pouco mais tarde.

## 7. Conclusão e Trabalhos Futuros

A disponibilidade dos serviços de tecnologia é primordial. Dessa forma, sem um monitoramento de infraestrutura em TI automatizado, não há como verificar se algum serviço ou aplicação está indisponível. Logo, automatizar o monitoramento é tão importante quanto otimizar outros processos de gestão.

Visando oferecer suporte a tais exigências, este artigo apresentou a proposta de uma ferramenta de monitoramento que possui a capacidade de monitorar sistemas e equipamentos em uma infraestrutura de IoT, e principalmente, de prever falhas e tomar ações de suporte proativas. A ferramenta IMAM foi modelada para dar suporte a ações que envolvem tolerância a falhas em sistemas, detectando e corrigindo problemas, reduzindo ou eliminando o tempo de inatividade.

Para tal, utilizaram-se técnicas descritivas e preditivas de aprendizado de máquina para a realização de análises envolvendo sistemas e dispositivos em ambientes baseados em IoT. Dessa forma, a hipótese considera que, a partir da aplicação dessas técnicas, o IMAM possa inferir a respeito de falhas, auxiliando o suporte e a manutenção da disponibilidade e da confiabilidade sobre um conjunto de sistemas e equipamentos.

Um dos desafios na área de tolerância a falhas é o fato de aferir se as técnicas aplicadas irão resultar em um real aumento da confiabilidade de sistemas. Como em grande parte dos sistemas, as falhas podem acontecer de forma aleatória e incontrolável. Uma solução para esse problema é a injeção de falhas.

Assim, com o objetivo de avaliar as funcionalidades do IMAM, é proposto como trabalho futuro a realização de um estudo de caso considerando a utilização de técnicas de injeção de falhas e a atuação da ferramenta IMAM monitorando um conjunto de sistemas e equipamentos de uma plataforma de assistência em saúde baseada em IoT. Dessa forma, a hipótese considera que o IMAM possa detectar falhas, auxiliando o suporte e a manutenção da disponibilidade e da confiabilidade sobre um conjunto de sistemas e equipamentos de uma plataforma de saúde baseada em IoT.

## Referências

- Anderson, T. and Lee, P. (1981). *Fault tolerance: Principles and practice*. Prentice-Hall International, Inc. Englewood Cliffs, New Jersey, USA.
- Avizienis, A. (1998). *Infrastructure-based design of fault-tolerant systems*, volume 98.
- Bass, L., Clements, P., and Kazman, R. (2003). *Software Architecture in Practice*. JPearson Education, Inc., New York, EUA, 3rd edition.
- Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W., and Lu, C. (2015). A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, 3:10–23.
- Faceli, K., Carvalho, A., Lorena, A. C., and Gama, J. (2011). *Inteligência Artificial: Uma abordagem de aprendizado de máquina*. Grupo Gen-LTC.
- Ferrera, E., Conzon, D., Brizzi, P., Rossini, R., Pastrone, C., Jentsch, M., Kool, P., Kamienski, C., and Sadok, D. (2017). Xmpp-based infrastructure for iot network management and rapid services and applications development. *Annals of Telecommunications*, 72(7):443–457.
- Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2002). Clustering validity checking methods: part ii. *SIGMOD Rec.*, 31 (3):19–27.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.
- Silva, P. T. and Torres, C. B. (2010). *Gestão e Liderança Para Profissionais de Ti*. 1yh edition.
- Soysal, M. and Schmidt, E. G. (2010). Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Perform. Eval.*, 67(6):451–467.
- Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. *SIGMETRICS Perform. Eval. Rev.*, 41(4):70–73.
- Tokuda, Z., Takuro, Yonezawa, and Nakazawa, J. (2014). Amonitoring dependability of city-scale iot using d-case. *IEEE World Forum on Internet of Things (WF-IoT)*.
- Webb, P., Syer, D., Long, J., Nicoll, S., Winch, R., Wilkinson, A., Overdijk, M., Dupuis, C., and Deleuze, S. (2013). Spring boot reference guide. *Part IV. Spring Boot features*, 24.
- Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., and Cunningham, S. J. (1999). *Weka: Practical machine learning tools and techniques with java implementations*.
- Xu, R. and Wunsch, II, D. (2005). Survey of clustering algorithms. *Trans. Neur. Netw.*, 16(3):645–678.
- Yaseen, Q., Jararweh, Y., Al-Ayyoub, M., and AlDwairi, M. (2017). Collusion attacks in internet of things: Detection and mitigation using a fog based model. *Sensors Applications Symposium (SAS)*.

# Integrando as Plataformas FIWARE e KNoT para o Desenvolvimento de Aplicações de Internet das Coisas

César Perdigão Batista, Pedro Victor Silva, Thais Batista, Everton Cavalcante

Departamento de Informática e Matemática Aplicada (DIMAp)  
Universidade Federal do Rio Grande do Norte (UFRN)  
Natal-RN, Brasil

{cesar.perdigao, pedrovictor.caldas}@gmail.com,  
{thais, everton}@dimap.ufrn.br

**Abstract.** *Middleware platforms have been used as underlying infrastructure to ease development of Internet of Things (IoT) applications. FIWARE is a platform that has stood out in this scenario as an open, extensible solution for application development, but it supports only devices with native Internet connection. On the other hand, the KNoT platform allows virtualizing physical devices that do not have IP communication with minimal development effort. This paper presents the integration of the two platforms through the development of a component compatible with the FIWARE standard interface, inserted into KNoT aiming at enhancing their use for IoT applications.*

**Resumo.** *Plataformas de middleware têm sido utilizadas como infraestrutura subjacente para facilitar o desenvolvimento de aplicações de Internet das Coisas (IoT). A FIWARE é uma plataforma que tem se destacado nesse cenário como uma solução aberta e extensível para desenvolvimento de aplicações, porém ela admite apenas dispositivos com conexão nativa à Internet. Por outro lado, a plataforma KNoT possibilita a virtualização de dispositivos físicos que não possuem comunicação via IP com mínimo esforço de desenvolvimento. Este artigo apresenta a integração dessas duas plataformas por meio do desenvolvimento de um componente compatível com a interface padrão da FIWARE, inserido na KNoT para potencializar o uso de ambas as plataformas para aplicações de IoT.*

## 1. Introdução

A multiplicação de dispositivos computacionais de diversas proporções possibilita a criação de um espaço virtual massivo que, quando integrado de maneira eficiente a objetos do mundo físico, permite a exploração de uma variedade de aplicações envolvendo usuários e dispositivos. Dessa forma, é interessante que tanto *hardware* quanto *software* aglutinem-se para que dispositivos conectados ofereçam serviços e atuem de forma incisiva nas duas realidades. O paradigma de *Internet das Coisas* (do Inglês *Internet of Things* - IoT) [Atzori et al. 2010, Miorandi et al. 2012] abrange desde como a infraestrutura de *hardware* se organiza até a abstração em diferentes camadas de *software* para conectar objetos do mundo real (coisas) às redes de computadores.

Aplicações de IoT demandam plataformas que visem facilitar o seu processo de desenvolvimento, o qual pode envolver a integração de uma diversidade de dispositivos heterogêneos, com capacidades variadas, meios de transmissão de dados e protocolos de comunicação distintos. A literatura apresenta várias plataformas de *middleware* que servem como infraestrutura subjacente ao desenvolvimento de aplicações de IoT

[Mineraud et al. 2016, Ngu et al. 2017]. Nesse contexto, a Comissão Europeia disponibilizou a plataforma FIWARE<sup>1</sup>, um *middleware open-source* desenvolvido para contribuir com a criação de tecnologias voltadas para a Internet do Futuro e que é capaz de atender a uma ampla gama de requisitos considerados relevantes para plataformas de IoT [Batista et al. 2016]. A FIWARE provê diversos componentes extensíveis e reutilizáveis que podem facilitar o desenvolvimento de aplicações, os chamados *habilitadores genéricos* (do Inglês *generic enablers* - GEs). Tais componentes seguem o modelo de especificação NGSI (*Next Generation Service Interfaces*)<sup>2</sup> para padronizar a troca de informações e permitir a interoperabilidade entre eles.

Também visando oferecer facilidades para interoperabilidade e gerenciamento de dispositivos para o desenvolvimento de aplicações de IoT, o Centro de Estudos e Sistemas Avançados do Recife (CESAR) concebeu a plataforma KNoT<sup>3</sup>. Baseada em *hardware* de baixo custo e *software* com implantação simplificada, a KNoT propõe-se a permitir interoperabilidade entre plataformas e dados de diversos dispositivos, oferecendo protocolos de diferentes camadas do modelo OSI com o intuito de possibilitar adaptabilidade às necessidades de cada aplicação em particular. Essa plataforma promove ainda a virtualização de dispositivos físicos para que estes possam ser acessados por outras plataformas, usuários e aplicações.

Ainda que seja possível conectar dispositivos de IoT à FIWARE, existem dificuldades para lidar com a inerente heterogeneidade a nível de *hardware*, uma vez que a própria plataforma é focada em prover módulos de *software* para facilitar o desenvolvimento de aplicações. Os GEs de IoT da FIWARE não oferecem suporte satisfatório para dispositivos que não possuem em seu *hardware* capacidade de conexão direta à Internet. Portanto, no ecossistema da FIWARE, a plataforma KNoT poderia ser inserida com o papel de prover novos meios de comunicação entre os dispositivos, a fim de facilitar a integração de variados objetos físicos em um ambiente virtual e interoperável. A KNoT destaca-se pela facilidade de implantação de novos dispositivos ao ambiente, exigindo mínima intervenção por parte do desenvolvedor, o qual basicamente terá de implementar as funções de leitura e escrita dos dispositivos a nível de microcontrolador e registrá-los no seu *gateway*. Para isso, microcontroladores comuns do mercado, compatíveis com os mais diversos dispositivos, comunicam-se através de radiofrequência com o *gateway* da KNoT que, por sua vez, fornece os dados providos por tais dispositivos para plataformas, aplicações e/ou usuários que deles irão fazer uso.

No intuito de suplantar as limitações mencionadas, este trabalho apresenta a integração da plataforma KNoT com o ecossistema FIWARE para que dispositivos a serem utilizados por uma aplicação construída sobre a FIWARE possam fazer uso das facilidades de implantação e acesso a dispositivos físicos providas pela KNoT. Do lado da KNoT, essa integração faz com que seja possível a utilizar os diversos mecanismos providos pela FIWARE para o desenvolvimento de aplicações de IoT. Além de apresentar a integração realizada no contexto deste trabalho, o artigo apresenta uma validação através do desenvolvimento de uma aplicação que explora o cenário integrado entre as duas plataformas.

---

<sup>1</sup> <https://www.fiware.org/>

<sup>2</sup> <https://goo.gl/RqXk12>

<sup>3</sup> <https://www.knot.cesar.org.br/>

O restante deste artigo está organizado como segue. A Seção 2 apresenta brevemente as plataformas FIWARE e KNoT. A Seção 3 apresenta a solução desenvolvida neste trabalho para integração das duas plataformas. A Seção 4 ilustra o uso da integração FIWARE-KNoT no desenvolvimento de uma aplicação. Alguns trabalhos relacionados são discutidos na Seção 5. A Seção 6 traz algumas considerações finais e direções para trabalhos futuros.

## 2. As Plataformas FIWARE e KNoT

Apesar de tanto a FIWARE quanto a KNoT oferecem uma infraestrutura facilitadora para aplicações de IoT, o foco das duas plataformas não é o mesmo. Conectar objetos inteligentes a uma camada superior agregadora a partir da qual eles serão de fato acessados é apenas um dos módulos da FIWARE e isso já é um desafio por si só justamente pelo alto grau de heterogeneidade de tais objetos. A KNoT busca não apenas descomplicar essa barreira, mas também adicionar maior flexibilidade à configuração e instalação de dispositivos físicos pelo fato de dar suporte a outros meios de transmissão de dados entre eles (*e.g.*, rádios subgigahertz, rádios Mesh, rádios de área ampla e baixa potência, *Bluetooth*, etc.). Nesta seção será apresentada uma visão geral dos elementos das plataformas FIWARE e KNoT e como eles podem ser utilizados para a integração de dispositivos físicos e aplicações de IoT.

### 2.1. FIWARE

A FIWARE é uma plataforma aberta, genérica e extensível voltada principalmente para dar suporte ao desenvolvimento de aplicações em cidades inteligentes. A FIWARE provê um conjunto de especificações disponíveis através de APIs abertas e está estruturada sobre componentes de *software* de alto nível, os GEs. Os GEs seguem o modelo de especificação NGSI para padronizar a troca de informações e permitir a interoperabilidade entre os componentes. No NGSI, as informações são estruturadas de forma genérica por meio de *entidades de contexto* que podem ser utilizadas para representar, de forma virtual, os elementos do mundo real, tais como um lugar, uma pessoa, um objeto, sensores, atuadores, etc. Um elemento de contexto possui um identificador, um tipo e uma lista de atributos, os quais possuem um nome, um tipo, um valor e uma lista de metadados, cada um deles com nome, tipo e valor. De forma prática, o contexto dos dispositivos é disponibilizado no *Orion Context Broker* (ou simplesmente *Orion*), o GE responsável por gerenciar o contexto das entidades. Uma vez que esses dados estejam no *Orion*, uma aplicação pode realizar a consulta de seu estado ou outros GEs podem ser por ela utilizados via NGSI.

A FIWARE provê alguns GEs que podem ser utilizados no desenvolvimento de aplicações de IoT. Dentre eles, o *IoT Discovery* e o *IoT Broker* oferecem um conjunto de funcionalidades e abstrações que facilitam o compartilhamento e o consumo de dados no âmbito da plataforma e não agem como fornecedores de dados dos dispositivos físicos ao *Orion*. Por sua vez, o *Backend Device Management*, apesar de ser um agente IoT que permite a conversão dos protocolos dos dispositivos e sensores (tais como MQTT, CoAP ou HTTP) para o padrão NGSI, limita-se a receber essas mensagens a nível de *software* através de rede Ethernet, convertê-las e encaminhá-las a um *broker*. Essas capacidades oferecidas pelos GEs de IoT do FIWARE são convenientes quando os equipamentos já estão instalados, porém não solucionam de maneira independente os problemas relacionados com a instalação física, meios de

transmissão entre o dispositivo e o *gateway*, ou facilidades em prover conectividade para torná-los virtualizáveis.

## 2.2. KNoT

A Figura 1 ilustra a arquitetura da KNoT, mostrando como dispositivos físicos (referidos como *devices* ou *KNoT Things*), que não se conectam diretamente à Internet, são conectados através de um *Gateway*. Nessa arquitetura, os *devices* são compostos por um microcontrolador e um módulo de radiofrequência. O desenvolvedor pode conectar seus sensores e atuadores a este dispositivo principal para criar um *device*.

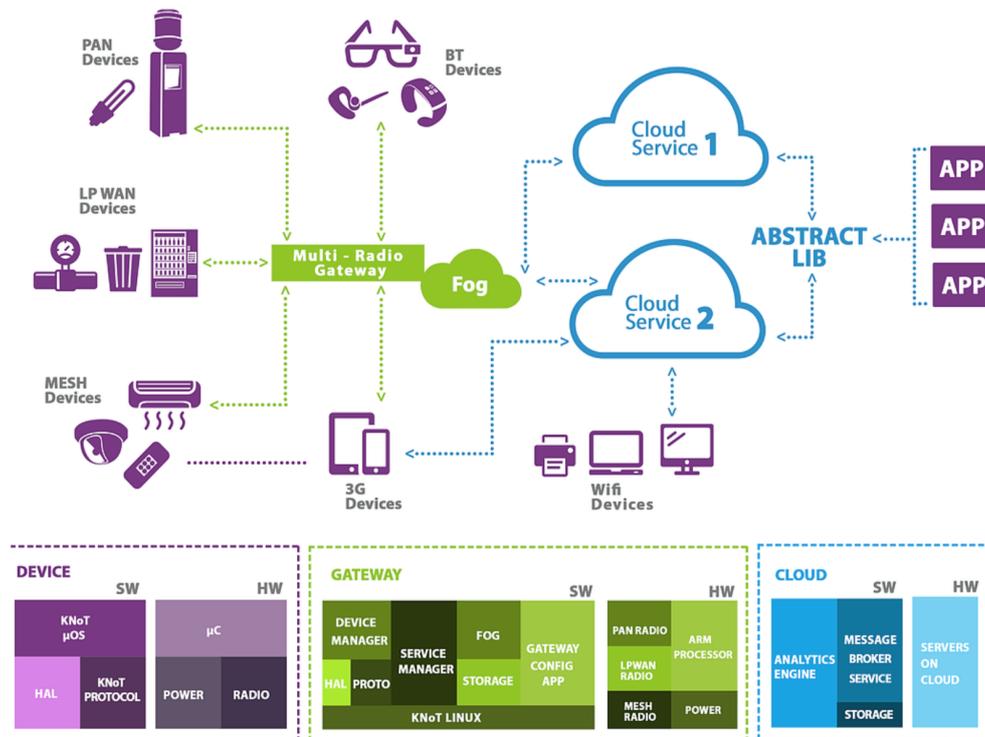


Figura 1. Arquitetura da plataforma KNoT versão 1.0 (<https://www.knot.cesar.org.br/>)

De maneira resumida, os macrocomponentes da plataforma são três, a saber, *Things* (*devices*), *Gateway* e *Cloud*. As *Things*, mais comumente constituídas por um dispositivo Arduino com sensores ou atuadores, conectam-se ao *Gateway* por meio de um transmissor de rádio para troca de dados. No caso do *Gateway*, elemento central que normalmente consiste em um dispositivo Raspberry Pi dotado de conexão Ethernet e transmissores de rádio nRF24L01 e/ou LoRa que traduz o protocolo das *Things* para JSON. O *Gateway*, que consiste em uma distribuição Linux embutida e personalizada chamada *KNoT Linux*, também oferece um módulo de *software* chamado *WebUI*, utilizado para gerenciar *Things*. Por fim, a *Cloud* é responsável por coletar os dados do *Gateway* e atua como um ponto de acesso remoto para as aplicações, além de ser responsável pelo roteamento de mensagens para aplicativos e dispositivos.

A prática de se usar um *gateway* como nó de comunicação entre os dispositivos e plataformas de nuvem é comum em IoT, pois são os *gateways* que normalmente orquestram os dispositivos agregados [Cavalcante et al. 2016]. Entretanto, o diferencial do KNoT é que os dispositivos instalados abaixo da camada do *Gateway* são abstraídos, não importando qual tipo de conexão física o sensor possui. Contanto que o dispositivo

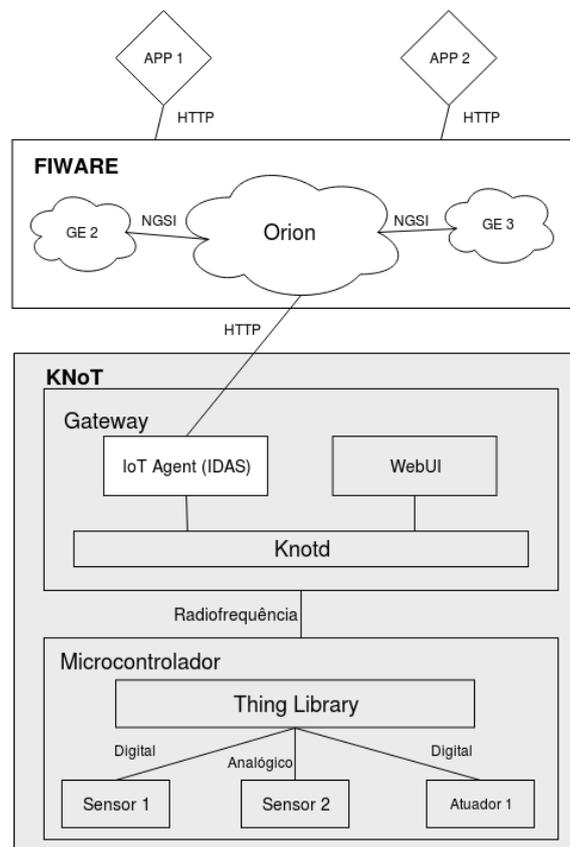
seja instalável em um microcontrolador compatível, os dados podem ser enviados para o *Gateway* e, portanto, serem virtualizados para camadas superiores. Isso é viabilizado pela comunicação entre a *Thing* e o *Gateway* ao permitir que novos meios de transmissão sem fio sejam empregados para fazer esse intermédio.

A flexibilidade referente a como os dados dos dispositivos podem ser transmitidos para o *Gateway* é interessante, já que estabelece uma perspectiva em que equipamentos que normalmente não tem conectividade à Internet possam ser integrados a ambientes de nuvem com facilidade e, com isso, diversas aplicações podem consultar seus dados e atuar sobre eles. Considerando que a ponte *Thing–Gateway–Cloud* é feita de maneira transparente, o desenvolvedor que utilizará a KNoT para construir uma aplicação apenas precisará se preocupar com a exposição dos dados dos dispositivos físicos, definindo o comportamento deles a nível de microcontrolador. Para isso, são oferecidas bibliotecas disponíveis para os mais variados tipos de *hardware* que determinam a operação e estabelecem a comunicação entre o microcontrolador e o *Gateway*.

### 3. Integração entre FIWARE e KNoT

A plataforma KNoT propõe uma solução prática para a integração de dispositivos físicos heterogêneos sem comunicação via IP nativa. Dessa forma, a ideia de complementar esse atributo tornando-o mais flexível e exposto a um ambiente rico como o da FIWARE é proveitosa, visto que a KNoT passará a ter acesso ao ecossistema FIWARE, que oferece variadas ferramentas para o desenvolvimento de aplicações com diferentes funcionalidades. No entanto, para realizar essa integração, é necessário que as camadas inferiores da KNoT, responsáveis pela comunicação com os equipamentos físicos, sejam adaptadas para interagir com o *Orion*, componente central da FIWARE. A diferença de estrutura entre as duas plataformas exige a modificação de alguns componentes de menor granularidade da KNoT e a adição de um componente da FIWARE à KNoT para fazer com que as informações providas sejam compatíveis com o formato NGSI e endereçáveis ao *Orion*.

A Figura 2 ilustra a arquitetura da integração da KNoT com a FIWARE e seus componentes. Dentre esses componentes da arquitetura, a *KNoT Thing Library*, implementada na KNoT, possibilita que microcontroladores diversos estabeleçam comunicação com o *Gateway*. Os desenvolvedores precisam apenas usá-la configurando alguns dados e funções para que a comunicação seja feita. Feito isso, é necessário acessar o componente *WebUI* do *Gateway*, o qual atua no gerenciamento dos dispositivos vinculados ao *Gateway*, e se comunicar com o componente *knotd* para compartilhar informações sobre o registro de dispositivos e permitir a interação com eles.



**Figura 2. Arquitetura da integração entre KNoT e FIWARE**

O *knotd* é o *software* executado no *Gateway*, que foi modificado para dar suporte à integração com a FIWARE e receber os dados dos equipamentos via rádio ou outras tecnologias de comunicação sem fio. Tais dados são traduzidos para o formato JSON antes de serem enviados ao *IoT Agent*, componente do *Intelligence Data Advanced Solution* (IDAS), uma implementação do *GE Backend Device Management* da FIWARE.

O *IoT Agent* foi desenvolvido e instanciado no *Gateway* para receber os dados do *knotd* e desempenhar toda a tarefa de cadastro, atualização, exclusão e leitura, exigidas para a produção de um contexto compatível com NGSI. Seu papel torna-se relevante pelo fato de ele não se limitar a criar um canal de comunicação com o *Orion* ou converter o protocolo adotado na KNoT para NGSI, mas também por oferecer serviços comuns ao ambiente da FIWARE, tais como lidar com aspectos de segurança da plataforma e enviar comandos. Além disso, ele assume os papéis de criar subscrições nas entidades do *Orion* para que o *knotd* seja avisado sempre que haja uma atualização e encaminha as ações ocorridas aos devidos dispositivos. Isso é possível devido à modificação do *knotd*, realizada neste trabalho, para que as mensagens provenientes dos dispositivos sejam traduzidas para uma estrutura compatível com a do *Orion*. Além disso, foi necessário elaborar um controle de mensagens entre o *Orion* e o *knotd* para tomar ações de acordo com as subscrições nas entidades do *Orion*. A Figura 3 ilustra a troca de mensagens do KNoT integrado com o *Orion*, exemplificando o cadastro de um dispositivo.

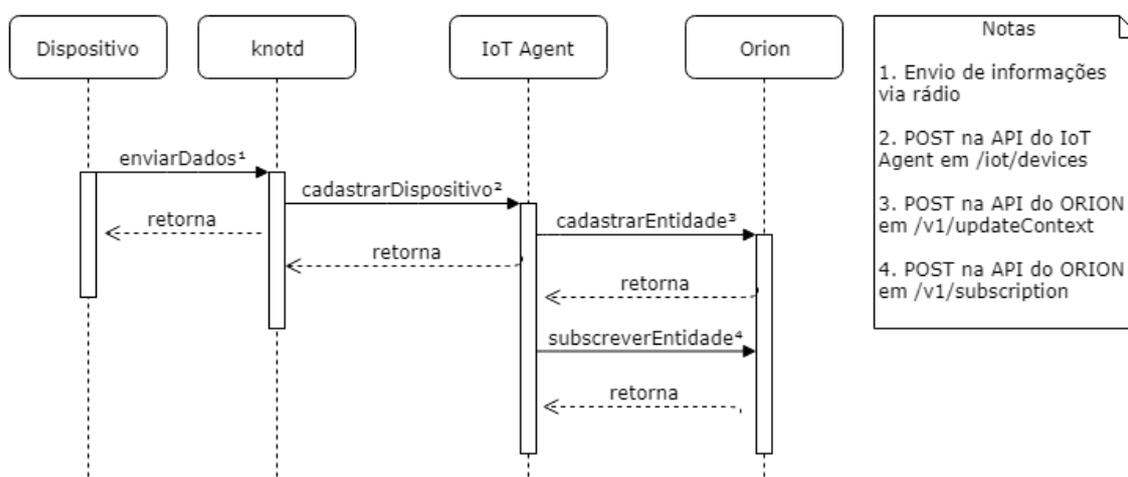
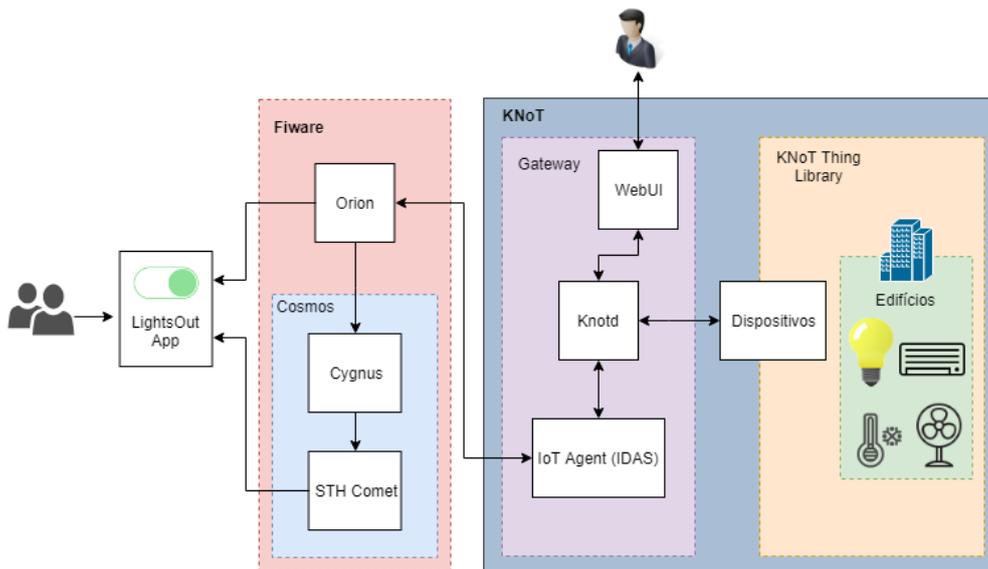


Figura 3. Diagrama de Sequência ilustrando a operação de cadastro de um dispositivo

Com a integração realizada, dispositivos que normalmente não poderiam ser inseridos no ecossistema FIWARE podem ser agora modelados como entidades no *Orion* e, conseqüentemente, ser acessados e utilizados por outros GEs e/ou por aplicações diversas construídas sobre a plataforma. O procedimento genérico que deve ser realizado por um desenvolvedor para que uma aplicação possa utilizar um dispositivo provido ao *Orion* pela KNoT consiste em: (i) conectar os sensores/atuadores do dispositivo e o transmissor de radiofrequência fisicamente ao microcontrolador; (ii) importar a biblioteca *KNoT Thing Library* no microcontrolador para definir o funcionamento do dispositivo e permitir que as entidades de contexto desejadas sejam criadas no *Orion* e o gerenciamento delas pela KNoT funcione; (iii) acessar o *WebUI* do *Gateway* para configurar o endereço do *Orion* a ser utilizado para registro de entidades; (iv) realizar a aceitação dos dispositivos disponíveis no *WebUI*, admitindo-os automaticamente no *Orion*, e; (v) usar a entidade cadastrada na aplicação que realiza operações HTTP no *Orion*.

#### 4. Validação

De forma a validar a solução apresentada neste trabalho, foi desenvolvida a aplicação *LightsOut*, que controla a iluminação de uma sala (ver Figura 4). Nessa aplicação, uma lâmpada originalmente sem conexão à Internet está modelada como uma entidade de contexto no *Orion* da FIWARE e, como tal, contém informações sobre seu estado (ligada e desligada) e disponibilidade. Isso foi possível graças à instalação da lâmpada em um Arduino Uno dotado de um dispositivo transmissor de radiofrequência nRF24L01, o qual envia informações para o *gateway* que implementa, de maneira transparente, a integração proposta com as modificações no *knotd* e o desenvolvimento do *IoT Agent*, permitindo assim disponibilizar as informações acerca do estado da lâmpada no *Orion*. Por fim, a entidade da lâmpada modelada no *Orion* contém atributos que, ao serem modificados, geram uma notificação que avisa à KNoT sobre a mudança, acompanhada desses dados. Após a KNoT receber os dados, ela imediatamente encaminha as informações para os dispositivos através de radiofrequência ou qualquer outro meio sendo utilizado para a comunicação.



**Figura 4. Arquitetura da aplicação que utiliza a integração entre KNoT e FIWARE**

Partindo de uma visão global do ambiente FIWARE, foi incorporada à aplicação o uso de uma ferramenta do GE *Cosmos*, chamada *STH Comet*, cuja função é fornecer dados históricos e séries temporais agregadas por meio de uma API. O *STH Comet* trabalha em conjunto com o *Cygnus*, também componente do *Cosmos*, encarregado de persistir dados de contexto do *Orion* e criar uma visão histórica desses dados que podem posteriormente ser lidos. Para tirar proveito dessa funcionalidade, a *LightsOut* consulta a API do *STH Comet* e exibe gráficos acerca do histórico da iluminação da sala.

Apesar de ser uma aplicação com funcionalidade simples quando observada de maneira isolada, a *LightsOut* exemplifica o uso da solução que integra as duas plataformas abordadas. O uso adicional dos GEs do FIWARE - neste caso o *Cosmos* - ilustra o potencial de incremento que esse ecossistema oferece para os desenvolvedores criarem novas aplicações de IoT.

## 5. Trabalhos Relacionados

O desenvolvimento de plataformas de *middleware* para IoT tem atraído atenção da indústria e da academia pela capacidade que esse tipo de solução tem para endereçar problemas de heterogeneidade a nível de implantação física e dos tipos de dados utilizados para comunicação entre componentes. Esses desafios vêm das especificidades dos dispositivos e se tornam barreiras para ampla adoção do paradigma. Portanto, para promover a interoperabilidade e abstração dessas particularidades, é necessário que as plataformas atendam a diversos requisitos considerados importantes nesse cenário.

Pires *et al.* (2015) analisaram onze plataformas para IoT e reportaram como cada uma atende aos seguintes requisitos: (i) interoperabilidade entre dispositivos, serviços e aplicações, (ii) gerenciamento e descoberta dinâmica de dispositivos, (iii) coleta, gerenciamento e processamento de informações de contexto, (iv) escalabilidade para gerenciar a crescente quantidade de dispositivos, (v) gerenciamento de grandes volumes de dados gerados pelos dispositivos físicos e transmitidos através da rede, (vi) segurança, integridade e privacidade de dados e (vii) adaptação dinâmica para garantir a disponibilidade e qualidade de aplicações em tempo de execução. Os autores

concluíram que nenhuma delas oferece uma solução completa de maneira que atenda a todos os requisitos. Em um levantamento posterior, feito por Batista *et al.* (2016), outras quatro plataformas foram analisadas em complemento a esse estudo, a saber, Kaa<sup>4</sup>, SOFIA<sup>5</sup>, FIWARE e CityHub [Lea et al. 2014]. Como resultado, constatou-se que apenas a plataforma FIWARE foi capaz de satisfazer os requisitos.

Dado que o foco do presente trabalho é analisar a facilidade da implantação física de dispositivos nas plataformas em conjunto com a interoperabilidade dos seus dados, uma solução satisfatória não foi encontrada nas outras iniciativas. Nos dois estudos mencionados, observou-se que as plataformas atendem apenas a subconjuntos dos requisitos e os abordam de formas diversas. Sobre a questão da heterogeneidade de dispositivos, a maioria das plataformas oferece suporte a protocolos comumente utilizados em IoT, como CoAP e MQTT, exigindo que os dispositivos possuam conexão direta à Internet ou o uso de um *gateway* para os que não têm essa capacidade. Em casos como esse, apesar da maior flexibilidade, é necessário adaptar os equipamentos para se comunicarem com o *gateway* e isso não é uma tarefa trivial. Plataformas como EcoDiF [Pires et al. 2014], *WoT Enabler* [Gao et al. 2011] e a própria FIWARE se valem desse mecanismo, porém há um esforço de integrar cada dispositivo com suas particularidades. A KNoT, por sua vez, poupa os desenvolvedores de aplicações desse trabalho adicional de elaborar a comunicação entre os dispositivos e o *gateway*, de modo que, ao usarem a infraestrutura fornecida pela KNoT, eles não terão que produzir *drivers* individuais para cada tipo de dispositivo a ser implantado. Como foi mostrado com a solução proposta neste trabalho, a integração entre as plataformas KNoT e FIWARE destaca-se pela simplicidade que a implantação de novos dispositivos pode ocorrer, favorecendo o acesso deles ao ambiente diverso que os GEs oferecem, os quais desempenham papel significativo no cumprimento dos requisitos citados acima.

## 6. Conclusão

Este artigo apresentou a integração das plataformas FIWARE e KNoT para permitir o desenvolvimento de aplicações de IoT. Esta proposta passa a oferecer a possibilidade de cooperação entre as duas plataformas, visto que agora é possível conectar dispositivos normalmente sem conexão à Internet (tais como os dotados de transmissão por radiofrequência) para serem virtualizados na FIWARE. O benefício desta contribuição se dá pelos novos tipos de dispositivos que os desenvolvedores de aplicações sobre a FIWARE podem utilizar em suas soluções, com mínimo esforço para sua implantação, sem falar que as informações providas por esses dispositivos passam a estar disponíveis para aplicações diversas.

Para fins de validação inicial da integração, foi desenvolvida uma aplicação para controlar a iluminação de uma sala e exibir estatísticas históricas sobre o uso de lâmpadas. Futuramente, serão desenvolvidas aplicações mais complexas para emprego da proposta de integração em maior escala. Ademais, a questão do desenvolvimento em conjunto com GEs de segurança da FIWARE será explorada, considerando que é um aspecto fundamental na disponibilização de aplicações para um grande número de usuários.

---

<sup>4</sup> <https://www.kaaproject.org/>

<sup>5</sup> [http://sofia2.com/home\\_en.html](http://sofia2.com/home_en.html)

## Referências

- Atzori, L., Iera, A., Morabito, G. (2010) “The Internet of Things: A survey”, *Computer networks* 54(15), pp. 2787-2805.
- Batista, T. et al. (2016) Requisitos e plataformas de middleware para cidades inteligentes. Relatório técnico - Instituto Metr pole Digital, Universidade Federal do Rio Grande do Norte, Natal, Brasil. Dispon vel em <https://goo.gl/UMf1s4>.
- Cavalcante, E., Pereira, J., Alves, M. P., Maia, P., Moura, R., Batista, T., Delicato, F. C., Pires, P. F. (2016) “On the interplay of Internet of Things and Cloud Computing: A systematic mapping study”, *Computer Communications* 89-90, pp. 17-33.
- Gao, L., Zhang, C., Sun, L. (2011) “RESTful Web of Things API in sharing sensor data”, *Proceedings of the 2011 International Conference on Internet Technology and Applications*. USA: IEEE, pp. 1-4.
- Lea, R., Blackstock, M., (2014) “City hub: A cloud-based IoT platform for smart cities”. *Proceedings of the IEEE 6th International Conference on Cloud Computing Technology and Science*. USA: IEEE, pp. 799-804.
- Mineraud, J., Mazhelis, O., Su, X., Tarkoma, S. (2016) “A gap analysis of Internet-of-Things platforms”, *Computer Communications* 89-90, pp. 5-16.
- Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I. (2012) “Internet of Things: Vision, applications and research challenges”, *Ad Hoc Networks* 10(7), pp. 1497-1516.
- Ngu, A. H., Gutierrez, M., Metsis, V., Nepal, S., Sheng, Q. Z. (2017) “IoT middleware: A survey on issues and enabling technologies”, *IEEE Internet of Things Journal* 4(1), pp. 1-20.
- Pires, P. F., Cavalcante, E., Barros, T., Delicato, F. C., Batista, T., Costa, B. (2014) “A platform for integrating physical devices in the Internet of Things”, *Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing*. USA: IEEE, pp. 234-241.
- Pires, P. F., Delicato, F., Batista, T., Barros, T., Cavalcante, E., Pitanga, M. (2015) Plataformas para a Internet das Coisas. In: Martinello, M., Robeiro, M. R. N., Rocha, A. A. A. (org.) XXXIII Simp sio Brasileiro de Redes de Computadores e Sistemas Distribu dos – Minicursos. Porto Alegre, Brasil: SBC, pp. 110-169.

# Modelo Baseado em Rede de Sensores Sem Fio Para Predição de Disfunções em Cultivos Agrícolas

Uélison Jean L Santos<sup>1</sup>, Rodrigo da Rosa Righi<sup>1</sup>, Cristiano André da Costa<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Computação Aplicada - SoftwareLab - Unisinos

uelison.jean@gmail.com, rrrighi@unisinos.br, cac@unisinos.br

**Abstract.** *One of the challenges for the future is to ensure food for all. However, the increase in food production is of concern, since most agricultural systems use environmental impacts resources. In this context, researches with IoT have shown great potential. However, there are still few viable options for application in the rural environment and that support the producer in the decision making. This article presents AgriPrediction, a model proposal that can be used by farmers using LoRa network architecture and predict dysfunctions using the ARIMA model. The results obtained during tests are positive since its application in arugula cultivation a 17.94 % gain was obtained concerning leaf development and 14.29 % in weight.*

**Resumo.** *Um dos desafios para o futuro é garantir alimento à todos. Porém, o aumento da produção de alimentos preocupa, uma vez que sistemas agrícolas utilizam recursos que causam impactos ambientais. Neste contexto, pesquisas com Internet das Coisas mostram alternativas, porém ainda existem poucas opções viáveis de aplicação no ambiente rural que auxiliem o produtor agrícola na tomada de decisões. Este artigo apresenta o AgriPrediction, uma proposta de modelo que pode ser aplicado por agricultores utilizando arquitetura de rede LoRa e predição de disfunções utilizando o modelo ARIMA. Os resultados obtidos são positivos, pois em sua aplicação em cultivo de rúcula obteve-se ganho de 17.94% em relação ao desenvolvimento das folhas e 14.29% no peso.*

## 1. Introdução

Conforme [DESA 2017] a população mundial irá atingir em 2050 quase 10 bilhões de pessoas, tendo como desafio a produção de alimentos para todos. Este autor aponta como alternativas o uso de práticas sustentáveis e o aumento da produtividade, reduzindo assim os custos no uso de energia e água. Para atingir este objetivo ressalta-se a relevância de pesquisas voltadas para o auxílio da produção agrícola com a utilização da predição.

[Mohapatra and Lenka 2016] ressaltam que existe grande potencial em aplicações de Redes de Sensores sem Fio na área de Agricultura. Os autores relatam, no que tange a previsão em cultivos, que apenas um trabalho foi encontrado, refletindo a necessidade de futuros trabalhos nesta temática. Sugerem ainda pesquisas na área de monitoramento da vitalidade da plantação e previsão de rendimento utilizando serviços móveis e redes de sensores.

Uma vez analisados trabalhos relacionados à temática desta pesquisa, identificou-se que ainda faltam estudos que possam ser aplicados de fato em um cultivo prático. Por meio da revisão da literatura percebeu-se que são utilizadas tecnologias de comunicação

que suportam no máximo poucas centenas de metros, sendo que um cultivo geralmente é realizado em áreas que ocupam hectares e, por vezes, sem nenhum tipo de conectividade [Tripathy et al. 2011, Sawant et al. 2014, Liu et al. 2016, Mohapatra and Lenka 2016].

Para contribuir com o aumento da produtividade da produção agrícola, por meio da aplicação de novas tecnologias à agricultura de precisão, propõe-se o modelo Agri-Prediction. O modelo utiliza sensores distribuídos pelo campo para obter informações de um cultivo. Estas informações trafegam por uma rede de sensores sem fio (WSN) até uma aplicação onde são analisados utilizando o modelo de previsão ARIMA. Os dados são analisados e comparados com valores pré-estabelecidos pelo agricultor, caso os valores previstos pelo ARIMA estejam fora dos ideais, notifica-se o agricultor através de um dispositivo móvel, possibilitando a tomada de ações de forma a prevenir perdas. Objetivando analisar o modelo proposto, construiu-se uma estufa para o cultivo de rúcula (*Eruca sativa*), hortaliça escolhida por seu rápido desenvolvimento.

No decorrer desta pesquisa estão apresentados os trabalhos relacionados (Seção 2); o modelo AgriPrediction (Seção 3); a metodologia de avaliação (Seção 4); a análise da aplicação do modelo (Seção 5) e por fim, sua conclusão, bem como contribuição científica do trabalho e desafios futuros.

## 2. Trabalhos Relacionados

[Tripathy et al. 2011] realizaram um experimento para entender as relações entre doenças/pestes e clima utilizando sensoriamento sem fio com foco no Inseto Tripé e na peste Necrose do Broto. Já [Sawant et al. 2014] propuseram uma rede sem fio para integração de sistemas de sensoriamento voltado para agricultura de precisão, a qual denominou KrishSense. [Liu et al. 2016], trabalharam em um novo método de previsão baseado em um algoritmo Extreme Learning Machine (ELM) utilizado para prever a temperatura e a umidade em um ambiente de estufa prático em Nanjing, China.

[Yuhan et al. 2016] desenvolveram um experimento com plantas de tomateiro em uma estufa para modelagem de predição de taxa fotossintética. Os autores empregaram redes neurais para melhorar a precisão da solução. [Mohapatra and Lenka 2016] concentraram-se em duas estratégias de otimização utilizadas para prever a necessidade horária de CM no solo. [Edwards-Murphy et al. 2016] propuseram o modelo b + WSN, uma Rede de Sensores sem Fio heterogênea para coletar dados usando uma gama de parâmetros de uma colméia.

A Tabela 1 compara os artigos citados quanto a proposta de trabalho, coleta, exibição e análise dos dados, conectividade e localização dos sensores e local do experimento. Uma vez analisados os trabalhos relacionados, identificou-se que apesar de alguns pesquisadores estarem atuando com WSN na agricultura, ainda faltam estudos aplicados a um cultivo prático. As pesquisas utilizam tecnologias de comunicação que suportam poucas centenas de metros quando um cultivo geralmente é realizado em áreas que ocupam hectares, como já mencionado anteriormente.

Ressalta-se que poucos autores realizaram análises aprofundadas dos dados, desta forma, esta pesquisa se preocupa em fornecer não apenas dados do cultivo, mas possibilitar a tomada de decisões assertivas frente aos possíveis problemas. Além disto, os autores se limitam a realizar ações reativas somente quando algum limite é rompido. A tomada

**Tabela 1. Compartivo dos trabalhos Relacionados**

| Artigo                       | Análise           | Rede          | Dados    | Visualização | Localização | Aplicação  |
|------------------------------|-------------------|---------------|----------|--------------|-------------|------------|
| [Tripathy et al. 2011]       | Mineração         | ZigBee, GPRS  | Manual   | N/A          | India       | Amendoim   |
| [Sawant et al. 2014]         | N/A               | Wi-Fi         | Agendado | Twitter      | India       | Laranja    |
| [Liu et al. 2016]            | ELM               | ZigBee        | N/A      | N/A          | China       | Simulação  |
| [Yuhan et al. 2016]          | Rede Neural BP    | N/A           | Agendado | N/A          | China       | Tomate     |
| [Mohapatra and Lenka 2016]   | Rede Neural       | Zigbee, Wi-Fi | Agendado | N/A          | India       | Gramma     |
| [Edwards-Murphy et al. 2016] | Árvore de decisão | ZigBee, 3G    | Agendado | N/A          | Ireland     | Apicultura |

de ação de forma preditiva é melhor se comparada a reativa, pois, não exige que os limites sejam rompidos, mantendo o cultivo sempre dentro dos parâmetros.

### 3. Modelo AgriPrediction e Decisões do Projeto

De forma a possibilitar a aplicação do modelo em cenário real para que auxilie o produtor na previsão de problemas foram tomadas as seguintes decisões:

**Estrutura de comunicação WSN:** Optou-se pela utilização de Wireless Sensor Network (WSN) para coleta de dados do plantio e para garantir a conectividade entre os nós de sensores para transmitir os dados coletados dos nós para o servidor processá-los.

**Tecnologia de WSN LoRa:** O modelo propõe a construção de um link de rede entre as estações de coleta de dados e um Gateway utilizando a rede LPWAN LoRa<sup>1</sup>. Uma rede LoRa pode ter alcance de até 40km em campo aberto o que possibilita a real aplicação da solução proposta [Alliance 2015]. Além disto, LoRa é um protocolo de baixo consumo de energia, não necessita nenhum tipo de assinatura para uso e opera em frequências liberadas. Possui tráfego de dados adaptativos e é possível a conexão de milhões de nós, ficando a limitação no hub da rede. Características que possibilitam a sua utilização com os requisitos necessários para operar no campo.

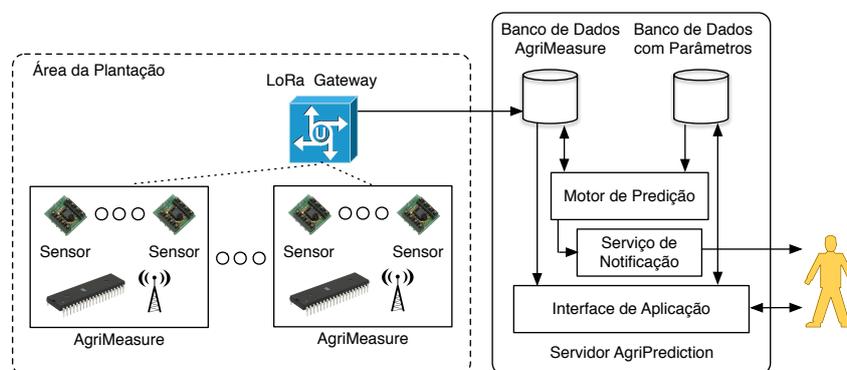
**Predição ARIMA:** São instrumentos de previsão cujo foco não está na construção de um modelo com equações únicas ou simultâneas, mas sim, na análise das propriedades probabilísticas das séries temporais, e assim, não são embasados em teorias econômicas [GUJARATI 2006]. A decisão de utilizar este modelo deu-se pelo fato de que os dados obtidos por esta pesquisa foram coletados sequencialmente ao longo do tempo, necessitando de uma análise de série temporal tendo o modelo apresentado resultados positivos em pesquisas com sua utilização.

#### 3.1. Arquitetura

A arquitetura do AgriPrediction foi dividida em AgriMeasure, responsável pela coleta das informações; Servidor AgriPrediction, responsável por armazenar as informações, hospedar os serviços de predição, notificação e a interface; Gateway, responsável por receber

<sup>1</sup><https://www.lora-alliance.org>

e encaminhar as informações dos AgriMeasures para o Servidor AgriPrediction e dispositivo móvel, que recebe as notificações. A Figura 1 exibe a representação do Modelo AgriPrediction.



**Figura 1. Arquitetura do AgriPrediction.**

Os dados do cultivo são lidos por meio de uma estação de leitura de sensores denominada AgriMeasure. Cada estação é composta de sensores; módulo de conectividade sem fio LoRa; micro controlador; placa solar; bateria e um identificador único de zona. O identificador de zona pode ser o ID do dispositivo, visando um menor custo, ou coordenadas obtidas através de uma placa de posicionamento GPS. O micro controlador é utilizado como solução para transformar os dados digitais e analógicos colhidos dos sensores para informações utilizáveis. Além disto, ele é responsável por enviar os dados através do chip de transmissão LoRa. A arquitetura do AgriMeasure é voltada para micro controladores open hardware.

Os sensores utilizados podem medir umidade, temperatura, luminosidade, radiação solar, condutividade elétrica do solo, PH, clorofila da planta entre outros parâmetros, basta configurar o sensor utilizado em uma entrada do micro controlador, que o mesmo encaminha as informações até um gateway LoRa. O Gateway recebe os dados no protocolo LoRa e os encaminha ao sistema de armazenamento de dados. Uma vez armazenados os dados, o sistema responsável por executar o cálculo do modelo ARIMA irá incluir as informações recentes. O resultado do cálculo é então comparado com os dados de referência inseridos pelo agricultor na interface da aplicação. Caso os valores previstos superem os limites inferiores ou superiores do cultivo em questão, o sistema de notificação é acionado informando o agricultor para que o mesmo possa tomar a ação. Pela interface da aplicação o agricultor pode acompanhar como estão os valores dos sensores utilizados, bem como configurar os valores de referência e o tempo de predição futuro em horas.

Para tornar o projeto viável financeiramente toda a arquitetura do projeto pode ser construída utilizando dispositivos open hardware. Por fim, considerando que o ambiente é de suma importância, é implementada uma topologia de rede estrela utilizando protocolo LoRa, pois este protocolo além de baixo consumo de energia possibilita a comunicação em até 40km. Isto torna a arquitetura viável para utilização no campo, onde fazendas possuem vários hectares.

### 3.2. Estratégia de Predição

O mecanismo de previsão apresentado no servidor funciona de acordo com o modelo de predição ARIMA. É adequado quando temos uma das seguintes situações: (i) quando a série temporal apresenta uma tendência; (ii) quando a série temporal não apresenta um componente sazonal, mas apenas o componente tendência; (iii) quando a série temporal é classificada como não estacionária e homogênea. O ARIMA considera a tendência de séries temporais com um triplo  $(p, d, q)$ , onde  $p$  é o número de termos autorregressivos,  $d$  é o número de diferenças e  $q$  é o número de termos e pode ser representado por  $\Theta(B)(1 - B)^d Y_t = \Theta(B)a_t$  [Sáfadi 2004].

Ao iniciar sua execução, o serviço de predição carrega as informações obtidas pelos AgriMesures que estão armazenadas dentro do banco de dados, sendo o mesmo responsável por organizar as séries dos dados coletados por cada sensor conectado. Por definição, somente as informações das últimas 24 horas são utilizadas, de forma a evitar tendências de valores do passado no cálculo da predição. Uma vez organizados os dados de cada sensor de cada AgriMeasure em suas séries iniciam-se os cálculos da predição. Ao executar o modelo, tem-se como resultado um valor máximo e um valor mínimo previsto para um dado momento no futuro. Por padrão, o modelo AgriPrediction define a previsão para a próxima hora, mas este parâmetro pode ser alterado pelo usuário.

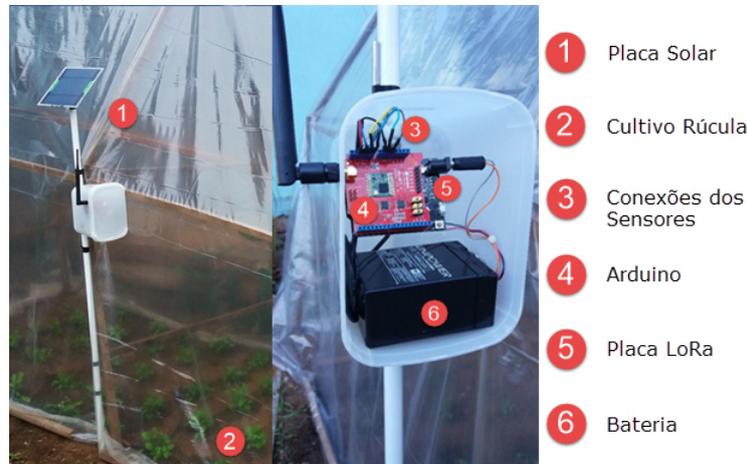
## 4. Metodologia de Avaliação

Cada AgriMeasure (estação de sensores) possui medição de temperatura e umidade do ar e do solo, luminosidade e um segundo sensor de temperatura externo, sendo este disposto fora da estufa de cultivo. Para medição da temperatura externa, o sensor ds18b20 foi utilizado. Este sensor trabalha com tensão de operação de 3 a 5,5V, possuindo atuação de -55 °C a +125 °C. Sua precisão é de 0,5 °C entre -10 °C a +85 °C. Para possibilitar seu uso externo o mesmo possui ponta de aço inoxidável de 6 x 50mm e é a prova d'água. Para transmissão de dados foi utilizado o protocolo One Wire.

Para medir a umidade e temperatura do solo, o sensor SHT10 foi utilizado em um encapsulamento de malha de metal para permitir seu uso sem danificá-lo. Este sensor opera com uma tensão de operação de 2,2V a 5,5V e permite a medição de umidade de 0 a 100 % RH, com uma precisão de 4,5 % e medição de temperatura de -40 °C a 120 °C com 0,5 % de precisão. Sua transmissão de dados ocorre de forma digital, através de comunicação serial. Para medir a luminosidade, o sensor TSL 2561, medindo de 0,1 a 45.000 lux, foi aplicado. Este sensor se comunica via protocolo I2C, consumindo 0,6 mA usando uma fonte de 3V.

Um Arduino Uno é utilizado para comunicação com os sensores, possui 14 pinos de entrada ou saída, dos quais 6 podem ser utilizados como saídas PWM e 6 entradas analógicas, controlados por um microcontrolador ATmega328 com clock de 16MHz suportando tensão de entrada de 6 a 20V e opera com 5V. Para alimentar o sistema, é utilizada uma placa solar de 6V de 3.8W e uma bateria auxiliar de 6V de capacidade de 4.5Ah. Cada AgriMeasure possui uma placa de comunicação LoRa que envia informações para o gateway LoRa. A placa utiliza o módulo SX1272 da Microchip e está conectada a uma antena omnidirecional de 3dbi. A frequência utilizada para comunicação é 868MHz. Este módulo possui sensibilidade de -134dBm e possui alcance de 21km em campo aberto

com enlace de linha visada e 2km em ambientes sem enlace de linha visada. A Figura 2 apresenta o protótipo do AgriPrediction.



**Figura 2. Protótipo do AgriPrediction**

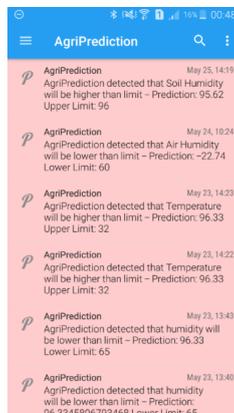
Para concepção do Gateway foi utilizado um Arduino Mega conectado a um módulo LoRa SX1272 de 868MHz. Este foi escolhido por possuir maior capacidade de memória e processamento para receber e processar os dados. Em seu software, foi configurado o módulo SX1272 com auxílio da biblioteca SX1272.h, distribuída pela Libelium de forma gratuita. Com o uso da biblioteca foi possível configurar o módulo LoRa e o Arduino para operar como receptor de mensagens, recebidas no primeiro canal da frequência 868MHz. Para comunicação com o SX1272 a biblioteca utiliza os pinos de SPI do Arduino para comunicação serial. Os dados recebidos pelo gateway através da placa LoRa são encaminhados para o servidor por comunicação serial, e no servidor são armazenados em arquivo. Para comunicação, uma antena externa omnidirecional de 2dbi foi conectada ao módulo LoRa, montada em uma haste de 3 metros.

#### 4.1. Implementação de Predição e Notificações

O arquivo contendo as leituras dos AgriMesures é lido pelo Software R a cada cinco minutos, onde o mesmo obtém as linhas referentes as últimas 24H de todos os nós. O R é um software estatístico desenvolvido por Ross Ihaka e Robert Gentleman. É ao mesmo tempo um ambiente de trabalho e uma linguagem de programação, que é escrita utilizando código descritivo. Os resultados são exibidos em formato texto ou exibição gráfica. É um software gratuito e de código aberto e suporta sistemas operacionais Linux, Mac OS e Windows [de Micheaux et al. 2013].

O script desenvolvido para o R é responsável pela separação das informações e executa o modelo ARIMA para cada um dos valores obtidos pelos sensores do AgriMeasure, obtendo a previsão de 10 próximos valores. Uma média destes valores é calculada e comparada com os valores preestabelecidos preenchidos previamente, presentes em um arquivo separado. Baseado nesta comparação o script verifica se é necessário informar um possível desvio. Caso necessário, as notificações são realizadas com a utilização da biblioteca pushover, que possibilita enviar uma notificação para um dispositivo móvel configurado com o aplicativo Pushover. Os valores obtidos pela predição são armazena-

dos em um outro arquivo, possibilitando o rastreamento das informações. A Figura 3 exibe as notificações recebidas por smartphone dos eventos ocorridos durante o experimento.



**Figura 3. Exemplo de Notificações recebidas em Smartphone**

#### 4.2. Cultivo de Rúcula

Selecionamos a rúcula como nossa cultura agrícola alvo, por ser um vegetal de fácil acesso e crescimento relativamente rápido, aproximadamente 30 a 40 dias após a semeadura [LINHARES 2007]. A área plantada mediu 3 metros de largura por 4 metros de comprimento e foi composta por uma estrutura de madeira e cobertura de filme plástico transparente, transformando-a em estufa. Foram plantadas um total de 100 mudas, semeadas na mesma data e uma semana após a germinação. Antes do plantio, o solo foi lixado por 20 centímetros e teve aplicação de lona plástica para permitir o controle da umidade do solo.

A estufa foi fisicamente dividida para a análise do modelo. De um lado, as ações foram tomadas ao longo da cultura para manter o plantio dentro dos parâmetros, enquanto o outro lado só foi regado a cada dois dias. O sensor de umidade e temperatura do solo foi posicionado a 10cm da haste AgriMeasure, enquanto o sensor de temperatura e umidade do ar foi disposto ao longo da mesma haste. O sensor de temperatura externo esteve abaixo da placa solar que estava no topo da haste em um ângulo de 45 graus. A Figura 4 ilustra o cultivo utilizado durante o experimento. Finalmente, o Arduino e a placa LoRa estavam localizados dentro de uma caixa de plástico protegida do clima. As conexões entre os dispositivos foram feitas por meio de fio elétrico compatível. O experimento foi realizado na cidade de Esteio, RS, em um local com clima ameno e sombra durante a maior parte do dia.

### 5. Análise da aplicação do AgriPrediction

Para testar o modelo, a estufa foi dividida em duas seções, sendo O cultivo mantido por três semanas. Então, realizou-se a coleta aleatória de cinco plantas de cada lado para diagnóstico. Cada planta foi pesada com uma balança de precisão e teve suas cinco maiores folhas medidas com um paquímetro, considerando-se a média das folhas. No decorrer do cultivo um dos lados recebeu água a cada dois dias (Lado A), enquanto o outro recebeu água conforme a necessidade monitorada, tendo umidade e temperatura controladas através da abertura ou fechamento da estufa e controle de incidência de luz (Lado B). O



**Figura 4. Estufa e sensores utilizados no experimento**

Gateway foi posicionado a 15 metros da plantação em um ambiente fechado com barreiras de construções entre a plantação e com sinal SNR em média de 8dBi. Este foi conectado a um servidor que estava executando o script desenvolvido para R, responsável por obter os dados coletados pelo Gateway e aplicar a predição nos dados.

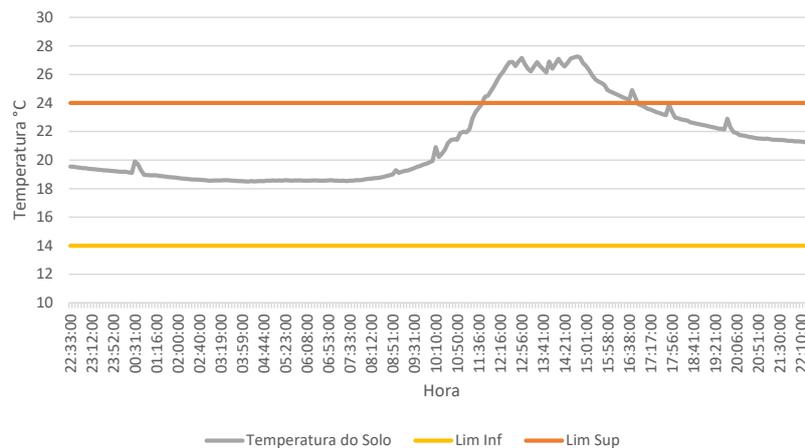
Um smartphone Android foi configurado para receber as notificações através do aplicativo Pushover. Os valores configurados como referência do cultivo foram de 16°C a 25°C de temperatura local e de 50% a 80% de umidade no ar; para o sensor no solo os valores foram de 14°C a 24°C e de 80% a 90% de umidade; por fim, o sensor de temperatura externo foi configurado de 15°C a 24°C. Os sensores AgriMesures foram ligados e o Gateway conectado utilizando um cabo USB e interface serial a um computador executando o software CoolTerm que recebe dados por uma porta serial e armazena em arquivo texto. Um script desenvolvido para o software R foi executado para ler as informações do arquivo de texto. Durante sua execução ele ignora as linhas iniciais do arquivo, que são de controle do Gateway, lê o conteúdo, organiza as informações dos sensores em objetos de séries temporais, executa o cálculo do modelo. Abaixo apresenta-se os resultados com e sem o uso do AgriPrediction.

Sem o AgriPrediction: As folhas obtiveram um tamanho médio de 7,67 cm e um peso médio de 0,0035 Kg. Neste experimento, nenhuma ação foi tomada contra os valores verificados pelos sensores AgriPrediction. Na Figura 5, podemos ver um gráfico do décimo dia de monitoramento da temperatura do solo deste experimento, onde a temperatura acabou por ultrapassar os limites pré-definidos (valores de referência ou limiares).

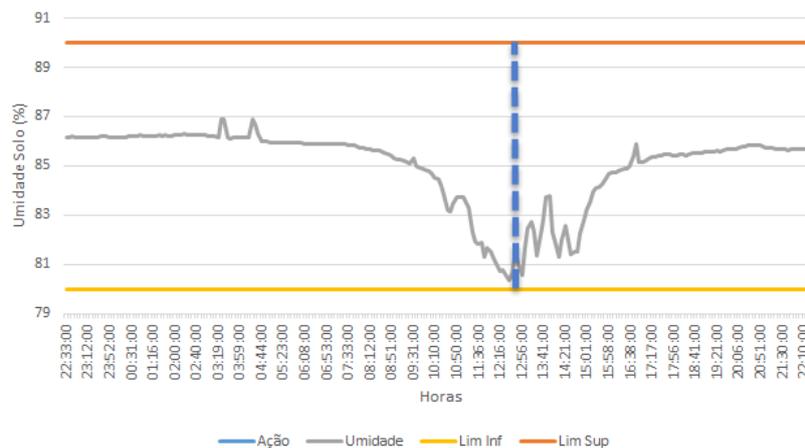
Com o AgriPrediction: As plantas obtiveram um tamanho médio de 9,04cm e peso médio de 0,004 Kg. Observa-se aqui que ao utilizar o monitoramento e realizar as ações sugeridas propostas pelo AgriPrediction foi possível obter um ganho de 17,94% quanto ao tamanho das folhas e 14,29% quando considerado o peso das plantas. A ideia é manter os legumes no chão sempre em condições ideais. Sempre que o algoritmo de previsão indicar um valor futuro fora das condições de cultivo seguras uma notificação ocorrerá no smartphone do agricultor, de modo que ações necessárias possam ser tomadas para corrigir/mitigar os problemas. Regar o plantio e controlar a umidade do ar representam as principais ações que fizemos neste experimento. A figura 6 mostra dados sobre a umidade do solo no décimo dia do experimento ao permitir as ações do AgriPrediction.

## 6. Conclusão

Este artigo apresentou um modelo de arquitetura de Redes de Sensores sem Fio (AgriPrediction) utilizando protocolo de rede LoRa para ser empregado por agricultores na



**Figura 5. Temperatura do solo em experimento sem usar previsão e notificações.**



**Figura 6. Umidade do solo e a interação do agricultor para corrigir eventuais problemas depois de receber uma notificação AgriPrediction.**

implementação de Agricultura de Precisão e Predição. Diferente dos trabalhos relacionados, o AgriPrediction trabalha com uma tecnologia que possibilita o seu uso em situações reais em conjunto com algoritmo de predição. Deste modo, possibilita a previsão de problemas, bem como a tomada de ações ativas. Sua contribuição está na área da agricultura combinando redes de sensores de curta e média cobertura e o modelo de predição ARIMA para o lançamento de ações corretivas e melhor rendimento de cultivos agrícolas.

Verificou-se que com a utilização do Arduino, chip LoRa SX1272, sensores SHT10, ds18b20 e TSL 2561 aplicados ao cultivo, obteve-se sucesso nos testes de envio e recebimento de mensagens LoRa, aplicação do modelo ARIMA e ganho de 17.95% no desenvolvimento das folhas e 14.29% no peso das plantas.

Os resultados obtidos sinalizam a viabilidade da proposta, entretanto, os autores reconhecem as limitações. Quanto aos trabalhos futuros pode-se incluir a criação de aplicativo para celular que possibilite o acompanhamento em tempo real do cultivo, além do desenvolvimento de mecanismo de segurança para a transmissão dos dados e testes em cultivos. Além disso, o modelo proposto tem como alvo a área da agricultura, porém, objetiva-se estender os testes do mesmo no contexto de cidades inteligentes, para monito-

ramento do trânsito.

## 7. Agradecimentos

Este trabalho é parcialmente financiado pelos seguintes órgãos de pesquisa brasileiros: CNPq, Fapergs e CAPES.

## Referências

- Alliance, L. (2015). Lorawan™ specification. *LoRa Alliance*.
- de Micheaux, P. L., Drouilhet, R., and Liquet, B. (2013). *The R Software*. Springer.
- DESA, U. (2017). World population prospects: The 2017 revision, key findings and advance tables. *Working PaperNo*.
- Edwards-Murphy, F., Magno, M., Whelan, P. M., O'Halloran, J., and Popovici, E. M. (2016). b+ wsn: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Computers and Electronics in Agriculture*, 124:211–219.
- GUJARATI, D. (2006). *N. econometria básica*. 3ª edição.
- LINHARES, P. C. F. (2007). *Produção de rúcula em função de diferentes quantidades e tempos de decomposição de jitrana*. PhD thesis, Dissertação (Mestrado em Fitotecnia)-Universidade Federal Rural do Semi-Árido. Mossoró, RN.
- Liu, Q., Jin, D., Shen, J., Fu, Z., and Linge, N. (2016). A wsn-based prediction model of microclimate in a greenhouse using extreme learning approaches. In *Advanced Communication Technology (ICACT), 2016 18th International Conference on*, pages 730–735. IEEE.
- Mohapatra, A. G. and Lenka, S. K. (2016). Neural network pattern classification and weather dependent fuzzy logic model for irrigation control in wsn based precision agriculture. *Procedia Computer Science*, 78:499–506.
- Sáfadi, T. (2004). Use of time series analysis for the water outflow at furnas dam. *Ciência e Agrotecnologia*, 28(1):142–148.
- Sawant, S. A., Adinarayana, J., and Durbha, S. S. (2014). Krishisense: A semantically aware web enabled wireless sensor network system for precision agriculture applications. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 4090–4093. IEEE.
- Tripathy, A., Adinarayana, J., Sudharsan, D., Merchant, S., Desai, U., Vijayalakshmi, K., Reddy, D. R., Sreenivas, G., Ninomiya, S., Hirafuji, M., et al. (2011). Data mining and wireless sensor network for agriculture pest/disease predictions. In *Information and Communication Technologies (WICT), 2011 World Congress on*, pages 1229–1234. IEEE.
- Yuhan, J., Yiqiong, J., Ting, L., Man, Z., Sha, S., and Minzan, L. (2016). An improved method for prediction of tomato photosynthetic rate based on wsn in greenhouse. *International Journal of Agricultural and Biological Engineering*, 9(1):146.

## Predição de dados de sensoriamento visando eficiência energética de redes de sensores sem fio

Charles M. P. Almeida<sup>1</sup>, Geymerson S. Ramos<sup>1</sup>, Andre L. L. Aquino<sup>1</sup>

<sup>1</sup>Instituto da Computação – Universidade Federal de Alagoas (UFAL)  
Maceió – AL – Brasil

{charles.pedrosa, geymerson.r, alla.lins}@gmail.com

**Abstract.** *This article aims to compare prediction models applied to wireless sensor networks in order to increase energy efficiency in data gathering. ARIMA, SVM and ANN models were deployed in an application to collect temperature data and for each model, an evaluation of its energy savings contribution has been made. The measurements were performed by real devices and it was possible to observe the performance of the models for a set of collected data in an office environment. In this study case, ARIMA model showed a better performance when compared to SVM and ANN in terms of energy efficiency.*

**Resumo.** *Este artigo tem como objetivo comparar modelos de predição de dados de sensoriamento em redes de sensores sem fio com a finalidade de economizar energia na coleta de dados. Modelos ARIMA, SVM e ANN foram utilizados em uma aplicação de coleta de dados de temperatura e avaliados quanto à economia de energia proporcionada. As medições foram realizadas por dispositivos reais e foi possível observar o desempenho dos modelos para um conjunto de dados de sensoriamento em um ambiente de escritório. Para este estudo de caso, o modelo ARIMA apresentou melhor desempenho em relação ao SVM e ANN em termos de eficiência energética.*

### 1. Introdução

As redes de sensores sem fio (RSSFs) [Akyildiz et al. 2002] permitem o monitoramento e controle de variáveis físicas como temperatura, som, luz, vibração, pressão, movimento ou poluição] [Ruela et al. 2009]. Nas RSSFs os nós sensores possuem geralmente energia limitada e a comunicação sem fio destes sensores pode ser considerada a principal forma de consumo. Portanto, reduzir a sobrecarga de comunicação pode ser o método mais eficaz para economizar energia nestes sistemas [Liu et al. 2005].

Em [Thamizhselvi and Mary 2016] é apresentada uma classificação das técnicas de eficiência energéticas nas RSSFs. Redução dos Dados é uma destas técnicas e seu foco está na redução do montante de dados produzidos, processados e transmitidos. Neste trabalho a redução do consumo de energia se dá através da redução da quantidade de transmissões entre nó sensor e na estação-base. A redução é obtida através da supressão do envio de parte dos dados coletados no nó sensor e sua substituição por dados gerados em um modelo de predição na estação-base. Diversos trabalhos nesta linha de pesquisa foram realizados [Liu et al. 2005, Wang et al. 2008, Moghadam and Keshmirpour 2011]. Em [Liu et al. 2005], utiliza-se um modelo ARIMA para predizer os valores dos dados

coletados. A acurácia e eficiência da predição para esta utilização pode ser melhor ou pior a depender do modelo que é utilizado.

Este artigo tem como objetivo comparar a eficiência energética em RSSFs a partir da supressão da comunicação entre nós utilizando modelos de predição de dados. Os modelos utilizados na comparação foram: i) ARIMA (do inglês, Autoregressive Integrated Moving Average) é um dos mais populares modelos lineares em previsão de séries temporais [Zhang 2003]; ii) SVM (do inglês, Support Vector Machines) [Cortes and Vapnik 1995] é um algoritmo de mineração de dados e considerado um dos métodos mais robustos e precisos entre os conhecidos algoritmos de mineração; iii) e ANN (do inglês, Artificial Neural Network) reúne seus conhecimentos detectando padrões e relacionamentos nos dados e aprendendo (ou sendo treinados) através da experiência, não da programação [Zhang et al. 1998].

Este artigo se organiza da seguinte forma: Seção 2 apresenta trabalhos relacionados com a pesquisa de eficiência energética e predição de dados utilizando os modelos ARIMA, SVM e ANN. Seção 3 trata dos modelos utilizados. Seção 4 detalha a aplicação desenvolvida para este estudo de caso. Seção 5 discute os resultados da comparação entre os modelos. Seção 6, conclusões e futuras direções de trabalho.

## 2. Trabalhos Relacionados

Diversos trabalhos apresentam abordagens para predição de leituras de sensores com o objetivo de economizar energia. Especificamente em [Liu et al. 2005] o objetivo é economizar energia na coleta de informações em redes de sensores sem fio. A metodologia empregada se baseia em manter os nós sensores da transmissão de informações redundantes, através da predição pelo nó coletor (estação-base). O ARIMA é utilizado como modelo para predição, propondo um esquema de coleta de informações eficiente em termos energéticos. A eficácia do método proposto apresentado no trabalho é demonstrada com uma aplicação real baseada na plataforma MICA2.

O trabalho de [Wang et al. 2008] investiga o problema da eficiência energética e propõe um método de organização eficiente de energia com previsão de séries temporais. A organização das redes de sensores sem fio está formulada para rastreamento de alvo. Modelo de alvo, modelo de multi-sensor e modelo de energia são definidos. A posição alvo futura é derivada de modelos de média móvel autorregressiva (ARMA). Os resultados experimentais verificam que a combinação do modelo ARMA e a decomposição do modo empírico podem estimar a posição alvo de forma eficiente e a economia de energia é alcançada pelo método de organização proposto.

A abordagem de [Moghadam and Keshmirpour 2011] se baseia no argumento que uma das maneiras de reduzir o consumo de energia dos nós sensores é reduzir o número transmissões sem fio entre nós por predição dupla.

Em linhas gerais existem uma diversidade de técnicas e soluções para o problema aqui apresentado. Contudo, diferente do que apresentamos nesse trabalho, não existe um estudo comparativo aprofundado do impacto das técnicas para a predição de dados de sensoriamento visando a eficiência energética em RSSFs.

### 3. Modelos de Predição

#### 3.1. Modelo ARIMA

A metodologia de Box-Jenkins para a previsão se baseia no ajuste de modelos denominados ARIMA para séries temporais de valores observados. A diferença entre os valores gerados pelos modelos e os valores observados resulta em séries de resíduos de comportamento aleatório em torno de zero. A análise de séries temporais usa os dados históricos para desenvolver um modelo para a previsão de valores. Neste trabalho os dados históricos são os valores coletados pelos nós sensores e enviados para a estação-base, onde é executado um modelo ARIMA para a predição dos dados futuros.

O modelo ARIMA [Liu et al. 2005] é amplamente utilizado para séries temporais univariadas. O mesmo é composto de três termos: O termo Auto Regressivo (AR), que é uma regressão linear do valor atual da série contra um ou mais valores anteriores. Ele captura a dependência do valor atual e seus valores anteriores mais próximos; O termo de Média Móvel (MA), que é introduzido para capturar a influência de choques aleatórios no futuro, uma vez que geralmente uma série temporal pode receber choques aleatórios em um ambiente ruidoso e pode memorizar os choques recebidos anteriormente por um tempo [Liu et al. 2005]; e o terceiro termo Integrador (I) para remover o efeito da não estacionariedade com a diferenciação dos dados, formando assim um modelo ARIMA. Vale destacar que o modelo ARMA, também utilizado, é a combinação dos dois primeiros termos, uma vez que assume que a série é estacionária. No entanto, nem sempre isso acontece, logo o terceiro termo precisa ser introduzido.

O modelo ARMA( $p, q$ ) refere-se ao modelo com  $p$  termos auto-regressivos e  $q$  termos na média móvel. O termo AR( $p$ ), de ordem  $p$ , pode ser descrito como

$$y_t = c + a_1 y_{t-1} + \dots + a_p y_{t-p} + u_t \quad (1)$$

onde  $a_1, \dots, a_p$  são parâmetros,  $c$  é uma constante, e a variável aleatória  $u_t$  é ruído branco. O termo MA( $q$ ), de ordem  $q$ , pode ser descrito como

$$y_t = \mu + u_t + m_1 u_{t-1} + \dots + m_q u_{t-q} \quad (2)$$

onde  $m_1, \dots, m_q$  são parâmetros do modelo,  $\mu$  é a expectativa para  $y_t$  (normalmente 0),  $u_t, u_{t-1}, \dots, u_{t-q}$  são termos de erro de ruído branco. O modelo contendo os AR( $p$ ) e MA( $q$ ) pode ser descrito como

$$y_t = c + a_1 y_{t-1} + \dots + a_p y_{t-p} + u_t + m_1 u_{t-1} + \dots + m_q u_{t-q} \quad (3)$$

Quando AR( $p$ ), MA( $q$ ) e ARMA( $p, q$ ) são aplicados em alguns casos onde os dados mostram evidência de não-estacionariedade, uma diferenciação inicial deve ser aplicada para reduzir esta não-estacionariedade, chegando-se assim a um modelo ARIMA. Modelos não sazonais são geralmente denotados por ARIMA( $p, d, q$ ), onde  $p$  é a ordem do modelo AR,  $d$  é o grau de diferenciação e  $q$  é a ordem do modelo MA [Yuan et al. 2016].

#### 3.2. Modelo SVM

As máquinas de vetor de suporte (SVMs) implementam o princípio da minimização de risco estruturada (SRM, do inglês, Structured Risk Minimization) em vez do princípio de

minimização de risco empírico implementado pela maioria dos tradicionais modelos de rede neural. O conceito mais importante de SRM é a minimização de um limite superior para o erro de generalização em vez de minimizar o erro de treinamento. Com base nesse princípio, as SVMs conseguem uma ótima estrutura de redes [Pai and Hong 2005].

Os modelos SVMs geram a função de regressão aplicando um conjunto de funções lineares de altas dimensões. A função de regressão SVM é formulada da seguinte forma [Pai and Lin 2005]

$$y = w\phi(x) + b \quad (4)$$

onde  $\phi(x)$  é chamado de recurso (*feature* em inglês), que é oriunda do mapeamento não linear do espaço de entrada de  $x$ . Os coeficientes  $w$  e  $b$  são estimados pela minimização

$$R(C) = C \frac{1}{N} \sum_{i=1}^N L_{\varepsilon}(d_i, y_i) + \frac{1}{2} \|w\|^2 \quad (5)$$

onde

$$L_{\varepsilon}(d, y) = \begin{cases} |d - y| - \varepsilon & |d - y| \geq \varepsilon \\ 0 & \text{outros} \end{cases} \quad (6)$$

$C$  e  $\varepsilon$  são parâmetros prescritos. O primeiro termo  $L_{\varepsilon}(d, y)$  é chamado de função de perda  $\varepsilon$ -itensiva. O  $d_i$  é o valor atual da série no período de índice  $i$ . Esta função indica que o erro abaixo  $\varepsilon$  não é penalizado. O termo  $C \frac{1}{N} \sum_{i=1}^N L_{\varepsilon}(d_i, y_i)$  é o erro empírico. O segundo termo,  $\frac{1}{2} \|w\|^2$  mede a planicidade da função.  $C$  avalia o *trade-off* entre o risco empírico e a planicidade do modelo. Introduzindo as variáveis de folga positivas  $\zeta$  e  $\zeta^*$ , que representam as distâncias dos valores reais para os valores limites de  $\varepsilon$ -tubo. A equação 5 é transformada na seguinte forma

Minimizar:

$$R(w, \zeta, \zeta^*) = \frac{1}{2} ww^T + C^* \left( \sum_{i=1}^N (\zeta_i + \zeta_i^*) \right) \quad (7)$$

Sujeito a:

$$w\phi(x_i) + b_i - d_i \leq \varepsilon + \zeta_i^* \quad (8)$$

$$d_i - w\phi(x_i) - b_i - d_i \leq \varepsilon + \zeta_i \quad (9)$$

$$\zeta_i, \zeta_i^* \geq 0 \quad (10)$$

com  $i = 1, 2, \dots, N$ . Introduzindo o multiplicador de Lagrange e maximizando a função dupla da Equação 7, temos

$$R(\alpha_i - \alpha_i^*) = \sum_{i=1}^N d_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_{i=1}^N (\alpha_i - \alpha_i^*) - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) \times (\alpha_j - \alpha_j^*) K(x_i, x_j) \quad (11)$$

com as restrições

$$\sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad (12)$$

sendo  $0 \leq \alpha_i \leq C$  e  $0 \leq \alpha_i^* \leq C$  com  $i = 1, 2, \dots, N$ . Na Equação 11,  $\alpha_i$  e  $\alpha_i^*$  são chamados de multiplicadores de Lagrange [Pai and Lin 2005]. Eles satisfazem as igualdades

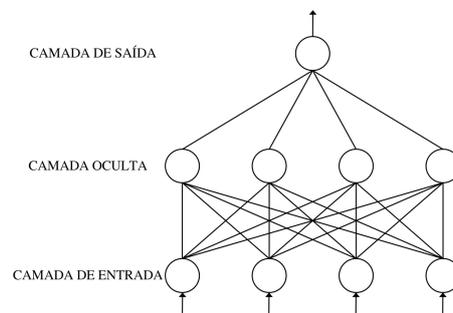
$$\alpha_i * \alpha_i^* = 0, \quad (13)$$

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \quad (14)$$

onde  $K(x, x_i)$  é a função Kernel. O valor da Kernel é igual ao produto interno dos vetores  $x_i$  e  $x_j$  no espaço do recurso  $\phi(x_i)$  e  $\phi(x_j)$ , de modo que  $K(x_i, x_j) = \phi(x_i) * \phi(x_j)$ . Qualquer função que satisfaça a condição de Mercer pode ser usada como função Kernel [Pai and Lin 2005]. A condição Mercer é a regra para determinar uma Kernel admissível de vetor de suporte [Xiao-Ying et al. 2009]. Neste estudo utilizamos função Kernel do tipo radial.

### 3.3. Modelo ANN

O tipo de rede neural utilizada neste trabalho foi a multi-layer perceptrons (MLP). As redes MLP são usadas na predição devido a sua capacidade de mapeamento arbitrário de entrada e saída. Uma rede MLP é tipicamente composta por várias camadas. A primeira camada serve como entrada e a última é uma camada de saída onde a solução do problema é obtida. A camada de entrada e a camada de saída são separadas por uma ou mais camadas intermediárias, chamadas camadas ocultas. Os elementos das camadas adjacentes (neurônios artificiais ou nós) são totalmente conectados por arcos acíclicos de uma camada inferior a uma camada superior. A Figura 1 dá um exemplo de um MLP totalmente conectado com uma camada oculta [Zhang et al. 1998]. Para o problema de predição, as



**Figura 1. Uma típica rede neural feedforward**

entradas são geralmente as variáveis independentes ou preditoras. A relação funcional estimada pela ANN pode ser escrita como

$$y = f(x_1, x_2, \dots, x_p) \quad (15)$$

onde  $x_1, x_2, \dots, x_p$  são as  $p$  variáveis independentes e  $y$  é a variável dependente. Nesse sentido, a rede neural é funcionalmente equivalente a um modelo de regressão não linear. Para um problema de predição de séries temporais, as entradas são tipicamente as observações passadas e a saída é o valor futuro. A ANN executa o seguinte mapeamento de funções

$$y_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-p}) \quad (16)$$

onde  $y_t$  é a observação no tempo  $t$ . Assim, o modelo ANN é equivalente ao modelo autorregressivo não linear para problemas de predição de séries temporais.

O modelo ANN necessita ser treinado antes de realizar a tarefa desejada. O treinamento é o processo de determinar os parâmetros do modelo (geralmente ajuste dos pesos) e ele pode ocorrer de forma supervisionada ou não supervisionada. Para o problema de predição em séries temporais, um padrão de treinamento consiste em um número fixo de observações defasadas da série. Ou seja, suponha que temos  $N$  observações  $y_1, y_2, \dots, y_N$  no conjunto de treinamento e precisamos da predição em um passo à frente. Usando um modelo ANN com  $n$  nós de entrada, temos  $N - n$  padrões de treinamento. O primeiro padrão de treinamento será composto de  $y_1, y_2, \dots, y_n$  como entradas e  $y_{n+1}$  como a saída alvo. O segundo padrão de treinamento conterá  $y_2, y_3, \dots, y_{n+1}$  como entradas e  $y_{n+2}$  como a saída desejada. Finalmente, o último padrão de treinamento será  $y_{N-n}, y_{N-n+1}, \dots, y_{N-1}$  para entradas e  $y_N$  para o alvo. Comumente uma função objetivo utilizada durante o processo de treinamento é dada por

$$E = \frac{1}{2} \sum_{i=n+1}^N (y_i - \alpha_i)^2 \quad (17)$$

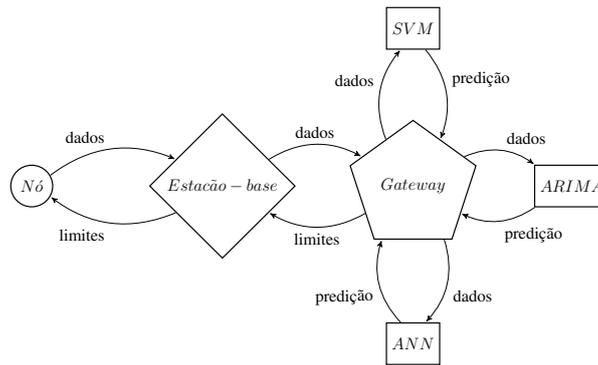
onde  $\alpha_i$  é a saída real da rede e o valor  $1/2$  é o fator de simplificação da expressão de derivadas computadas no algoritmo de treinamento.

#### 4. Aplicação

Para coleta dos dados de temperatura, 5 dispositivos da plataforma MICAZ<sup>1</sup> foram utilizados, sendo 4 destes utilizados como nós (sensores) e 1 como estação-base. A estação-base foi conectada a um computador de mesa que atuará como *gateway*. No *gateway* é feito processamento de dados e nele são executados os modelos ARIMA, SVM e ANN. Os nós foram colocados próximos um do outro para mensurar simultaneamente sob as mesmas condições do ambiente e o distanciamento de cada nó para a estação-base é de aproximadamente 5 centímetros. Um diagrama da aplicação é apresentado na Figura 2, onde é observada a relação entre nó, estação-base e o *gateway*.

Cada dispositivo foi programado conforme seu papel na RSSF, sendo que, os nós sensores mensuram e transmitem dados de temperatura diretamente para a estação-base, que por sua vez os recebe e os transmite para o *gateway*. Ao receber os dados, o *gateway* processa e gera um conjunto de predições para um dos modelos citados, isto é, um conjunto de predições para o ARIMA, SVM ou ANN, a depender do nó que realizou a transmissão.

<sup>1</sup>[http://www.memisc.com/userfiles/files/Datasheets/WSN/micaz\\_datasheet-t.pdf](http://www.memisc.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf), acessado em março de 2018



**Figura 2. Diagrama da aplicação**

Cada conjunto de previsões possui 30 valores gerados a partir das 30 últimas temperaturas transmitidas e, com os valores preditos, calcula-se uma média. Com a média, são definidos limites superiores e inferiores que serão utilizados como parâmetros para determinar quando um nó deve ou não transmitir um pacote de dados. O limite superior é dado pela soma de um valor  $\delta$  à média de um conjunto de previsões e, de maneira análoga, o limite inferior é dado pela subtração deste valor da média. O *gateway* é responsável por calcular estes limites e transmiti-los para para um determinado nó da rede através da estação-base.

Cada nó mensura e armazena localmente dados de temperatura de segundo em segundo e, ao final de 30 segundos, os transmite para a estação-base, caso os limites calculados pelo *gateway* tenham sido ultrapassados. Para fins de comparação, as transmissões foram feitas de forma contínua (o nó sempre transmitindo a cada 30 segundos) e intermitente (transmite às vezes). Dos quatro nós sensores utilizados, além da estação-base, um destes realizou transmissão continuamente e os outros três transmitiram apenas no caso de ultrapassagem dos limites superiores ou inferiores, sendo que, um nó teve seus limites definidos com base nas previsões do ARIMA, um nas previsões do SVM e o terceiro baseado em previsões do ANN. Caso um nó ultrapasse seu limite superior ou inferior, um pacote de medições é transmitido, e o *gateway* calcula novos limites e os transmite para o nó.

O *gateway* mantém versões virtuais dos nós físicos da rede, ambos funcionam em sincronia de modo que, sempre que um nó físico não realizar uma transmissão, o *gateway* assumirá que os valores das medições estão dentro dos limites calculados e o nó virtual armazenará um conjunto de temperaturas preditas por seu respectivo algoritmo. Este conjunto de temperaturas será armazenado repetidamente até que um novo pacote de dados seja recebido e novas previsões sejam realizadas. Nesta situação, os valores reais de temperatura recebidos serão armazenados e, passados 30 segundos, se um novo pacote não for recebido, o nó virtual armazenará periodicamente valores de previsão baseados nos valores reais. A curva final de temperatura de um nó é portanto constituída de valores reais e preditos.

## 5. Avaliações

O erro dos modelos de previsão propostos foi avaliado pelo MAPE (do inglês, Mean Absolute Percentage Error), MAD (do inglês, Mean Absolute Deviation) e pelo NRMSE

(do inglês, Normalized Root Mean Square Error), como determinado pelas Equações 18, 19 e 20, respectivamente. Foram comparadas a curva real de temperatura e as curvas preditas nos nós sensores. O resultado destas avaliações pode ser verificado na tabela 1.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{d_i - y_i}{d_i} \right| \times 100\% \quad (18)$$

$$MAD = \frac{\sum_{i=1}^n |d_i - y_i|}{n} \quad (19)$$

$$NRMSE = \sqrt{\frac{\sum_{i=1}^n (d_i - y_i)^2}{\sum_{i=1}^n d_i^2}} \quad (20)$$

A amostra avaliada corresponde a um período de aproximadamente 6 horas com 23.070 medições de temperatura em cada um dos nós. Apesar de estarem na mesma localidade, os nós apresentaram variações nas temperaturas mensuradas. O nó com previsões dadas pelo modelo ARIMA realizou o menor número de transmissões, seguido pelo SVM e por último, ANN.

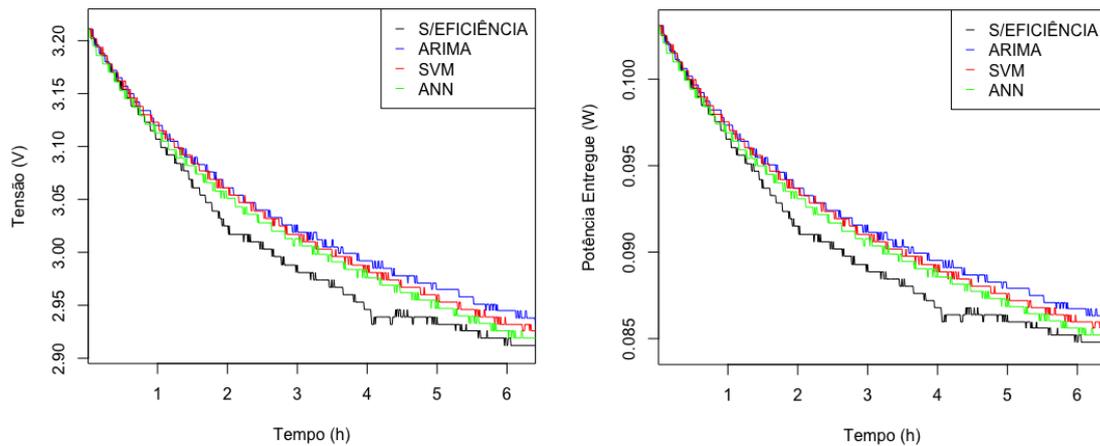
|       | ARIMA | SVM   | ANN   |
|-------|-------|-------|-------|
| MAPE  | 1,98% | 2,07% | 2,75% |
| MAD   | 0,47  | 0,49  | 0,65  |
| NRMSE | 0,02  | 0,02  | 0,03  |

**Tabela 1. Avaliação de Erro dos Modelos**

Foi realizada também avaliação com a finalidade de comparar o desempenho dos modelos ARIMA, SVM e ANN na predição de dados de sensoriamento visando eficiência energética na RSSF. Durante o fornecimento de potência elétrica de um conjunto de baterias para uma aplicação, o descarregamento destas baterias pode ser graficamente visualizada através de uma curva onde temos no eixo  $x$  a variável tempo e no eixo  $y$  a variável tensão elétrica. Durante o ciclo de funcionamento, a tensão elétrica decai de forma suave até valores toleráveis pela aplicação, e então, diminui abruptamente fazendo com que o sistema não opere mais, devido ao fato de não estar com o nível de tensão inadequado. Esta curva pode ser chamada de curva de descarregamento das baterias. Na Figura 3, lado esquerdo, temos uma amostra de aproximadamente 6h onde podemos ver parte do decaimento da tensão das baterias em cada um dos quatro nós durante o sensoriamento da temperatura. Ainda na Figura 3, lado direito, são apresentadas as curvas de Potência Entregue em cada um dos nós. A curva de Potência Entregue é proporcional ao quociente do quadrado da tensão pela resistência do circuito e interna das baterias.

Para a avaliação do modelo que apresenta maior eficiência energética no sensoriamento de temperatura foi verificada a quantidade de energia, correspondente às diferenças das áreas sob as curvas do gráfico da Figura 3, de cada um dos modelos em relação ao nó que sempre envia seus dados, desta forma, verificando quanto a supressão do envio de dados trouxe de economia de energia para as baterias.

O modelo ARIMA apresentou, para esta amostra, uma maior eficiência energética na RSSF. Na Tabela 2 podemos verificar os valores percentuais de economia de energia



**Figura 3. Curva de Descarga das Baterias e Curva de Potência Entregue**

|                               | ARIMA | SVM   | ANN   |
|-------------------------------|-------|-------|-------|
| Redução do Consumo de Energia | 1,90% | 1,53% | 1,07% |

**Tabela 2. Eficiência Energética dos Modelos**

para as baterias no intervalo de tempo supracitado. O nó com o modelo ARIMA economizou cerca de 2% de energia quando comparado ao nó que sempre transmite. Vale salientar que nesta amostra de aproximadamente 6 horas, o valor da constante  $\delta$  incrementada ou somada para os limites superiores e inferiores respectivamente era de apenas 0,03 graus Celcius e, a depender destes limites e do intervalo de tempo da coleta, o valor de economia certamente será bem maior.

## 6. Conclusão e trabalhos futuros

Nas RSSFs o gasto de energia elétrica devido à transmissão de dados entre os nós sensores e a estação-base, em geral, a maior parcela do consumo. Diversas abordagens no sentido de suprimir o envio das informações existem e uma delas é a supressão do envio de dados através da predição dos valores num modelo matemático [Liu et al. 2005]. Neste artigo comparamos três modelos ARIMA, SVM e ANN para verificar qual apresenta maior eficiência energética.

O modelo ARIMA apresentou, para as amostras coletadas, um desempenho superior aos modelos SVM e ANN, sendo mais preciso nas predições e economizando mais energia das baterias da RSSF. Trabalhos anteriores já avaliaram o modelo ARIMA isoladamente aplicado para eficiência energética de RSSF, no entanto este trabalho ratifica a sua qualidade quando comparado com outros modelos.

Os modelos se mostraram sensíveis quanto à escolha dos seus parâmetros e uma abordagem de otimização ou utilizando modelos híbridos pode contribuir para melhorar a análise em estudos futuros.

## Agradecimentos

Agradecemos ao CNPq, FAPEAL e FAPESP pelo apoio financeiro.

## Referências

- Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422.
- Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.
- Liu, C., Wu, K., and Tsao, M. (2005). Energy Efficient Information Collection with the ARIMA model in Wireless Sensor Networks. *IEEE Communications*.
- Moghadam, R. A. and Keshmirpour, M. (2011). Hybrid ARIMA and Neural Network Model for Measurement estimation in Energy-Efficient Wireless Sensor Networks. In *Informatics Engineering and Information Science, PT III*, volume 253 of *Communications in Computer and Information Science*, pages 35–48.
- Pai, P. and Hong, W. (2005). Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conversion and Management*, 46(17):2669–2688.
- Pai, P. and Lin, C. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega-International Journal of Management Science*, 33(6):497–505.
- Ruela, A. S., Cabral, R. S., and Aquino, Andre L. L. Guimaraes, F. G. (2009). Evolutionary design of wireless sensor networks based on complex networks. In *Proceedings of the 2009 Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 237–242.
- Thamizhselvi, S. and Mary, P. S. (2016). A Survey about Data Prediction in Wireless Sensor Networks with Improved Energy Efficiency. *Research Journal of OF Pharmaceutical Biological and Chemical Sciences*, 7(2):2118–2120.
- Wang, X., Wang, S., Ma, J., and Bi, D. (2008). Energy efficient organization of wireless sensor networks with adaptive forecasting. *SENSORS*, 8(4):2604–2616.
- Xiao-Ying, D., Ding-Hui, Y., Tao, L., and Jing, X. (2009). Study on Mercer condition extension of support vector regression based on Ricker wavelet kernel. *Chinese Journal of Geophysics-Chinese Edition*, 52(9):2335–2344.
- Yuan, C., Liu, S., and Fang, Z. (2016). Comparison of china’s primary energy consumption forecasting by using arima (the autoregressive integrated moving average) model and gm(1,1) model. *Energy*, 100:384 – 390.
- Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50(Supplement C):159 – 175.
- Zhang, G., Patuwo, E., and Hu, M. Y. (1998). Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*, 14(14):35–62.

# Seleção de Nó Raiz Ótimo para Sincronização Eficiente de Relógios em Redes de Sensores Sem Fio

Tiago C. de S. Xavier<sup>1</sup>, Claudio L. Amorim<sup>1</sup>, Flavia C. Delicato<sup>1</sup>

<sup>1</sup>Universidade Federal do Rio de Janeiro (COPPE-UFRJ)  
Rio de Janeiro – RJ – Brasil

{tiagocariolano, amorim}@cos.ufrj.br, fdelicato@dcc.ufrj.br

**Abstract.** *To reduce clock synchronization error in wireless sensor networks, recent works aimed to optimize network topology. Such approaches presented limited performance since the root node, who provides global clock, may remain being located in unfavorable positions after topological optimization. Moreover, in case of node failures, recuperation is a very costly operation. This work presents two proposals to select network optimal root node: the first is a graph-based approach and the second is a distributed algorithm which it is applied to a failure scenario. Experiments show that selecting optimal root node, rather than optimizing topology, decreases synchronization error close to six times less and distributed approach is able to be robust to failures of the root node.*

**Resumo.** *Para reduzir o erro de sincronização de relógios em redes de sensores sem fio, trabalhos recentes buscaram otimizar a topologia da rede. Essas abordagens apresentaram desempenho limitado pois o nó raiz, o qual fornece o relógio global, pode continuar mal localizado mesmo após a otimização e em caso de falha, a recuperação é muito custosa. Este trabalho propõe duas abordagens para selecionar o nó raiz ótimo da rede: a primeira é baseada em grafos e a segunda é um algoritmo distribuído aplicado em um cenário de falha. Experimentos indicam que, comparada à otimização de topologia, a seleção do raiz ótimo reduz cerca de seis vezes o erro de sincronização e que a abordagem distribuída é capaz de ser resiliente em cenários de falha do nó raiz.*

## 1. Introdução

Redes de sensores sem fio (RSSFs) têm sido cada vez mais utilizadas como poderosos sistemas de aquisição de dados. Suas características de flexibilidade, facilidade de implantação e baixo custo as tornam atraentes para uma ampla gama de domínios de aplicações [Kamilaris and Pitsillides 2016].

Um dos principais problemas no projeto RSSF de larga escala consiste na sincronização temporal dos relógios dos diversos nós. O objetivo da sincronização de relógios é fazer com que todos os nós tenham uma visão comum do tempo. A principal forma de sincronizar relógios de maneira estável e precisa é utilizar um nó como referência e, por meio de troca de mensagens de sincronização, fazer com que todos os nós estimem o valor de relógio deste nó (chamado de raiz) [Maróti et al. 2004, Yıldırım et al. 2018, Cardoso et al. 2017, Huang et al. 2014, Lenzen et al. 2015, Upadhyay et al. 2018]. A principal dificuldade com esta abordagem é que, conforme se aumenta a escala da rede, os nós que estão a mais de um salto de distância do nó raiz apresentam um erro de sincronização maior. Isso ocorre pois o erro de sincronização é cumulativo em função da distância para o raiz. Os autores em

[Lenzen et al. 2008] demonstraram teoricamente que o diâmetro da rede é correlacionado com o erro de sincronização na rede. Em seu estudo eles mostraram que, quanto maior a distância de um nó para o raiz, maior o erro do protocolo.

Para tentar mitigar este problema, trabalhos recentes tentam melhorar a sincronização de relógios explorando propriedades topológicas da rede para atenuar o problema do aumento da escala [Li et al. 2014, Panigrahi and Khilar 2015, Su et al. 2016]. Tais abordagens consistem em mudar virtualmente a topologia da rede ou escolher nós específicos por onde as mensagens irão trafegar. A topologia é modificada definindo *links* de comunicação entre os nós, forçando as mensagens a seguir determinada rota. A escolha estratégica de nós consiste em selecionar nós que iniciem a sincronização ou sejam responsáveis por repassar as mensagens de sincronização. Tanto os *links* quanto os nós são escolhidos baseados em parâmetros críticos do protocolo de sincronização.

O principal parâmetro considerado nesse tipo de abordagem é a distância em *hops* dos nós para o raiz, mas outros como a densidade da rede, a contenção do canal ou uma combinação deles podem ser utilizados. A distância em *hops* para o nó raiz é um dos principais fatores que afetam os protocolos de sincronização, pois quando o nó raiz inicia uma rodada de resincronização existe um atraso entre o instante em que a mensagem sai do emissor e chega a cada um dos nós vizinhos. Esse atraso é decorrente do tempo de envio e recebimento da mensagem, tempo de transmissão da mensagem no enlace e tempo de propagação da mensagem no meio físico.

As abordagens de otimização topológica para mitigar este problema são limitadas em desempenho, porque mesmo que alguns caminhos das mensagens sejam otimizados ou que alguns nós sejam escolhidos para encaminhar os pacotes de maneira eficiente, o nó raiz pode estar mal localizado (por exemplo quando está em uma região distante dos outros nós da rede). Neste cenário, o erro de sincronização pode se tornar proibitivo para aqueles nós mais distantes do raiz. Além disso, abordagens de otimização de topologia possuem um alto custo em casos de falha de nós, principalmente do raiz. Quando ocorre uma falha, o algoritmo de otimização de topologia deve ser reexecutado e toda a rede reconfigurada, caso se deseje manter o erro de sincronização em níveis aceitáveis.

Visando abordar o problema do erro cumulativo de protocolos de sincronização baseados em nó raiz, apresentamos duas estratégias que exploram propriedades de topologia da rede para atenuar o erro de sincronização conforme se aumenta a distância dos nós para o nó raiz. A primeira estratégia consiste em um algoritmo baseado em grafos que seleciona o nó que minimiza as distâncias para todos os outros da rede. Ao invés de otimizar toda a topologia, a estratégia realiza uma seleção do nó raiz ótimo considerando a distância em *hops*. O algoritmo possui complexidade de tempo polinomial e é executado *offline*, durante o projeto da rede, antes do posicionamento dos nós. A segunda estratégia é um novo algoritmo que resolve o mesmo problema de selecionar o nó ótimo da rede, porém realiza isso de maneira distribuída. Este algoritmo distribuído é aplicado em situações de falha do nó raiz, é executado durante a operação da rede. É importante ressaltar que a estratégia proposta não requer qualquer dispositivo GPS (*Global Positioning System*), em qualquer nó sensor, o qual é um fator importante para a construção de RSSFs de baixo custo.

As principais contribuições deste trabalho são:

- Um algoritmo para seleção do nó raiz ótimo baseado em grafos. Este algoritmo é executado na implantação da rede e serve para encontrar o nó que minimiza as

distâncias para todos os outros da rede e possui complexidade de tempo polinomial, enquanto estratégias como [Li et al. 2014, Panigrahi and Khilar 2015], que fazem otimização topológica, são algoritmos de complexidade exponencial.

- Um algoritmo distribuído para resolver o problema de seleção de nó raiz ótimo em cenários de falhas do nó raiz. Este algoritmo é executado durante a operação dos nós sensores, monitorando por falhas do nó raiz e quando esse evento ocorre, disparando ações para mitigar o problema.

Resultados de simulações demonstraram que nossa abordagem de seleção de nó raiz baseada em grafos supera em quase seis vezes o ganho em relação ao erro de sincronização de uma abordagem que otimiza a topologia da rede. Além disso, experimentos exaustivos demonstraram que a abordagem distribuída é capaz de manter o ganho de erro de sincronização obtido pela abordagem baseada em grafos, mesmo em situações de falhas do nó raiz.

## 2. Trabalhos Relacionados

Muitas estratégias têm sido desenvolvidas para melhorar a acurácia de sincronização, porém sem considerar a localização dos nós, ou fazendo isso com o auxílio de dispositivos GPS. [Benzaïd et al. 2017] desenvolveram um mecanismo que divide a rede em *clusters* sincronizados e permite aos nós utilizarem mensagens de outros *clusters*, a fim de reduzir o *overhead* de comunicação. [Cardoso et al. 2017] propuseram uma solução híbrida para o problema de localização e de sincronização, em que um veículo aéreo não-tripulado equipado com um GPS é responsável por enviar as mensagens de sincronização aos nós. [Yıldırım et al. 2018] propuseram um protocolo de sincronização de relógios que aplica um mecanismo de correção de *feedback* proporcional e integral nos erros das mensagens de sincronização. [Upadhyay et al. 2018] desenvolveram um método de correção gaussiano para as diferenças de *offsets* entre os relógios dos nós.

Alguns trabalhos procuraram otimizar a topologia da rede, a fim de controlar adequadamente a comunicação envolvida na sincronização. [Li et al. 2014] propuseram uma estratégia para reduzir logicamente o diâmetro da rede a fim de diminuir o impacto do aumento da escala em termos de número de nós. Eles fizeram isso por meio da criação de um *overlay* virtual da rede, de modo que o caminho que uma mensagem passa do raiz até qualquer nó é significativamente reduzido. O problema gerado pelo algoritmo dos autores é NP-Completo. [Panigrahi and Khilar 2015] propuseram uma estratégia de otimização da topologia da rede, em que o algoritmo proposto é aplicado para protocolos de sincronização baseados em consenso. Ele seleciona um subconjunto de nós e seus vizinhos que serão responsáveis por iniciar a sincronização. Este problema de seleção é mapeado para o problema de encontrar o conjunto dominante conectado de um grafo, o qual é NP-Completo, e é resolvido por meio de um novo algoritmo genético. [Su et al. 2016] desenvolveram um esquema que seleciona nós para controle de mensagens de sincronização num ambiente de redes náuticas *ad hoc*. Inicialmente a rede é dividida em camadas, de acordo com a distância em *hops* dos nós, e em cada camada, pelo menos um nó previamente selecionado é o responsável por repassar as mensagens de sincronização para os nós regulares. A principal diferença em relação ao presente trabalho é que na abordagem de [Su et al. 2016] é que ocorre uma seleção de nós por camadas, ao invés do nó raiz ótimo da rede, de modo que o problema de má localização do nó raiz não é solucionado.

### 3. Seleção de Nó Raiz Ótimo

#### 3.1. Modelo de Rede e Definição do Problema

A rede de nós sensores é modelada como um grafo não-direcionado  $G = (V, E)$ , em que  $V$  é um conjunto de  $n = |V|$  nós e  $E$  é um conjunto de arestas que representam *links* bidirecionais de comunicação. O conjunto de nós dentro de uma região de *broadcast* de qualquer nó  $u \in V$  corresponde a seu conjunto de vizinhos e é denotado por  $N_u = \{v \in V | (u, v) \wedge (v, u) \in E\}$ . Um único nó da rede  $r \in V$  é denotado como o nó raiz e é responsável por iniciar a sincronização. O custo de uma aresta conectando dois vértices  $u \wedge v \in V$  é dado por  $w(u, v) = 1$ , que significa que  $u$  e  $v$  estão na área de cobertura um do outro, e se  $w(u, v) = 1$  então  $w(v, u) = 1$ .

A seguir o problema de selecionar o nó que minimiza as distâncias para todos os outros é modelado como um problema de grafos. O conjunto de caminhos possíveis entre quaisquer dois nós  $v_0$  e  $v_k$ , em que  $0 \leq k \leq n$ , é dado pela função  $\pi(v_0, v_k) = \{\langle v_0, v_1, \dots, v_k \rangle | \forall i \in \{1..k\} \Rightarrow (v_{i-1}, v_i) \in E\}$  e a distância em *hops* de um caminho  $p \in \pi(v_0, v_k)$  é dada por

$$d(p) = \sum_{i=1}^k w(v_{i-1}, v_i) \quad (1)$$

A partir da função acima,  $D(u, v)$  é definido como o custo do caminho mais curto entre dois nós  $u$  e  $v$  da seguinte forma  $D(u, v) = \min(\{d(p) \forall p \in \pi(u, v)\})$ . Como o grafo é não-direcionado, tem-se que  $D(u, v) = D(v, u)$ . A distância do caminho mais longo entre o nó  $u$  e todos os outros, a qual corresponde à distância do nó mais distante de  $u$ , é definida a partir do conjunto  $M_u = \{D(u, v) \forall v \in V\}$ .  $M_u$  contém as distâncias de  $u$  para cada nó em  $V$ . Desse modo, a distância máxima de um nó  $u$  é  $\max(M_u)$ . O nó raiz  $r$  é aquele que minimiza as distâncias máximas entre todos os nós. Isso significa que  $r$  é o nó que minimiza  $\{\max(M_u) \forall u \in V\}$ , logo o problema de encontrar o nó que está mais próximo de todos os outros consiste em descobrir o nó  $r$  que torna a seguinte igualdade verdadeira

$$\min(\max(M_r)) = \min_{\forall u \in V}(\max(M_u)) \quad (2)$$

#### 3.2. Algoritmo de Seleção de Nó Raiz Baseado em Grafo

O Algoritmo 1 foi proposto para resolver o problema delineado na Expressão 2. Ele é uma modificação da estratégia de programação dinâmica de Floyd-Warshall para resolver o problema de encontrar os caminhos mais curtos entre todos os pares de um grafo. O algoritmo usa esta estratégia para encontrar o conjunto  $M_u$  para cada nó  $u$ , calcular a distância dos nós mais distantes de  $u$  e identificar o nó que minimiza a distância para cada  $M_u$ . Sua entrada é o grafo da rede, bem como o número de vértices e a saída é o nó  $r$ .

As linhas de 1-9 correspondem à fase de inicialização. Na matriz  $A = A_{ij}^k$ ,  $k$  representa as iterações que executarão operações nessa matriz. A cada iteração,  $A_{ij}^k$  contém a distância mínima entre  $i$  e  $j$  em algum caminho que contém os nós de 1 até  $k$ . As linhas de 11-21 fazem três tarefas: computam o caminho mais curto entre cada par de nó, computam a distância máxima de cada nó para todos os outros e calculam a distância mínima das distâncias máximas, bem como o nó  $r$  que a possui. Nas linhas 17-18, o algoritmo verifica se é mais curto o caminho passando pelo nó  $k$  ( $A[i, k] + A[k, j] < A[i, j]$ ) e se sim atualiza essa distância em  $A[i, j]$ . Nas linhas 17-21, pra cada nó  $i$ , as variáveis  $M_i$  e  $minmax$  são atualizadas, de modo que  $minmax$  é usada para decidir na iteração  $k$ , qual

**Algoritmo 1** Algoritmo de Seleção de Nó Raiz Ótimo Baseado em Grafos

---

**Entrada:**  $G = (V, E)$  /\* Grafo da rede de sensores \*/  
 $n = |V|$  /\* Número de vértices\*/

**Saída:**  $r \in V$  /\* Nó raiz \*/  
1:  $A = [n, n]$  /\* Matriz auxiliar \*/  
2:  $minmax \leftarrow \infty$  /\* Resultado do problema minmax \*/  
3: **for**  $i \in \{1..n\}$  **do**  
4:   **for**  $j \in \{1..n\}$  **do**  
5:     **if**  $i = j$  **then**  
6:        $A[i, j] \leftarrow 0$   
7:     **else if**  $(i, j) \in E$  **then**  
8:        $A[i, j] \leftarrow (i, j)$   
9:     **else**  
10:        $A[i, j] \leftarrow \infty$   
11:   **for**  $k \in \{1..n\}$  **do**  
12:     **for**  $i \in \{1..n\}$  **do**  
13:        $M_i \leftarrow 0$   
14:       **for**  $j \in \{1..n\}$  **do**  
15:         **if**  $A[i, k] + A[k, j] < A[i, j]$  **then**  
16:            $A[i, j] \leftarrow A[i, k] + A[k, j]$   
17:         **if**  $A[i, j] > M_i$  **and**  $A[i, j] \neq \infty$  **then**  
18:            $M_i \leftarrow A[i, j]$   
19:       **if**  $M_i < minmax$  **then**  
20:          $minmax \leftarrow M_i$   
21:          $r \leftarrow i$

---

é o raiz  $r$  até aquele momento. Por causa dos três *loops* aninhados este algoritmo tem complexidade de  $O(n^3)$ .

### 3.3. Algoritmo Distribuído de Seleção de Nó Raiz

Em caso de falha do nó raiz seria necessária uma re-execução do algoritmo da Seção 3.2 e uma reconfiguração do nó raiz. Esses passos de re-execução do algoritmo e reconfiguração de propriedades da rede são também necessários em abordagens como [Li et al. 2014, Panigrahi and Khilar 2015, Su et al. 2016]. No entanto, é possível utilizar um algoritmo distribuído para realizar a seleção de nó raiz ótimo, como será descrito a seguir.

Segundo a definição do problema formulada na Seção 3.1, o nó raiz é aquele que minimiza as distâncias para todos os outros. Suponha que todos os nós da rede enviassem uma mensagem para este nó  $r$  e que ele guardasse a identificação do nó emissor da mensagem em um conjunto  $L_r$ . Depois de algum tempo  $r$  receberia  $n$  mensagens e  $L_r = V$ . Se todos os nós realizassem o mesmo, então depois de algum tempo cada nó teria um dos conjuntos  $L_{v_0} = L_{v_1} = \dots = L_{v_n} = V$ . No entanto, como  $r$  minimiza a distância em *hops* para os demais, então o conjunto  $L_r$  estaria completo, com  $n$  elementos antes dos outros. Nesse momento,  $r$  saberia que ele é o nó escolhido, então ele poderia enviar uma mensagem anunciando que é o raiz para a rede. A partir desta ideia, o Algoritmo 2 foi concebido para selecionar o nó raiz ótimo da rede. O código é executado em todos os nós da rede ( $\forall u \in V$ ).

Em RSSFs o canal de comunicação entre os nós não é confiável e o tempo de enviar e receber uma mensagem, por *hop*, varia em todos os *links*, logo é necessário tratar estes problemas. Para resolver o primeiro problema pode-se usar um protocolo de

**Algoritmo 2** Algoritmo Distribuído de Seleção de Nó Raiz Ótimo

---

**Entrada:**  $n = |V|$  /\* Número de nós \*/  
**Saída:**  $r$  /\* Nó raiz ótimo \*/  
**At Start**  
 $r \leftarrow \lambda$  /\* Nó raiz \*/  
 $L_u \leftarrow \emptyset$  /\* Lista de nós verificados \*/  
 $max_u \leftarrow 0$  /\* Distância do nó mais distante de  $u$  \*/  
 $minmax \leftarrow \infty$  /\* Mínimo das distâncias máximas \*/  
**if** DETECTADA FALHA DO NÓ RAIZ **then**  
  **Broadcast**  $MSG(SLCT, u, 1)$   
**Upon Receiving**  $MSG(SLCT, x, seq_x)$  **of**  $v$   
  **if**  $max_u = 0$  **then**  
    **Broadcast**  $MSG(SLCT, u, 1)$   
  **if**  $x \notin L_u$  **and**  $x \neq u$  **then**  
    **Broadcast**  $MSG(SLCT, x, seq_x)$   
     $L_u \leftarrow L_u \cup \{x\}$   
     $max_u \leftarrow \max(seq_x, max_u)$   
  **if**  $|L_u| = n - 1$  **then**  
    **if**  $max_u < minmax$  **or** ( $max_u = minmax$  **and**  $u < r$ ) **then**  
       $r \leftarrow u$   
       $minmax \leftarrow max_u$   
      **Broadcast**  $MSG(ROOT, u, minmax)$   
**Upon Receiving**  $MSG(ROOT, x, max_x)$  **of**  $v$   
  **if**  $max_x < minmax$  **or** ( $max_x = minmax$  **and**  $x < r$ ) **then**  
     $r \leftarrow x$   
     $minmax \leftarrow max_x$   
    **Broadcast**  $MSG(ROOT, x, minmax)$

---

entrega confiável para RSSFs, já para resolver o segundo problema, que é mais difícil, o Algoritmo 2 faz uso de dois tipos de mensagens, *SLCT* e *ROOT*, que servem para: (i) cada nó descobrir sua distância para cada outro da rede (*SLCT*), (ii) cada nó verificar se algum outro nó da rede ou ele próprio deve ser o raiz (*ROOT*). O item (ii) é uma verificação que resolve o segundo problema acima. A detecção de falha do nó raiz é realizada por meio da configuração de um *timer* em cada nó, de modo que se nenhuma mensagem de sincronização do raiz chegar após um determinado número de *timeouts* o nó considera que o raiz não está mais em operação.

## 4. Avaliação Experimental da Estratégia de Seleção de Nó Raiz

### 4.1. Configuração Experimental

Experimentos de simulação foram conduzidos para avaliar a estratégia de seleção de nó raiz contra a abordagem baseada em teoria espectral de grafos que otimiza a topologia da rede [Li et al. 2014]. Esta abordagem foi escolhida para comparação, pois ela tenta otimizar exatamente a distância em *hops* dos nós da rede para o nó de referência, o qual também é o principal objetivo da estratégia de seleção aqui proposta. Os dois parâmetros a avaliados foram:

- Distância em *hops* dos nós em relação ao raiz.
- Erro de sincronização do protocolo de sincronização.

O objetivo foi avaliar esses dois parâmetros e observar seu comportamento conforme se aumenta a escala da rede, tanto em cenários de redes esparsas, quanto para redes saturadas. O simulador utilizado foi o NS-3. O experimento consistiu em configurar uma área quadrada, posicionar os  $n$  nós aleatoriamente e executar o algoritmo de seleção de nó

raiz. O FTSP foi o protocolo utilizado tanto na abordagem de seleção de nó raiz quanto na abordagem de otimização da topologia [Li et al. 2014]. O resultado de cada experimento é a média da execução de 100 rodadas independentes. A Tabela 1 descreve todos os parâmetros da simulação e do protocolo de sincronização. Os experimentos foram conduzidos nos seguintes cenários, todos possuindo o FTSP como protocolo de sincronização subjacente:

- **BASELINE:** Apenas o algoritmo de sincronização FTSP é executado.
- **OPTSEL :** Estratégia de seleção de nó raiz ótima baseada em grafos descrita na Seção 3.
- **TOPOREFINER :** Estratégia de otimização de topologia da rede baseada em teoria espectral de grafos para redução da distância em *hops* dos nós da rede para o nó de referência [Li et al. 2014].
- **BOTH (OPT+TOPO) :** Realiza o refinamento de topologia (TOPOREFINER) e em seguida seleciona o nó raiz ótimo (OPTSEL).

| Parâmetro                      | Valor                                 |
|--------------------------------|---------------------------------------|
| Número de nós máximo           | 2000                                  |
| Área                           | 250m x 250m, 500m x 500m              |
| Período de amostragem          | 18s – 22s (Uniformemente distribuído) |
| Número de repetições           | 100                                   |
| Raio de alcance do rádio       | 30m                                   |
| Modelo de atraso de propagação | RANDOMPROPAGATIONDELAYMODEL           |
| Atraso de mensagem             | 0 – 100ms                             |

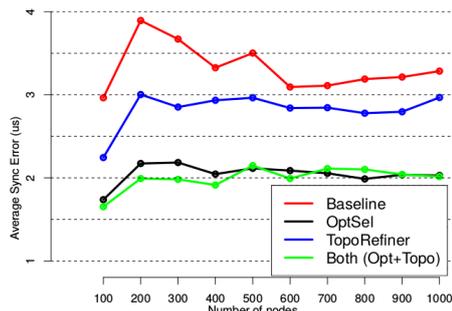
**Tabela 1. Parâmetros da simulação e do protocolo de sincronização.**

#### 4.2. Erro de Sincronização

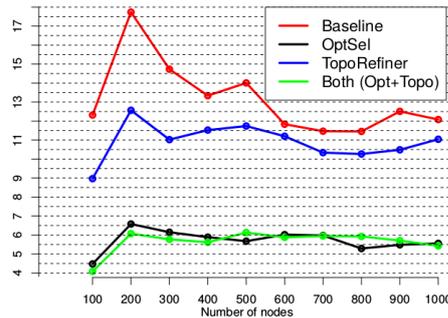
A Figura 1 e a Figura 2 apresentam os resultados de erro de sincronização em microssegundos em áreas de 250 x 250 m<sup>2</sup> e 500 x 500 m<sup>2</sup>. Na Figura 1, TOPOREFINER foi capaz de melhorar o desempenho médio do protocolo FTSP (BASELINE) em 14%, porém OPTSEL apresentou um resultado ainda melhor de 42%. Com relação ao erro máximo, TOPOREFINER manteve aproximadamente a mesma relação de melhoria, enquanto OPTSEL foi mais do que 2 vezes melhor do que o BASELINE. Na Figura 2, OPTSEL foi pelo menos 4 vezes melhor que o BASELINE e 3 vezes melhor do que TOPOREFINER. Considerando o erro máximo da rede (Figura 2(b)), OPTSEL supera o BASELINE e TOPOREFINER por pelo menos 4 e 5 vezes, respectivamente. Em todos os cenários BOTH (OPT+TOPO) apresentou ganho semelhante ao uso apenas de OPTSEL.

#### 4.3. Distância em *hops*

A fim verificar a hipótese de que a escolha adequada do nó raiz ótimo pode reduzir significativamente sua distância para todos os outros e, conseqüentemente, o erro de sincronização, a distância de todos os nós para o nó raiz foram computadas e são apresentadas na Figura 3 e na Figura 4. Nessas figuras, é possível observar uma correlação entre a distância dos nós para o raiz e o erro de sincronização, como esperado. Os resultados indicam que a estratégia de seleção é capaz de melhorar o erro de sincronização significativamente e é muito mais efetiva do que a abordagem que realiza uma otimização da topologia da rede.

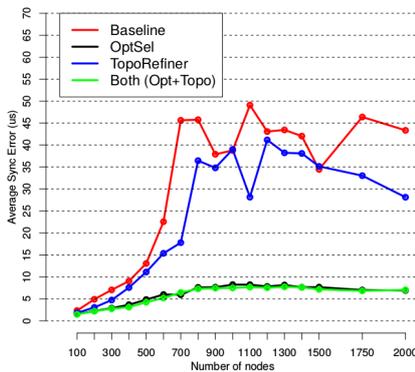


(a) Erro de Sincronização médio

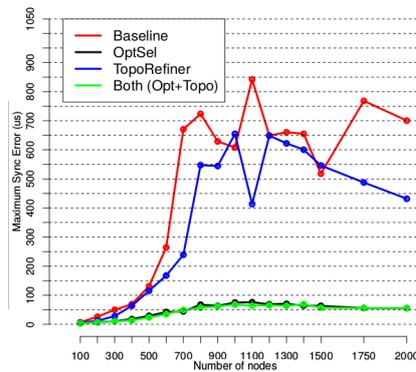


(b) Erro de Sincronização máximo

Figura 1. Erro de sincronização em uma área de 250 x 250 metros.

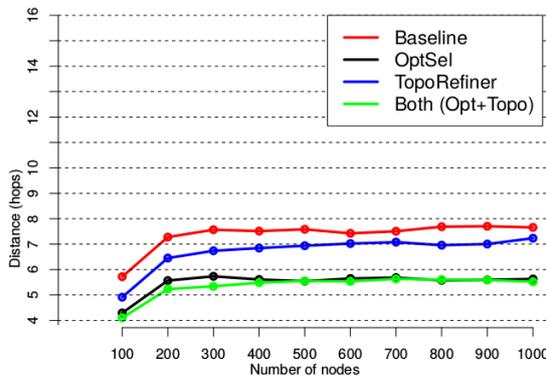


(a) Erro de Sincronização médio

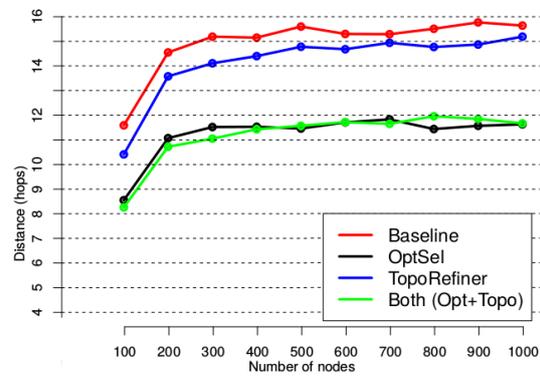


(b) Erro de Sincronização máximo

Figura 2. Erro de sincronização em uma área de 500 x 500 metros.



(a) Distância média



(b) Distância máxima

Figura 3. Distância em hops para o nó raiz.

#### 4.4. Tolerância à falha

Para demonstrar a efetividade do algoritmo distribuído descrito na Seção 3.3 em encontrar o nó raiz ótimo quando este falha, foi realizado um experimento de mais de 10 horas com 1000 nós, cujos resultados são apresentados na Figura 5. O experimento consistiu em: inicialmente executar o algoritmo de seleção baseado em grafos e desligar o nó raiz no instante 3000 segundos (linha vertical da figura). O objetivo foi verificar se a seleção do novo nó raiz ótimo faz com que o erro de sincronização permaneça nos mesmos níveis iniciais dados pelo algoritmo baseado em grafos. Para tal, dois cenários foram confrontados, um com a estratégia de seleção distribuída (OPTSELDIST) e outro sem ela (NOOPTSELDIST) - estratégia do FTSP de escolher o nó com menor ID [Maróti et al. 2004]. Na Figura 3.3, considerando o cenário OPTSELDIST, após a falha em 3000s o erro de sincronização per-

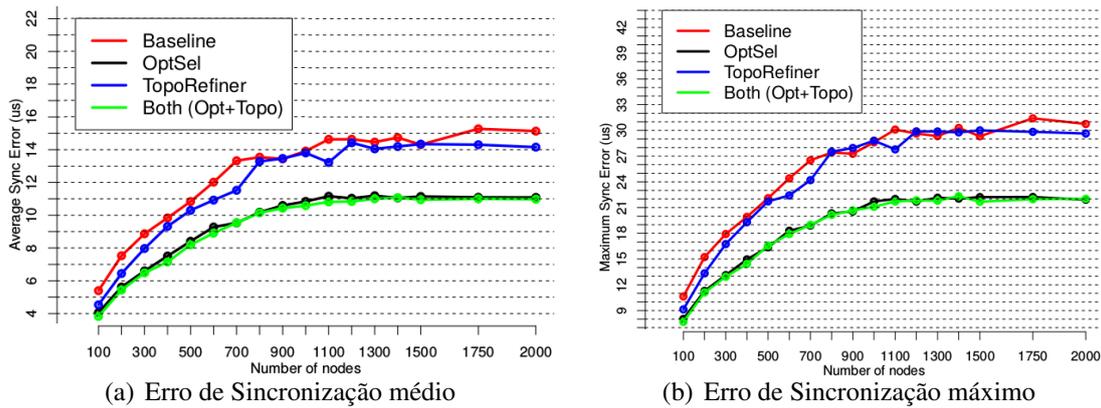


Figura 4. Distância em *hops* para o nó raiz.

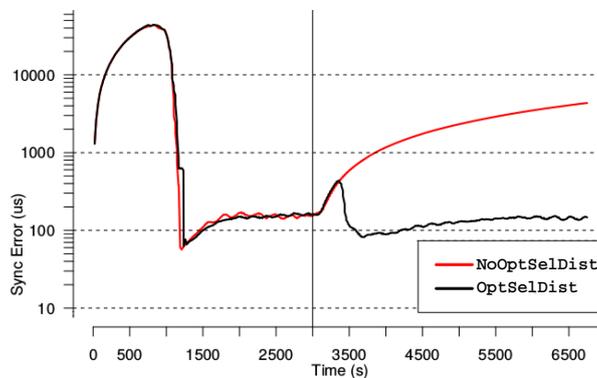


Figura 5. Erro de sincronização com desligamento do nó raiz aos 3000 segundos.

manece aproximadamente constante; sem a seleção de nó raiz ótimo, o erro aumenta. No início (entre 0 e cerca de 1000 s) o alto erro de sincronização se deve aos relógios iniciarem com valores aleatórios e à fase inicial de calibração do protocolo de sincronização subjacente FTSP.

## 5. Conclusões

Este trabalho apresentou duas propostas de seleção de nó raiz ótimo para reduzir a distância em *hops* dos nós para o raiz. Por meio de resultados experimentais é possível concluir que a abordagem baseada em grafos supera em cerca de seis vezes uma abordagem baseada em otimização topológica e a abordagem distribuída permite uma recuperação eficiente do nó raiz em caso de falha.

## Referências

- Benzaïd, C., Bagaï, M., and Younis, M. (2017). Efficient clock synchronization for clustered wireless sensor networks. *Ad Hoc Networks*, 56:13 – 27.
- Cardoso, C. B., Guidoni, D. L., Kimura, B. Y., and Villas, L. A. (2017). A hybrid solution for 3d location and time synchronization in wsn. In *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, MobiWac '17*, pages 105–112, New York, NY, USA. ACM.
- Huang, G., Zomaya, A. Y., Delicato, F. C., and Pires, P. F. (2014). Long term and large scale time synchronization in wireless sensor networks. *Computer Communications*, 37(Supplement C):77 – 91.

- Kamilaris, A. and Pitsillides, A. (2016). Mobile phone computing and the internet of things: A survey. *IEEE Internet of Things Journal*, 3(6):885–898.
- Lenzen, C., Locher, T., and Wattenhofer, R. (2008). Clock synchronization with bounded global and local skew. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 509–518.
- Lenzen, C., Sommer, P., and Wattenhofer, R. (2015). Pulsesync: An efficient and scalable clock synchronization protocol. *IEEE/ACM Trans. Netw.*, 23(3):717–727.
- Li, X., Ma, Q., Sun, W., Liu, K., and Liu, Y. (2014). Topology shaping for time synchronization in wireless sensor networks. In *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 33–40.
- Maróti, M., Kusy, B., Simon, G., and Lédeczi, A. (2004). The flooding time synchronization protocol. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems, SenSys '04*, pages 39–49, New York, NY, USA. ACM.
- Panigrahi, N. and Khilar, P. M. (2015). An evolutionary based topological optimization strategy for consensus based clock synchronization protocols in wireless sensor network. *Swarm and Evolutionary Computation*, 22:66–85.
- Su, X., Hui, B., and Chang, K. (2016). Multi-hop clock synchronization based on robust reference node selection for ship ad-hoc network. *Journal of Communications and Networks*, 18(1):65–74.
- Upadhyay, D., Dubey, A. K., and Thilagam, P. S. (2018). Application of non-linear gaussian regression-based adaptive clock synchronization technique for wireless sensor network in agriculture. *IEEE Sensors Journal*, 18(10):4328–4335.
- Yıldırım, K. S., Carli, R., and Schenato, L. (2018). Adaptive proportional integral clock synchronization in wireless sensor networks. *IEEE Transactions on Control Systems Technology*, 26(2):610–623.

## **SININHO: um serviço de mapeamento espaço-temporal para melhoria da tomada de decisão na gestão em saúde**

**Lindemberg Nascimento<sup>1</sup>, Renato Alves<sup>1</sup>, Daniel Victor Saraiva<sup>1</sup>,  
Reinaldo Braga<sup>1</sup>, Carina Oliveira<sup>1</sup>**

<sup>1</sup> Laboratório de Redes de Computadores e Sistemas (LAR)  
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)  
Caixa Postal 62.800-000 – Aracati – CE – Brazil

**Abstract.** *Information and communication technologies have been providing innovative health solutions over the years. In this context, this paper presents a spatiotemporal mapping service for e-Health and m-Health systems for health-related data management. The proposal, called SININHO, was incorporated into a real health system for the diagnosis, treatment and follow-up of childhood and juvenile cancer. By optimizing spatiotemporal indicators, SININHO generates multiple strategic views of marker maps, density maps and heat maps. In this way, health managers will be able to analyze the presented scenarios and to base their decisions.*

**Resumo.** *As tecnologias da informação e comunicação vêm proporcionando soluções inovadoras em saúde ao longo dos anos. Neste contexto, este trabalho apresenta um serviço de mapeamento espaço-temporal para sistemas e-Health e m-Health para gestão de dados relacionados à saúde. A proposta, denominada SININHO, foi incorporada em um sistema de saúde real de diagnóstico, tratamento e acompanhamento do câncer infanto-juvenil. Por meio da otimização de indicadores espaço-temporais, o SININHO gera múltiplas visualizações estratégicas de mapas de marcadores, mapas de densidade e mapas de calor (heat map). Dessa forma, gestores de saúde podem analisar os cenários apresentados e fundamentar as suas tomadas de decisões.*

### **1. Introdução**

O câncer em crianças e adolescentes não é uma única entidade patológica, mas um espectro de diferentes neoplasias malignas, que variam segundo tipo histológico, localização de origem da doença, sexo, idade e raça. A mortalidade por câncer em crianças e adolescentes em países desenvolvidos é considerada a segunda causa de morte na infância (crianças de 1 a 14 anos), correspondendo a cerca de 4% a 5% dos óbitos. Entretanto, nos países em desenvolvimento, essa proporção é bem menor, cerca de 1%<sup>1</sup>. Nos adolescentes e adultos jovens (de 15 a 29 anos), o câncer é a principal causa de morte nos países desenvolvidos [INCA 2016].

No Brasil, os óbitos por câncer entre crianças, adolescentes e adultos jovens correspondem à segunda causa de morte. O câncer em crianças e adolescentes (de 0 a 19 anos) é considerado raro quando comparado com o câncer em adultos, correspondendo

---

<sup>1</sup>As principais causas de óbito na faixa etária de 1 a 14 anos ocorrem em razão das mortes por doenças infecciosas [INCA 2016].

entre 2% e 3% de todos os tumores malignos registrados no país com taxas de mortalidade que variam entre 42,33 e 49,17 por milhão. A faixa etária que apresenta o maior risco de morte no país está no grupo de 15 a 19 anos (com 54,01 por milhão), especialmente para o sexo masculino (64,57 por milhão). Por outro lado, a faixa etária de menor risco está entre 5 a 9 anos. Além disso, em todas as faixas etárias, existe um predomínio nas taxas do sexo masculino, exceto para os menores de 1 ano [INCA 2016].

Para tentar reduzir esses índices de mortalidade, informação com qualidade é um recurso fundamental para chegar ao mais próximo da realidade, traçar suas características, detectar problemas e agir para solucioná-los. Muitos países em desenvolvimento adotaram sistemas eletrônicos de informação de gerenciamento de saúde para apoiar a gestão na prestação de serviços [Chikumba and Ramussen 2016]. Dessa forma, é essencial que a informação de saúde não seja apenas coletada, mas também analisada e disseminada entre as partes interessadas. Neste cenário, a informação, surge como o alicerce para o desenvolvimento do conhecimento tecnológico e científico para a inovação em saúde, sendo fundamental para guiar a gestão de serviços de saúde com eficiência, eficácia e efetividade.

Kanth *et al.* [Kanth et al. 2014] enfatizam que, nos últimos anos, as tecnologias da informação e comunicação têm desempenhado um papel vital em quase todos os setores do desenvolvimento humano, incluindo saúde, educação, transporte e desenvolvimento de infraestrutura. Os autores destacam que alguns dos impactos das tecnologias em sistemas de saúde atualmente têm os seguintes objetivos: melhorar a qualidade e a eficácia da saúde; ampliar as comunicações em tempo real entre os profissionais de saúde; aumentar a eficiência administrativa e dos processos de trabalho e diminuir a documentação e o tempo de trabalho improdutivo ou ocioso.

Visto que as informações são de extrema relevância para a tomada de decisões, o presente trabalho apresenta o SININHO, um sistema capaz de otimizar indicadores espaço-temporais de sistemas *e-Health* e *m-Health* para gestão de dados. Como estudo de caso, o SININHO foi incorporado ao SISAPP [Oliveira et al. 2017], um Sistema da Associação Peter Pan (APP) [APP 2018], entidade sem fins lucrativos que atende aproximadamente 2.200 pacientes, atuando na área da saúde voltada para o diagnóstico, tratamento e acompanhamento do câncer infanto-juvenil.

Por meio da otimização de indicadores espaço-temporais, o sistema SININHO gera múltiplas visualizações estratégicas de mapas de marcadores, mapas de densidade e mapas de calor (*heat maps*), possibilitando uma maior flexibilidade e visualização mais eficiente dos dados em comparação a utilização de formulários ou a consolidação de dados em tabelas. Desse modo, a principal questão que norteia esse sistema é a relevância dos mapas não apenas com uma estratégia de apresentação dos resultados, mas facilitando o estudo de correlações espaciais entre condições sociais, ambientais e a saúde, fazendo com que os profissionais de saúde tenham um maior direcionamento na tomada de decisões no planejamento, monitoramento e avaliação das ações de saúde.

O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta um referencial teórico destacando alguns trabalhos relacionados a ambientes de Sistemas de Informação Geográfica (SIGs); a Seção 3 detalha as fontes de informações do SININHO, desde a coleta de dados até sua integração com os mapas; na Seção 4 são ap-

resentados os serviços de mapeamento espaço-temporais; e, por fim, na Seção 5, são destacadas as principais contribuições deste estudo e apresentadas as propostas futuras para o aperfeiçoamento e continuidade do mesmo.

## 2. Trabalhos Relacionados

Sistemas de informação em saúde são compreendidos como um conjunto de componentes integrados que atuam com o objetivo de obter dados e transformá-los em informações úteis para que os problemas de saúde de uma população possam ser identificados e consequentemente resolvidos. Nesse contexto, os trabalhos a seguir apresentam um referencial teórico de algumas pesquisas relacionadas aos ambientes de Sistemas de Informação Geográfica (SIGs) no mesmo contexto desta proposta.

No trabalho de [Freifeld et al. 2008], os autores apresentam um mapeamento global de alertas de doenças. O sistema desenvolvido por eles tem por objetivo aumentar a vigilância de doenças infecciosas por meio da integração de um conjunto de dados, provenientes da Organização Mundial da Saúde (OMS), ProMED-mail e de outras fontes disponíveis na Internet. Através do cruzamento desses dados, a aplicação obtém uma maior utilidade, já que analisá-los em conjunto é mais eficiente do que analisá-los individualmente. Por meio de um sistema automatizado de processamento de texto, os dados são separados por doença e exibidos por localidade, fornecendo, assim, um ponto de partida para a informação em tempo real sobre doenças infecciosas emergentes.

Um sistema de vigilância em tempo real de doenças transmitidas pela água, na cidade de Ahmedabad, na Índia, é apresentado em [Hughes et al. 2014]. O sistema proposto utiliza várias tecnologias, desde a coleta dos dados, casos de doenças geocodificadas, utilizando *tablets* Android, até sua análise, usando o Pysal. Esta última tecnologia é um framework baseado em Python que realiza uma segmentação de bases de dados em camadas *Keyhole Markup Language* (KML), cujos resultados são armazenados na nuvem para auxiliar a visualização de mapas, por meio do *OpenStreetMap* (OSM) [OSM 2018].

No trabalho [Fei et al. 2016], os autores propõem uma relação espaço-temporal dos casos de câncer de tireoide e de mama na cidade de Hangzhou, na China. Eles utilizam o critério integrativo de previsibilidade da doença para observar a relação entre os dois tipos de câncer nas bases de conhecimento que mapeiam a doença. Após isso, são geradas previsões de incidência espaço-temporais de ambas doenças em dada região. Assim sendo, é realizada uma avaliação da variação espaço-temporal da incidência da doença para melhorar a compreensão das associações entre câncer de tireoide e de mama no nível populacional por meio da combinação da teoria da Entropia Máxima Bayesiana.

Bezerra *et al.* [Bezerra et al. 2017] desenvolveram um sistema denominado DENGOSA, um sistema *Web* de Informação Geográfica na área da epidemiologia capaz de fornecer serviços para controle de epidemias. O sistema complementa o ciclo epidemiológico fornecendo uma alimentação de qualidade à base do sistema federal por meio da integração das funcionalidades de coleta informatizadas do DENGOSA, suprimindo a burocracia no processo de coleta de dados em fichas de papel à nível municipal. A fim de oferecer uma apresentação amigável dos dados coletados para as análises epidemiológicas no DENGOSA, foi projetado o uso de mapas *Web* com a geovisualização da incidência da epidemia.

Diante dos trabalhos citados, nota-se que os SIGs são excelentes ferramentas

para combinar dados de diferentes fontes e apresentar informações de apoio à tomada de decisões. No entanto, o trabalho aqui proposto possui como diferencial em relação aos citados a geração de múltiplas visualizações estratégicas de mapas, como mapa de marcadores, mapas de densidade e mapas de calor (*heat maps*), identificando padrões, relações e anomalias de difícil detecção quando os dados são analisados de forma independente.

### 3. Fluxo de Dados do Estudo de Caso

Nesta seção, são apresentados os componentes e interações do SININHO, desde a coleta de dados até a geovisualização necessária para a geração dos mapas.

#### 3.1. Coleta de Dados

A avaliação da incidência, da mortalidade e da morbidade hospitalar é importante para o delineamento do perfil de câncer e a efetiva vigilância para transformação de informações em ações efetivas para o controle do câncer. A coleta de dados aplicada nesse estudo é proveniente de duas bases de dados.

A primeira base provém do SISAPP [Oliveira et al. 2017], sistema web utilizado pela Associação Peter Pan (APP), ao qual o SININHO encontra-se incorporado. A base do SISAPP contém dados de incidência, tratamento e mortalidade do câncer infanto-juvenil, juntamente com sintomas, histórico familiar de câncer, tempo de diagnóstico (precoce, intermediário ou tardio) entre outras. No caso da base de dados do SISAPP, um paciente pode ser associado a três pontos geográficos distintos: município onde nasceu, município de procedência (local onde o paciente residia quando a doença foi diagnosticada) e o endereço atual (local onde o paciente reside durante o tratamento).

Além disso, o sistema SISAPP apresenta em seu banco de dados uma modelagem que proporciona a interação com informações geográficas onde os municípios cadastrados são referenciados com dados de latitude e longitude. Assim, no ato do cadastro da cidade natal de um paciente é realizada uma busca via *Web Service* das coordenadas correspondentes àquela cidade, as quais são utilizadas para o preenchimento dos mapas.

As taxas de mortalidade representam um importante subsídio para análise dos indicadores de saúde, mesmo nos países desenvolvidos e industrializados. Por meio da análise dessas taxas é possível medir o progresso na luta contra o câncer, bem como considerá-la um indicador do acesso ao tratamento da saúde no país [INCA 2008]. As informações sobre mortalidade, obtidas por meio do recolhimento sistemático das informações dos atestados de óbito, são as mais simples e acessíveis para o estudo das condições de saúde de uma população [INCA 2016].

Dessa forma, a segunda base provém das informações sobre mortalidade divulgadas pelo Sistema de Informação sobre Mortalidade (SIM), do Ministério da Saúde. A base do SIM engloba informações sobre todas as mortes ocorridas em território nacional, entre elas, causa da morte, local e data do óbito, cidade natal do indivíduo. Vale destacar que, pelo fato da Associação Peter Pan direcionar seus esforços à causa do câncer infanto-juvenil, os dados do SIM que estão sendo utilizados no SININHO se referem apenas às crianças e adolescentes com idades entre 0 e 19 anos que foram acometidas com câncer.

### 3.2. Extração e Transformação dos Dados

O principal objetivo da extração é adquirir os dados necessários e prepará-los para gerar uma base de dados separada que proporcione uma análise de acordo com as principais necessidades da gestão. A extração auxilia no descarte de inconsistências da base de dados, tratando os valores nulos e eliminando registros irrelevantes.

Após a extração, os dados passam por um processo de transformação em um formato aberto como serviço. Para a troca dos dados entre o sistema e o serviço, é utilizado o formato *JavaScript Object Notation* (JSON), que tem capacidade de estruturar informações de forma mais compacta e legível, tornando, assim, mais fácil a análise de tais dados.

### 3.3. Geovisualização

Após concluída a coleta, extração e transformação, o próximo passo é a geovisualização. Nessa etapa, foi utilizada a *Leaflet*, uma biblioteca *JavaScript* de código aberto, que permite a construção de mapas interativos em projetos *Web* de maneira simples, com alta performance e usabilidade.

Através do framework *OpenStreetMap* (OSM) – um software livre de código aberto [OSM 2018], a *Leaflet* tem acesso à uma coleção de imagens cartográficas de mapas, baseadas em dados *Raster*. Com os dados do OSM, um usuário está livre para fazer o *download* de qualquer mapa que deseje, permitindo o uso dos dados *off-line*. Esse processo de geovisualização tem como principal objetivo auxiliar o aprimoramento da assistência ao paciente, a pesquisa de base clínica e também prover subsídios às atividades administrativas.

## 4. Sistema SININHO

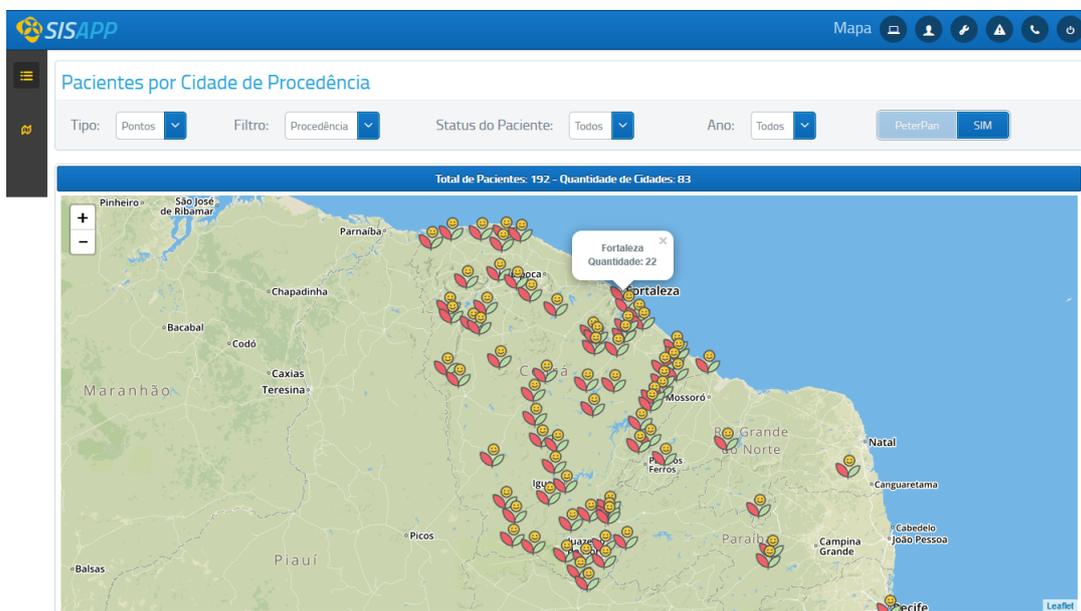
Esta seção aborda detalhes do SININHO, um serviço baseado em localização (*Location-Based Services* - LBS) capaz de gerar múltiplas visualizações estratégicas para o apoio à tomada de decisão na saúde através de mecanismos de geovisualização. No estudo de caso proposto, o SININHO aparece como uma nova camada de serviços, com o objetivo de gerar informações enriquecidas a partir de dados provenientes do SISAPP e dados do Sistema de Informações sobre Mortalidade (SIM). O serviço apresenta três visualizações distintas de mapas: (i) mapa de marcadores; (ii) mapa de densidade e (iii) mapa de calor. Em cada uma das visualizações há uma série de filtros que possibilitam aos gestores e profissionais de saúde uma visão a nível macro e micro de determinadas situações.

### 4.1. Mapa de Marcadores

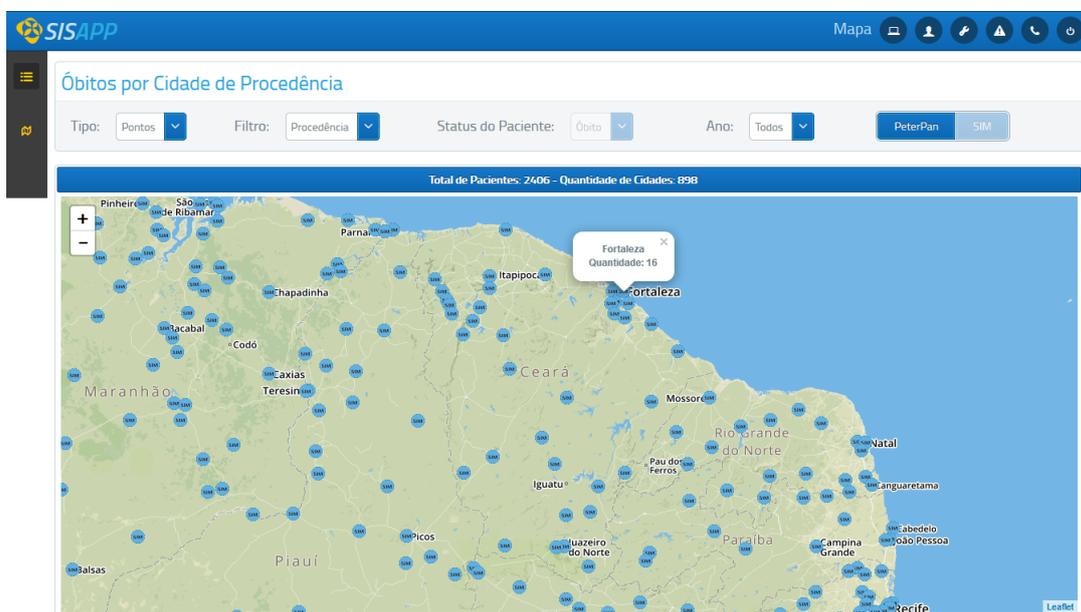
A Figura 1 apresenta um exemplo de Mapa de Marcadores (MM) (com a logomarca da APP). O mapa apresenta os municípios com pelo menos um paciente atendido pela associação de acordo com o filtro aplicado, onde cada marcador corresponde ao total de pacientes diagnosticados com câncer em uma determinada cidade<sup>2</sup>. Já a Figura 2 apresenta marcadores de dados reais de mortalidade por câncer com foco em crianças e

<sup>2</sup>Para não comprometer a privacidade da instituição e de seus pacientes, os dados aqui apresentados em relação à quantidade de pacientes e municípios não são reais. Vale destacar também que a Associação Peter Pan está localizada no Ceará, ou seja, a maior quantidade de seus pacientes é de procedência da região nordeste, principalmente, do estado do Ceará. Por isso, as imagens apresentadas destacam essa região.

adolescentes com idades entre 0 a 19 anos em 2014 que provêm das informações sobre mortalidade divulgadas no SIM em municípios com pelo menos um óbito.



**Figure 1. Exemplo de Mapa de Marcadores com pacientes por cidade de procedência com dados do SISAPP.**



**Figure 2. Exemplo de Mapa de Marcadores com óbitos por cidade de procedência com dados do SIM.**

Conforme ilustrado no topo das figuras, os filtros que podem ser aplicados são referentes aos seguintes indicadores: procedência dos pacientes (onde mora ou morava); naturalidade dos pacientes (cidade natal); pacientes que foram a óbito e municípios que receberam treinamento. Em relação ao Status dos Pacientes, as escolhas possíveis são: em análise; em tratamento; curado, óbito, abandono, dentre outras. As demais opções

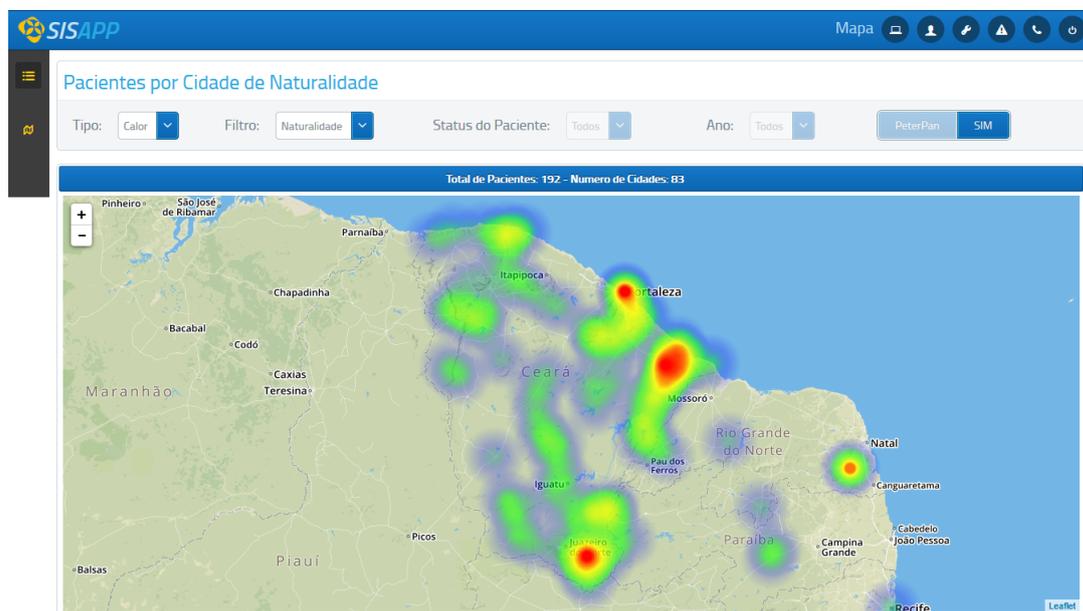
estão relacionadas a escolha do ano a ser pesquisado e a possibilidade de alternar entre as bases de dados da associação e do SIM.

Em vista disso, os gestores e profissionais de saúde podem facilmente buscar, por exemplo, a quantidade de crianças que vieram a óbito em um determinado ano, procedentes e/ou naturais de determinada cidade.

## 4.2. Mapa de Calor

A visualização de um Mapa de Calor (MC) consiste em um cálculo de densidade de pontos em dada região e, em sequência, a aplicação de uma cor (a partir de uma gama de cores predefinidas para esta região) [Meier and Heidmann 2014]. Tal visualização permite um entendimento rápido da situação e do impacto causado por ela, mesmo com um grande número de elementos apresentados de uma só vez. Este tipo de mapa faz parte de um *plug-in* da API *Leaflet* e funciona como uma camada sobreposta ao mapa padrão.

A Figura 3 apresenta, o que seria no Mapa de Calor, a naturalidade dos pacientes atendidos na associação. Além disso, a Figura 4 apresenta a visualização de parte da região Nordeste do Brasil em 2014. Para essa figura, são utilizados dados reais de mortalidade por câncer com foco em crianças e adolescentes da base de dados do SIM, que neste caso, também leva-se em consideração os endereços de naturalidade dos indivíduos.



**Figure 3. Exemplo de Mapa de Calor com pacientes atendidos pela associação.**

Nota-se que o MC apresenta uma abordagem bem diferente da ilustrada no mapa de marcadores. No MC é possível perceber um padrão, onde há um maior número de ocorrências nas capitais e regiões metropolitanas dos estados mostrados. A partir desses padrões é possível, por exemplo, direcionar treinamentos e campanhas publicitárias, além de estudos epidemiológicos para analisar as correlações e causas de incidência nos municípios analisados.

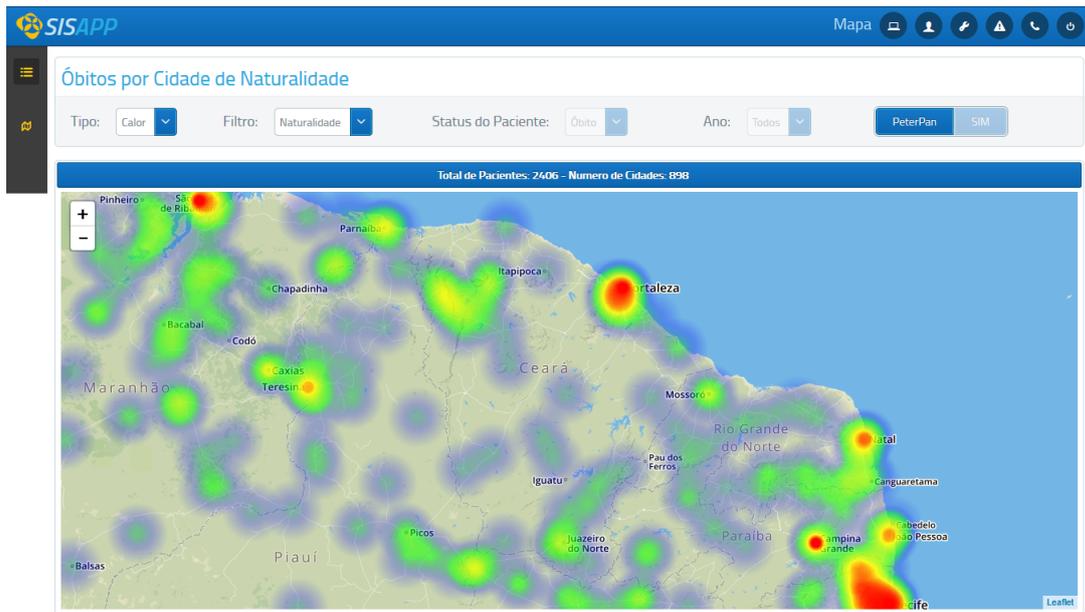


Figure 4. Exemplo de Mapa de Calor com Óbitos por Naturalidade na Região Nordeste do Brasil.

### 4.3. Mapa de Densidade

O serviço de Mapa de Densidade (MD) disponibiliza uma visualização mais refinada de cada município. Neste caso, utiliza-se a demarcação estabelecida pelo Instituto Brasileiro de Geografia e Estatística (IBGE).

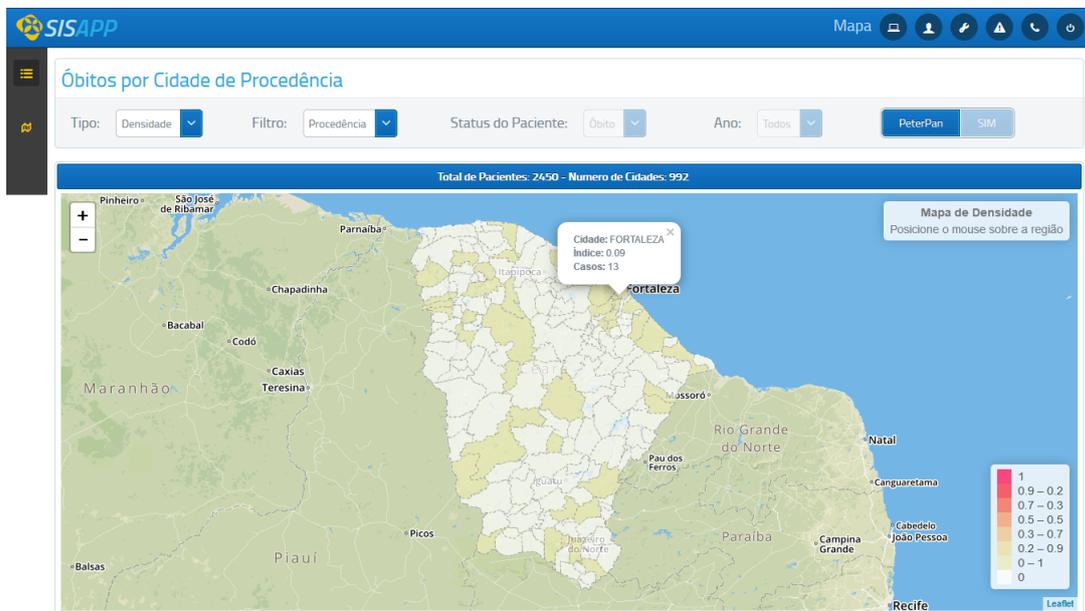


Figure 5. Exemplo de Mapa de Densidade com óbitos por procedência no Estado do Ceará.

A Figura 5 apresenta um exemplo de mapa de densidade utilizando os dados do Sistema de Informações sobre Mortalidade, que, assim como no mapa de calor referente aos dados do SIM, são reais. Nesse caso, são apresentados os óbitos por cidade de

procedência, no estado do Ceará em 2014. Em destaque, podemos observar uma caixa de texto mostrando a cidade selecionada (Fortaleza), a quantidade de casos ocorridos no local (13 casos) e o índice percentual das ocorrências daquele município (0,09) em relação ao total de casos no país.

O serviço do mapa de densidade permite uma visão mais aprofundada, já que delimita geograficamente cada cidade e possibilita, no caso dos dados do SIM, a identificação de áreas com maior ocorrência de óbitos onde as regiões com cores mais escuras apresentam maiores índices de óbitos e as regiões mais claras, conseqüentemente, apresentam menores índices.

## 5. Conclusões

Nos últimos anos, os principais sistemas informativos vêm sendo incorporados, majoritariamente, às tecnologias da informação e comunicação. Neste contexto, é importante que a área da saúde passe por um processo de modernização de suas metodologias, abrindo espaço para que as tecnologias se tornem, efetivamente, sua grande aliada no processo de gestão. Dentre os benefícios, em relação aos métodos que não utilizam a informática, podemos destacar maior agilidade, interatividade e facilidade de visualização dos dados. Com a utilização de formulários de papel, por exemplo, a gestão não tem como manusear grandes quantidades de dados de forma ágil e fácil a ponto de convertê-los em informações concisas e coesas.

Entretanto, por meio de interfaces gráficas, obtém-se uma visualização mais efetiva dos dados, além de uma maior flexibilidade. Em vista disso, ao serem agregados à gestão de saúde, por intermédio do SININHO, os três modos de mapeamento propostos geram uma melhora técnico-administrativa considerável na gestão de dados sobre saúde. O serviço permite aos gestores e profissionais da área realizar consultas de forma intuitiva e amigável por intermédio de uma interface compreensível. Desta maneira, a gestão consegue identificar, de maneira ágil, aspectos que podem subsidiar programas de prevenção e diagnóstico precoce do câncer infanto-juvenil, por exemplo.

É importante destacar que o SININHO pode ser utilizado de forma independente ou anexo a sistemas de *e-Health* ou *m-Health* (como é o caso do SISAPP). Por isso, através do serviço de mapeamento espaço-temporal aqui proposto, é possível atender, de imediato, a demanda de outras entidades de saúde devido sua grande capacidade de adaptação uma vez que a automatização do processo de geração de mapas estratégicos é claramente vantajosa, em especial quando a disponibilidade da informação para a tomada de decisão afeta a qualidade de vida de pessoas com câncer ou de outras doenças como diabetes, AIDS, cólera, dengue, entre outras principais do Brasil.

Como trabalhos futuros para o SININHO, almeja-se a adição de novos filtros, permitindo uma visão mais detalhada de cenários críticos em saúde e a integração com *Machine Learning*, utilizando Aprendizado Não Supervisionado, visando a Clusterização dos dados. Isso vai ajudar a gestão no processo de identificação de padrões. Por exemplo, é possível identificar que determinado tipo de doença é mais recorrente em crianças em uma determinada faixa de idade que já viveram em uma dada região. Com a agregação dessas novas ferramentas ao SININHO, espera-se que a gestão de saúde obtenha uma eficiência ainda maior em suas atividades e que, assim, eleve o índice de cura e melhore a qualidade de vida de crianças e adolescentes com câncer e de suas famílias.

## References

- APP (2018). Associação Peter Pan (APP). <http://www.app.org.br>. [Último Acesso em: 15-Março-2018].
- Bezerra, A., Bento Filho, J. J., Braga, R. B., Oliveira, C. T., and Oliveira, A. M. B. (2017). Dengosa: Um sistema de informação geográfica para apoio à decisão no controle de epidemias. In *XVI Workshop de Ferramentas e Aplicações (WFA)/Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 179–183.
- Chikumba, P. A. and Ramussen, S. L. (2016). Management and use of health information in malawi and burkina faso: The role of technology. In *IST-Africa Week Conference*, pages 1–9.
- Fei, X., Christakos, G., Lou, Z., Lou, Z., Liu, Q., and Wu, J. (2016). Spatiotemporal co-existence of female thyroid and breast cancers in hangzhou. *Scientific Reports*, 6.
- Freifeld, C. C., Mandl, K. D., Reis, B. Y., and Brownstein, J. S. (2008). Healthmap: Global infectious disease monitoring through automated classification and visualization of internet media reports. *Journal of the American Medical Informatics Association*, 15(2):150–157.
- Hughes, C., Naik, V. S., Sengupta, R., and Saxena, D. (2014). Geovisualization for Cluster Detection of Hepatitis A and E Outbreaks in Ahmedabad, Gujarat, India. In *ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health*, HealthGIS, pages 39–44, New York, NY, USA. ACM.
- INCA (2008). *Câncer da criança e adolescente no Brasil: dados dos registros de base populacional e de mortalidade*. Rio de Janeiro: INCA.
- INCA (2016). *Incidência, mortalidade e morbidade hospitalar por câncer em crianças, adolescentes e adultos jovens no Brasil: informações dos registros de câncer e do sistema de mortalidade*. Rio de Janeiro: INCA.
- Kanth, R. K., Liljeberg, P., Westerlund, T., Kumar, H., Tenhunen, H., Wan, Q., Yang, G., and Zheng, L. R. (2014). Information and communication system technology’s impacts on personalized and pervasive healthcare: A technological survey. In *IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, pages 1–5.
- Meier, S. and Heidmann, F. (2014). Too many markers, revisited: An empirical analysis of web-based methods for overcoming the problem of too many markers in zoomable mapping applications. In *International Conference on Computational Science and Its Applications*, pages 121–125.
- Oliveira, R. A., Nascimento Filho, L. B., Vieira, W. O., Bandeira, T. F. L., Oliveira, A. M. B., Braga, R. B., and Oliveira, C. T. (2017). Sistema de apoio à tomada de decisão na gestão de atendimento a pacientes com câncer infanto-juvenil. In *XVI Workshop de Ferramentas e Aplicações (WFA)/Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 189–193.
- OSM (2018). OpenStreetMap. <http://www.openstreetmap.org/>. [Último Acesso em: 15-Março-2018].

# Uma Abordagem não Intrusiva para Reconhecimento de Atividades em Casas Inteligentes Explorando Processamento Semântico\*

Eduardo Abreu<sup>1</sup>, Gerson Cavalheiro<sup>1</sup>, Ana Pernas<sup>1</sup>, Adenauer Yamin<sup>1</sup>,  
João Lopes<sup>2</sup>, Cláudio Geyer<sup>3</sup>

<sup>1</sup>Universidade Federal de Pelotas (UFPEL) – Pelotas – RS – Brasil

<sup>2</sup>Instituto Federal Sul-rio-grandense (IFSUL) – Pelotas – RS – Brasil

<sup>3</sup>Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre – RS – Brasil

{eabreu, gerson.cavalheiro, marilza, adenauer}@inf.ufpel.edu.br  
joaolopes@cavg.ifsul.edu.br  
geyer@inf.ufrgs.br

**Abstract.** *The purpose of this work, named EXEHDA-AR (Execution Environment for Highly Distributed Applications-Activity Recognition), is to provide architectural features that allow the recognition of activities in the EXEHDA middleware, exploiting a semantic processing approach. For this purpose, architectural components were designed and integrated in the Subsystem of Adaptation and Context Recognition of EXEHDA. A case study about smart homes was developed to evaluate the functionalities proposed for the EXEHDA-AR, obtaining an average accuracy of 94.36% in the recognition of activities. These results indicate that methods based on semantic processing constitute a viable alternative, with low level of intrusion.*

**Resumo.** *O objetivo deste trabalho, denominado EXEHDA-AR (Execution Environment for Highly Distributed Applications-Activity Recognition), é prover recursos arquiteturais que permitam o reconhecimento de atividades no middleware EXEHDA, explorando uma abordagem baseada em processamento semântico. Para tanto foram concebidos componentes arquiteturais, os quais foram integrados ao Subsistema de Reconhecimento de Contexto e Adaptação do EXEHDA. Um estudo de caso sobre casas inteligentes foi desenvolvido para avaliar as funcionalidades propostas para o EXEHDA-AR, sendo obtida uma acurácia média de 94,36% no reconhecimento de atividades. Estes resultados apontam que métodos baseados em processamento semântico constituem uma alternativa viável, com baixo nível de intrusão.*

## 1. Introdução

Uma das aplicações do reconhecimento de atividades está no domínio ubíquo da *e-Health*, o qual pode ser explorado em sistemas de reabilitação, gerenciamento de doenças crônicas e monitoramento de idosos. Esse interesse na área de reconhecimento de atividades tem tido como motivação o envelhecimento da população, o que resulta em significativos desafios socioeconômicos no setor da saúde [Al-Shaqi et al. 2016].

\*Trabalho realizado com apoio da CAPES (Programa Nacional de Cooperação Acadêmica- Procad) e da FAPERGS (Programa Pesquisador Gaúcho - PqG)

Nesta perspectiva, uma alternativa é a utilização de casas inteligentes que proporcionem um ambiente assistido de vivência no contexto da saúde. Nestas residências devem ser contemplados serviços computacionais que possam auxiliar as pessoas nas suas práticas diárias, da forma mais transparente possível [Röcker et al. 2014]. Busca-se assim integrar a tecnologia ao cotidiano do usuário, caracterizando uma infraestrutura computacional de natureza ubíqua que deve exigir o menor envolvimento possível das pessoas no seu gerenciamento, bem como deve possibilitar que as atividades sejam reconhecidas a partir de dados de contexto coletados por sensores, os quais podem estar presentes em objetos carregados pelas pessoas ou incorporados ao ambiente em que as mesmas interagem [Perera et al. 2015].

Este trabalho, denominado EXEHDA-AR (*Execution Environment for Highly Distributed Applications-Activity Recognition*), contribui com o Subsistema de Reconhecimento de Contexto e Adaptação do *middleware* EXEHDA [Lopes et al. 2014], contemplando uma abordagem que permite o reconhecimento de atividades em casas inteligentes. Para tanto, explora métodos baseados em processamento semântico, sendo concebidos componentes arquiteturais integrados ao EXEHDA, bem como um modelo ontológico. O principal diferencial do EXEHDA-AR, quando comparado com os trabalhos relacionados, é a exploração do potencial de *middleware* para reconhecimento não intrusivo de atividades em intervalos de tempo próximos ao momento em que as rotinas estão sendo observadas.

## 2. EXEHDA-AR: Visão Geral e Funcionalidades

A proposta do EXEHDA-AR contempla os seguintes aspectos na sua concepção: (i) ser distribuída, permitindo a coleta e processamento das informações contextuais em diferentes localizações simbólicas; (ii) ser dirigida por eventos, possibilitando a associação de regras aos contextos de interesse das aplicações; e (iii) viabilizar a utilização de técnicas baseadas em especificação de regras para o reconhecimento de atividades. A premissa buscada no EXEHDA-AR é explorar de forma sinérgica a integração destes aspectos no suporte ao reconhecimento de atividades.

O ambiente ubíquo gerenciado pelo EXEHDA é constituído por células nas quais se distribuem os seguintes tipos de dispositivos computacionais: (i) EXEHDAbase: elemento central da célula responsável por todos serviços básicos; (ii) EXEHDA nodo: corresponde aos dispositivos computacionais responsáveis pela execução das aplicações; (iii) EXEHDA nodo móvel: tipo específico de EXEHDA nodo que corresponde aos dispositivos móveis; e (iv) EXEHDA borda: elemento de borda do ambiente ubíquo, responsável por fazer a interoperação entre os serviços do *middleware* e o ambiente.

O *middleware* EXEHDA possui dois tipos de servidores. O Servidor de Borda que é responsável por interagir com ambiente através de sensores e atuadores e o Servidor de Contexto que provê funcionalidades para ciência de contexto. Estes servidores são mapeados sobre o ambiente ubíquo, sendo o Servidor de Borda instanciado em equipamentos do tipo EXEHDA borda, enquanto o Servidor de Contexto é alocado no EXEHDAbase da célula.

Na proposta do EXEHDA-AR o *middleware* tem sua arquitetura estendida estendida com a inclusão de componentes para prover reconhecimento de atividades. De modo mais específico, a abordagem para reconhecimento de atividades no EXEHDA-AR

tem por base o Servidor de Contexto e as funcionalidades de armazenamento e processamento das informações de contexto providas por este. Uma visão geral da arquitetura do Servidor de Contexto é apresentada na Figura 1.

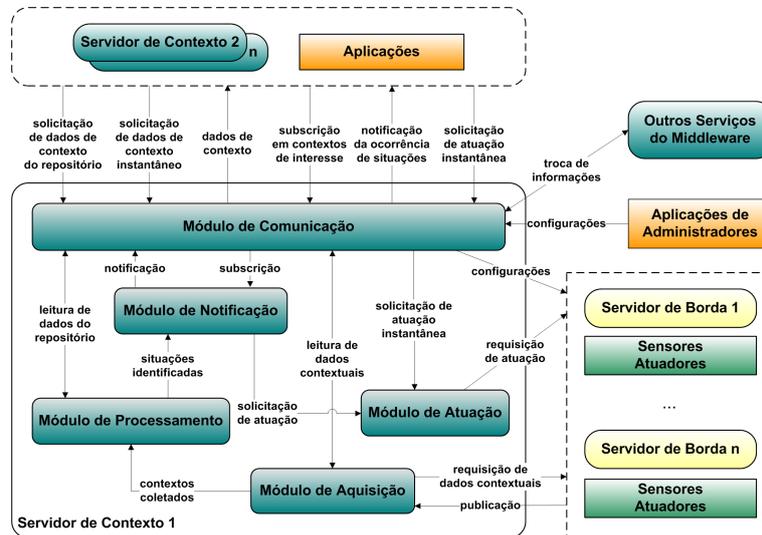


Figura 1. Servidor de Contexto

Os componentes para reconhecimento de atividades concebidos para o EXEHDA-AR foram introduzidos nos módulos de Aquisição e Processamento do Servidor de Contexto. Estes módulos, na perspectiva do EXEHDA-AR, são discutidos nas próximas seções.

## 2.1. Módulo de Aquisição

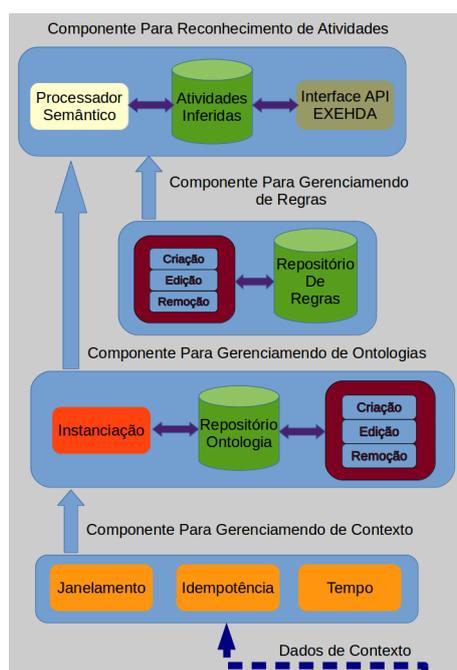
Responsável pelo suporte à captura dos dados contextuais coletados pelos Servidores de Borda, provendo suporte tanto para uma coleta agendada, como para a obtenção instantânea por solicitações de consumidores interessados.

Para o reconhecimento de atividades considerou-se necessário analisar o tempo da sequência de eventos, descrevendo-os como discretos para identificar padrões de interesse. Assim, o modelo de janelas deslizantes utilizado no EXEHDA-AR considerou o trabalho de [Salfner et al. 2010] que possui dois tipos janela: (i) baseada em uma quantidade específica de número de eventos que corresponde a um padrão de interesse; e (ii) janela baseada em tempo, que tem um tempo finito onde eventos de interesse são monitorados.

Como as atividades em uma casa inteligente se apresentam de forma dinâmica, foi concebido um algoritmo de janelamento de tempo finito. Este algoritmo foi implementado em *Python* com o objetivo de agrupar informações coletadas, oferecendo opções de aumentar ou diminuir a janela de tempo de acordo com perfil do morador da casa. Uma vez realizado o janelamento, as informações passam para o Módulo de Processamento.

## 2.2. Módulo de Processamento

Realiza as tarefas pertinentes ao tratamento das informações contextuais para identificar atividades relacionadas aos contextos de interesse das aplicações. Na Figura 2 é apresentada uma visão geral do Módulo de Processamento do EXEHDA-AR.



**Figura 2. Módulo de Processamento**

O Módulo de Processamento é constituído por quatro componentes, cujas funcionalidades são descritas a seguir.

### **Componente para Gerenciamento de Contexto**

Responsável pelas etapas de interpretação e agregação das informações contextuais. Estas etapas tem como objetivo aumentar o grau de abstração para melhorar a disponibilidade e usabilidade no reconhecimento de atividades.

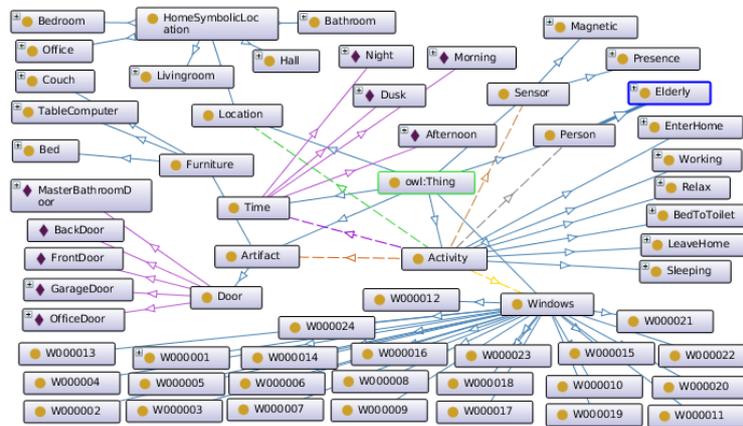
Uma das formas de abstração providas por esse componente é a idempotência, a qual independentemente do número de vezes que sensor detecta eventos, é considerado que mesmo identificou uma ocorrência em uma janela de tempo válida. Outra abstração provida por este componente é a normalização do tipo de dado *date time*, cujas informações são convertidos para milissegundos, com intuito de serem utilizados na construção das regras.

Neste componente são agregadas informações, como identificação do sensor, o horário de início e fim e a identificação da janela em que foi coletado o evento.

### **Componente para Gerenciamento de Ontologias**

Gerencia a camada semântica do Módulo de Processamento, possibilitando a instanciação de ontologias. A ontologia concebida para o reconhecimento de atividades no EXEHDA-AR, apresentada na Figura 3, utiliza como base a ontologia do projeto PalSPOT [Riboni et al. 2011], devido esta ser também instanciada em um *middleware* e focada no reconhecimento de atividades. Foram herdadas da ontologia original as classes *Activity*, *Location*, *Artifact*, *Time* e *Person*, das quais foram criadas subclasses, seus atributos e relacionamentos. Também, foi criada a classe *Windows* com a finalidade de receber as informações do Componente para Gerenciamento de Contexto.

No EXEHDA-AR quando uma atividade é reconhecida, é criada uma nova instância na classe *Activity* relacionando-a com as demais classes para agregar informações relevantes como localização, objeto, segmento do dia e a janela em que a atividade foi reconhecida.



**Figura 3. Visão Geral da Ontologia**

A classe *Activity* tem subclasses com as atividades que se deseja reconhecer, como *Sleeping*, *Working*, *EnterHome*, *LeaveHome* e *BedToToilet*. Cada vez que uma atividade é reconhecida, o EXEHDA-AR cria uma instância na classe correspondente com as informações do tempo inicial e final da janela, objetos, localização, indivíduo e sensores.

As propriedades de dados recebem os valores oriundos dos sensores. Por exemplo, *hasStatusSensor* recebe o valor "dt" quando há eventos detectados. Devido a utilização da idempotência e do janelamento dos eventos de sensores apenas, recebe o valor de tempo do primeiro e do último evento na janela corrente.

A ontologia empregada pelo EXEHDA-AR é armazenada no Repositório de Ontologia e gerenciada por meio da API JENA [Riboni and Bettini 2011].

### Componente para Gerenciamento de Regras

As regras são construídas em SWRL (*Semantic Web Rule Language*) [Okeyo et al. 2014], utilizando os conceitos propostos por [Chen and Nugent 2009], através da relação entre a localização simbólica e objeto para reconhecer qual atividade o indivíduo está realizando. Essas regras são processadas em nível de instâncias e, desta forma, motores de inferência conseguem raciocinar sobre as informações contextuais. As regras são armazenadas no Repositório de Regras, possibilitando ao desenvolvedor a criação, edição e remoção das mesmas.

Considerando que as regras estão associadas aos objetos do ambiente e a localização na casa, a partir dessas relações é possível identificar a atividade que está sendo realizado pelo usuário.

### Componente para Reconhecimento de Atividade

Disponibiliza as funcionalidades para o processo de raciocínio sobre os dados contextuais, visando o reconhecimento das atividades. Neste componente as regras realizam a

inferência nas instâncias da classe *Windows*, criando novas instâncias nas classes referentes às atividades através da função `swrlx:makeOWLThing(?y, ?x)`.

Uma instância é criada quando a regra para reconhecer a atividade tem suas condições atendidas, sendo as informações relacionadas à instância criada, tais como: localização, objeto, sensores envolvidos na atividade.

### 3. EXEHDA-AR: Estudo de Caso

Para realização da avaliação do EXEHDA-AR foi selecionado um *dataset* pertencente ao grupo de pesquisa CASAS<sup>1</sup>, Universidade de Washington. Este *dataset* é proveniente de uma casa denominada Aruba, na qual uma senhora idosa reside sozinha, apresentando características alinhadas com as motivações do EXEHDA-AR.

Para modelar as regras de descrição lógica empregadas na avaliação do EXEHDA-AR foram escolhidas cinco atividades alvo: entrar em casa, sair de casa, dormir, ir da cama para banheiro e trabalhar no escritório. A linguagem SWRL foi utilizada para a elaboração das regras, sendo utilizados métodos descritos no trabalho [Chen and Nugent 2009]. O estudo de caso foi segmentado em cinco cenários que correspondem cada um deles a uma das atividades alvo que se busca reconhecer. Na próxima seção é apresentado o cenário "dormir". A descrição dos demais cenários pode ser encontrada em [Abreu 2017].

#### 3.1. Atividade "dormir"

Maria dirige-se para o quarto, passa pela porta e deita-se na cama. Para inferir essa atividade considera-se que a localização neste cenário é o quarto, o objeto de interação é a cama (representada pelo sensor M003) e a interação da pessoa com objeto deve ser maior que 120000 milissegundos (20 minutos). Com isso, detecta-se que a pessoa está dormindo. Na Figura 4 é mostrada a distribuição dos sensores do quarto.

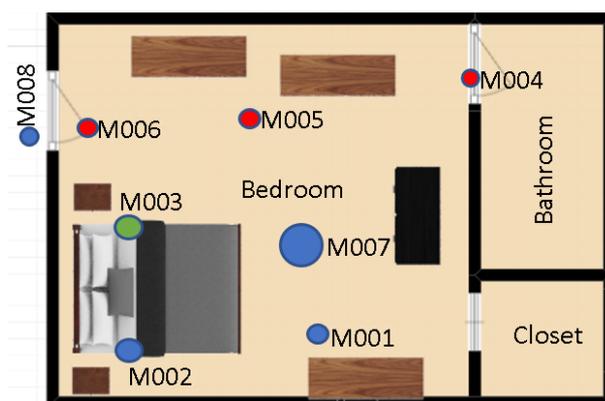


Figura 4. Sensores do quarto

Após observar a distribuição dos sensores pelo quarto, foi definido que o contexto de interesse está associado a detecção de presença pelo sensor M003. Por sua vez, os sensores M005, M006, M004 não devem detectar presença, assim indicando que pessoa permanece na cama. Na Figura 5 é possível observar a regra que foi elaborada para reconhecer a atividade "dormir" na casa Aruba.

<sup>1</sup><http://casas.wsu.edu/>

|  |
|--|
| Name   |
| Sleeping   |
| Comment  |
| Status   |
| Ok   |
| Windows(?w) ^ hasSensorName(?wsn) ^ swrl:equal(?wsn, "M003") ^ hasStatusSensor(?w, ?wss) ^ swrl:equal(?wss, "dt") ^ hasEventBegin(?w, ?eb) ^ hasEventEnd(?w, ?ee) ^ hasSensorName(?w, ?wsn1) ^ swrl:equal(?wsn1, "M004") ^ hasStatusSensor(?w, ?wss1) ^ swrl:equal(?wss1, "ndt") ^ hasSensorName(?w, ?wsn2) ^ swrl:equal(?wsn2, "M005") ^ hasStatusSensor(?w, ?wss2) ^ swrl:equal(?wss2, "ndt") ^ hasSensorName(?w, ?wsn3) ^ swrl:equal(?wsn3, "M006") ^ hasStatusSensor(?w, ?wss3) ^ swrl:equal(?wss3, "ndt") ^ swrl:subtract(?d, ?ee, ?eb) ^ swrl:greaterThan(?d, 1200000) ^ Sensor(?s) ^ hasSensorName(?s, ?sn) ^ swrl:equal(?sn, ?wsn) ^ hasSensorName(?s, ?sn1) ^ swrl:equal(?sn1, ?wsn1) ^ hasSensorName(?s, ?sn2) ^ swrl:equal(?sn, ?wsn2) ^ hasSensorName(?s, ?sn3) ^ swrl:equal(?sn3, ?wsn3) ^ Location(?l) ^ hasLocation(?l, ?lt) ^ swrl:equal(?lt, "Bedroom") ^ Artifact(?ar) ^ hasArtifact(?ar, ?at) ^ swrl:equal(?at, "Bed") ^ Person(?p) ^ hasWindowBegin(?w, ?wb) ^ hasWindowsEnd(?w, ?we) ^ Activity(?a) ^ hasNameActivity(?a, ?an) ^ swrl:equal(?an, "sleeping") ^ Person(?p) ^ hasActor(?p, ?at) ^ swrl:equal(?at, "Maria") ^ swrl:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifact(?an, ?at) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w] |

Figura 5. Regra em SWRL para atividade "dormir"

Utilizando essa regra, o EXEHDA-AR consegue reconhecer 398 das 401 ocorrências registradas na documentação do *dataset*, obtendo nessa atividade uma acurácia de 99,25%. Após a realização das inferências é possível realizar uma busca semântica nos dados através da linguagem SPARQL, para obter mais informações sobre a atividade reconhecida. O resultado da consulta SPARQL mostra diferentes informações referentes a atividade "dormir", tais como: localização, objeto, o ator da atividade, sensores envolvidos, nome da atividade e a instância da janela na qual a atividade foi reconhecida.

### 3.2. Avaliação de Acurácia no EXEHDA-AR

Para avaliação da acurácia no EXEHDA-AR foi aplicado o método desenvolvido por [van Kasteren et al. 2011]. O método consiste em criar um matriz de confusão na qual as linhas representam atividades e as colunas a frequência dos verdadeiros positivos das atividades reconhecidas. A partir da matriz de confusão é possível mensurar a acurácia do reconhecimento das atividades. A seguinte fórmula apresenta uma visão geral do método:

$$Accuracy = \frac{\sum_{i=1}^Q TP_i}{Total}$$

**TP** representa o número de verdadeiros positivos das atividades reconhecidas, **Q** o número de atividades a serem reconhecidas (dormir, entrar em casa...) e **Total** representa o número total de janelas de tempo deslizantes no conjunto de dados.

Assim, considerando esse método, foi construída uma matriz de confusão, mostrada na Figura 6, para avaliar a acurácia do EXEHDA-AR. Esta matriz apresenta uma estrutura de 5x5 colunas, onde cada célula com fundo amarelo representa o número de atividades reconhecidas. As atividades estão representadas por números de 1 a 5. A última coluna Ac/Atv representa a acurácia por atividade e a última linha a acurácia média do EXEHDA-AR. A representação das atividades estão descritas a seguir: (1) "dormir" tem 401 verdadeiros positivos; (2) "cama para banheiro" tem 157 verdadeiros positivos; (3) "entrar em casa" tem 431 verdadeiros positivos; (4) "sair de casa" tem 431 verdadeiros positivos; e (5) "trabalhar no escritório" tem 171 verdadeiros positivos.

Foi atingida uma acurácia média de 94,36%, sendo que a atividade "dormir" teve o melhor resultado, enquanto que a atividade "trabalhar no escritório" apresentou um percentual menor de acurácia. Acurácia obtida pelo EXEHDA-AR mostrou-se promissora frente a trabalhos encontrados na revisão do estado da arte como [Cook 2012], [Riboni et al. 2011], [Van Kasteren et al. 2008] e [Tapia et al. 2004].

| Atividade      | 1   | 2   | 3      | 4   | 5   | Ac/Atv |
|----------------|-----|-----|--------|-----|-----|--------|
| 1              | 398 | -   | -      | -   | -   | 99,25% |
| 2              | -   | 154 | -      | -   | -   | 98,08% |
| 3              | -   | -   | 400    | -   | -   | 92,80% |
| 4              | -   | -   | -      | 405 | -   | 93,96% |
| 5              | -   | -   | -      | -   | 150 | 87,71% |
| Acurácia Média |     |     | 94,36% |     |     |        |

**Figura 6. Matriz de confusão do EXEHDA-AR**

#### 4. Trabalhos Relacionados

Tendo como base as premissas que motivaram o desenvolvimento desta pesquisa, os seguintes trabalhos foram selecionados: (I) *Context-Aware System Infrastructure for Monitoring* [Ni et al. 2016]; (II) *AAL Domain Ontology for Event-based Human Activity Recognition* [Culmone et al. 2014]; (III) *Activity Recognition using Context-Aware Infrastructure Ontology in Smart Home Domain* [Wongpatikaseree et al. 2012]; e (IV) *Semantic Smart Homes: Towards Knowledge Rich Assisted Living Environments* [Chen et al. 2009].

A Tabela 1 mostra uma comparação entre os trabalhos relacionados e o EXEHDA-AR. Para esta comparação foram considerados os seguintes critérios: (i) proposição de um modelo semântico específico; (ii) forma de avaliação do modelo; (iii) verificação da acurácia; (iv) exploração de conceitos de objeto e localização; e (v) tipo de regra empregado para o reconhecimento de atividades.

**Tabela 1. Comparação dos Trabalhos Relacionados**

| Trabalhos Relacionados | Propõe Modelo Ontológico    | Utiliza dataset | Verifica a Acurácia | Explora Objeto e Localização | Tipo de Regras |
|------------------------|-----------------------------|-----------------|---------------------|------------------------------|----------------|
| <b>I</b>               | Baseado em Modelo Existente | Não             | Não                 | Parcialmente                 | JENA           |
| <b>II</b>              | Baseado em Modelo Existente | Não             | Não                 | Parcialmente                 | JENA           |
| <b>III</b>             | Propõe Modelo Próprio       | Não             | Não                 | Utiliza                      | JENA           |
| <b>IV</b>              | Propõe Modelo Próprio       | Não             | Não                 | Utiliza                      | AXIOMA         |
| <b>EXEHDA-AR</b>       | Baseado em Modelo Existente | Sim             | Sim                 | Utiliza                      | JENA e SWRL    |

O EXEHDA-AR tem o seu modelo ontológico baseado na ontologia PalSPOT. Por sua vez, os trabalhos I e II, são baseados nas ontologias DOLCE e DnS combinadas, sendo estas ontologias empregadas tanto para o reconhecimento de atividades como no gerenciamento da própria infraestrutura do ambiente inteligente. No caso do EXEHDA-AR, o gerenciamento do ambiente é feito por serviços do *middleware* EXEHDA. Os trabalhos III e IV desenvolveram seus modelos apenas utilizando os conceitos de localização e objeto, sendo que o trabalho IV emprega somente uma ontologia assim como EXEHDA-AR. Já o trabalho III utiliza duas ontologias uma para gerir o contexto da casa inteligente e outra para reconhecer atividades.

O EXEHDA-AR utiliza um *dataset* público para avaliar sua proposta, com dados de contextos coletados em um ambiente real, enquanto os trabalhos simulam atividades sem que as mesmas tenham sido coletadas de um ambiente real. O trabalho III faz uso de um *framework* chamado DB2R para criar suas ontologias a partir de uma base de dados relacional, realizando o reconhecimento analisando os *logs* de uma casa inteligente por meio de regras de descrição, o que o deixa próximo a abordagem do EXEHDA-AR.

No EXEHDA-AR a coleta dos dados ocorre através do Servidor de Borda do *middleware* EXEHDA. De acordo com o dado coletado e a natureza da aplicação do usuário, os dados podem ser convertidos através de regras para um nível maior de abstração. Os demais trabalhos teriam que criar esses conceitos na ontologia ou ontologias como é caso do II e III, para realizar essa conversão e abstrair o contexto para nível mais alto.

Outro diferencial do EXEHDA-AR em relação aos trabalhos relacionados é que o EXEHDA-AR verifica a acurácia com relação à inferência em atividades diárias, uma vez que esta medida representa a qualidade do modelo ontológico frente aos desafios no reconhecimento de atividades humanas por meio de sensores não intrusivos.

## 5. Considerações Finais

A principal contribuição deste trabalho é a concepção dos componentes arquiteturais integrados ao *middleware* EXEHDA, os quais são responsáveis pelo reconhecimento de atividades, utilizando processamento semântico. A proposta do EXEHDA-AR, empregando sensores associados ao ambiente, e não diretamente no usuário, mostrou-se oportuna para a detecção de atividades, não necessitando que o usuário se envolva de forma intencional com o procedimento de monitoramento das suas atividades, caracterizando assim uma abordagem não intrusiva.

Os resultados do estudo de caso realizado no domínio de uma casa inteligente utilizando dados reais, mostraram-se promissores com uma acurácia média de 94,36% de reconhecimento de atividade.

Dentre os aspectos levantados para continuidade do trabalho de pesquisa destacam-se: (i) desenvolver um modelo híbrido para reconhecimento de atividades, empregando técnicas baseadas em aprendizagem de máquina e em processamento semântico; (ii) viabilizar o reconhecimento de atividades oportunístico direcionado a atender a dinamicidade usual dos ambientes de IoT, quanto a entrada e saída de recursos de sensoriamento; e (iii) explorar o emprego dos mecanismos concebidos para o EXEHDA-AR em outros cenários de uso.

## Referências Bibliográficas

- Abreu, E. (2017). *Reconhecimento de Atividades em Casas Inteligentes: Uma abordagem não Intrusiva explorando Processamento Semântico*. Dissertação (mestrado em ciência da computação), Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, RS.
- Al-Shaqi, R., Mourshed, M., and Rezgui, Y. (2016). Progress in ambient assisted systems for independent living by the elderly. *SpringerPlus*, 5(1):624.
- Chen, L. and Nugent, C. (2009). Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430.
- Chen, L., Nugent, C., Mulvenna, M., Finlay, D., and Hong, X. (2009). Semantic smart homes: towards knowledge rich assisted living environments. In *Intelligent Patient Management*, pages 279–296. Springer.
- Cook, D. J. (2012). Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, 27(1):32–38.

- Culmone, R., Falcioni, M., Giuliadori, P., Merelli, E., Orru, A., Quadrini, M., Ciampolini, P., Grossi, F., and Matrella, G. (2014). Aal domain ontology for event-based human activity recognition. In *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, pages 1–6. IEEE.
- Lopes, J. L., de Souza, R. S., Geyer, C., da Costa, C., Barbosa, J., Pernas, A. M., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *J. UCS*, 20(9):1327–1351.
- Ni, Q., García Hernando, A. B., and Pau de la Cruz, I. (2016). A context-aware system infrastructure for monitoring activities of daily living in smart home. *Journal of Sensors*, 2016.
- Okeyo, G., Chen, L., Wang, H., and Sterritt, R. (2014). Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10:155–172.
- Perera, C., Member, C. H. L., Jayawardena, S., and Chen, M. (2015). Context-aware computing in the internet of things: A survey on internet of things from industrial market perspective. *arXiv preprint arXiv:1502.00164*.
- Riboni, D. and Bettini, C. (2011). Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289.
- Riboni, D., Pareschi, L., Radaelli, L., and Bettini, C. (2011). Is ontology-based activity recognition really effective? In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 427–431. IEEE.
- Röcker, C., Ziefle, M., and Holzinger, A. (2014). From computer innovation to human integration: current trends and challenges for pervasive healthtechnologies. In *Pervasive Health*, pages 1–17. Springer.
- Salfner, F., Lenk, M., and Malek, M. (2010). A survey of online failure prediction methods. *ACM Computing Surveys (CSUR)*, 42(3):10.
- Tapia, E. M., Intille, S. S., and Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*, pages 158–175. Springer.
- Van Kasteren, T., Noulas, A., Englebienne, G., and Kröse, B. (2008). Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM.
- van Kasteren, T. L., Alemdar, H., and Ersoy, C. (2011). Effective performance metrics for evaluating activity recognition methods. *ARCS 2011*.
- Wongpatikaseree, K., Ikeda, M., Buranarach, M., Supnithi, T., Lim, A. O., and Tan, Y. (2012). Activity recognition using context-aware infrastructure ontology in smart home domain. In *Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on*, pages 50–57. IEEE.

# Uma Máquina de Estados para Especificação de Códigos de Simulação para Redes de Sensores sem Fio Urbanas

Marcos Carrero<sup>1,3</sup>, Katriny Zamproni<sup>1</sup>, Martin A. Musicante<sup>2</sup>,  
Aldri Santos<sup>1</sup>, Carmem Hara<sup>1</sup>

<sup>1</sup>DINF – Universidade Federal do Paraná – UFPR – Paraná, Brasil

<sup>2</sup>DIMAp – Universidade Federal do Rio Grande do Norte – UFRN – Natal, Brasil

<sup>3</sup>FAE Centro Universitário – Paraná, Brasil

{macarrero, kz14, aldri, carmem}@inf.ufpr.br, mam@dimap.ufrn.br

**Abstract.** *Ubiquitous urban sensing systems face challenges related to the large number of sensors in the network and to the dynamicity of data generation. Individually, sensor activities are triggered in response to events. However, modeling the collaborative process requires the execution of functions that start not only in response to an event, but also based on a logical condition. This paper proposes a state machine with two types of transitions: event-based and logic-based. A case study which models a sensor storage system is presented. Its development using a framework of reusable components shows a correspondence between our machine and the code, which shows the machine usefulness for code development in simulation environments.*

**Resumo.** *Os sistemas ubíquos de sensoriamento urbano enfrentam desafios relacionados à grande quantidade de sensores na rede e à dinamicidade de geração de dados. Individualmente, os sensores reagem em resposta a eventos. No entanto, a modelagem do processo colaborativo exige mudanças de estado dos sensores não apenas como resposta a um evento, mas também por uma condição lógica. Este artigo propõe uma máquina de estados com dois tipos de transição: por evento e por lógica. Um estudo de caso que considera um modelo de armazenamento em sensores, desenvolvido com o apoio de um framework de componentes reusáveis, mostra uma correspondência entre a máquina proposta e o código, facilitando a sua implementação em ambientes de simulação.*

## 1. Introdução

O conceito de cidades inteligentes tem por objetivo prover serviços ubíquos para oferecer qualidade de vida e conforto para sua população. Ela é deslumbrada em diversas áreas de aplicação como alertas de congestionamentos de tráfego [Araújo et al. 2014], aplicações médicas [Cremonezi et al. 2017] e casas inteligentes [Filho et al. 2015]. Uma questão primordial aos serviços inteligentes consiste no desafio da coleta e disseminação de grande quantidade de dados nestes ambientes físicos e dinâmicos. As Redes de Sensores Sem Fio (RSSFs) têm oferecido essa infraestrutura para o desenvolvimento de diversos serviços para as redes urbanas em diferentes contextos de aplicação [Carrero et al. 2015].

Os sistemas ubíquos de sensoriamento urbano são tipicamente sistemas orientados a eventos [Muñoz and Leone 2017]. A fim de oferecer serviços ubíquos ao cidadão e ao

gestor público, as RSSFs podem ser usadas como infraestrutura para atender à grande demanda pelas aplicações de sensoriamento em diversos domínios. Dada a natureza reativa das RSSFs, a maioria dos simuladores existentes, dentre eles o NS2, OMNeT++ e TOS-SIM, são baseados em eventos. É um modelo intuitivo de programação que associa eventos a ações, mas oferece um **baixo nível de abstração**. Assim, faz-se necessário o uso de ferramentas que reduzam a complexidade do desenvolvimento de sistemas ubíquos orientados a eventos, aumentando a produtividade e minimizando possíveis erros de projetos.

As máquinas de estados (MEs) são uma abordagem usada para tratar a complexidade do desenvolvimento de programas orientados a eventos, possibilitando abstrair detalhes de implementação [Desai et al. 2013]. Os estados são representações lógicas que descrevem o contexto atual de execução. As MEs são frequentemente usadas para especificar códigos para RSSFs, uma vez que os sensores, considerados individualmente, reagem a eventos como a obtenção de uma nova leitura ou o recebimento de uma mensagem. No entanto, sensores e outros dispositivos autônomos frequentemente realizam funções colaborativas, que exigem mudanças de estado dos dispositivos não apenas como resposta a um evento, mas também por uma condição lógica ou por um temporizador. Logo, a modelagem de sistemas com MEs torna-se mais complexa, dificultando sua implementação em simuladores orientados a eventos. Portanto, faz-se necessária uma abordagem de alto nível que associe de maneira clara modelos de máquinas de estados complexas, que tratam diferentes casos de mudanças de estados e que facilitem sua codificação.

Para reduzir o esforço e a complexidade do desenvolvimento de aplicações, [Cañete et al. 2011] propõem um modelo de componentes de software orientado a serviços para desenvolver sistemas complexos a partir de serviços mais simples. No entanto, o modelo proposto não é flexível para a criação de sistemas autônomos, como por exemplo, em aplicações que realizam a formação de agrupamentos na própria rede de sensores. Estratégias propostas por [Krämer et al. 2013] usam máquinas de estados para modelar sistemas orientados a eventos. No entanto, os estudos de caso mostram modelos de máquinas de estados simples, o que não é a realidade encontrada em cenários urbanos.

Este artigo propõe uma máquina de estados para fazer o detalhamento do fluxo de operações em um simulador baseado em eventos, criando dois tipos de transição: baseada em evento e baseada em lógica. Este detalhamento pode então ser utilizado para estruturar o código de simulação. Um estudo de caso descreve uma ME para o DCSSC (*Distributed Clustering Scheme based on Spatial Correlation in WSNs*) [Le et al. 2008], um modelo apropriado para o sensoriamento urbano, que realiza agrupamentos explorando a similaridade de dados. A ME foi implementada usando o *framework* RCBM, que oferece um conjunto de bibliotecas e componentes reusáveis para desenvolver sistemas de simulação para o NS2. O código do DCSSC desenvolvido com o apoio do *framework* RCBM mostrou uma correspondência direta entre os tipos de estados propostos na ME e o seu detalhamento no programa de simulação. Além disso, os resultados mostram uma reutilização de código de até 71.6%, o que demonstra o potencial da proposta para facilitar o desenvolvimento de novas aplicações para RSSFs e outras de natureza reativa e colaborativa, como os sistemas ubíquos e pervasivos.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o modelo de máquina de estados. A Seção 4 mostra um estudo de caso da aplicação da máquina em um modelo de armazenamento de

RSSF urbana. A Seção 5 apresenta os detalhes da implementação e os resultados obtidos na avaliação de desempenho. A conclusão e trabalhos futuros são discutidos na Seção 6.

## 2. Trabalhos Relacionados

Na literatura sobre técnicas de desenvolvimento de sistemas para as redes de sensores [Malavolta and Muccini 2014], verificou-se a inexistência de uma abordagem que trate da modelagem e implementação de uma maneira sistemática, requisitos necessários para atender à grande demanda por serviços ubíquos de grande escala. O trabalho proposto por [Krämer et al. 2013] descreve uma biblioteca de desenvolvimento de aplicativos para RSSFs orientada à máquina de estados. Um programa pode ser decomposto em módulos, cada um executando sua própria máquina de estados. A coordenação de vários módulos é feita pela máquina de estados do módulo principal. Embora a biblioteca promova modularidade e reusabilidade de código, a análise de desempenho da biblioteca não demonstra que sua arquitetura seja capaz de implementar sistemas escaláveis e autônomos, requisitos desejados para aplicações urbanas.

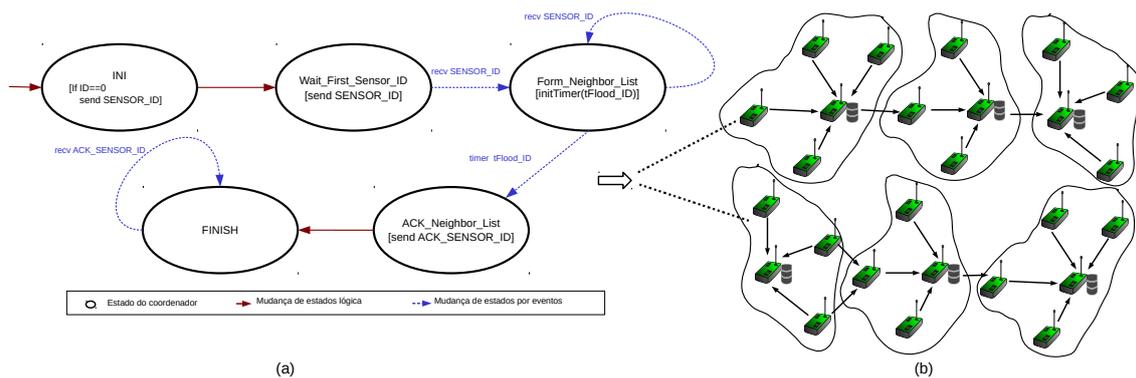
O modelo de ME descrita por [Cecílio and Furtado 2012] é composta por duas classes de estados: estados que tratam de requisitos funcionais e estados que tratam de requisitos não-funcionais. Por exemplo, coletar e encaminhar dados é um requisito funcional e reencaminhar dados quando a perda de pacotes for maior que 1% é um requisito não-funcional. Contudo, o foco deste artigo considera os requisitos funcionais encontrados em RSSFs urbanas enquanto a proposta deles lida com os requisitos não-funcionais de ambientes industriais. As estratégias usadas por Tokenit [Taherkordi et al. 2015] integram um ambiente de modelagem e implementação orientado à máquina de estados. O modelo define uma máquina de estados como um conjunto de atividades, que são operações executadas por um estado. As mudanças de estado ocorrem por eventos de *timers* ou eventos assíncronos. A detecção de um movimento, por exemplo, é um evento assíncrono. A máquina de estados é descrita em XML e o compilador do *framework* gera automaticamente código para a plataforma de execução do sistema operacional Contiki. Contudo, a avaliação de desempenho não aborda aspectos sobre a escalabilidade do sistema.

## 3. Uma Máquina de Estados Orientada a Eventos

Em RSSFs, os sensores podem desempenhar diversas funções, desde a participação no roteamento de mensagens em uma comunicação multi-salto até a atuação como um repositório de dados coletados do ambiente. Estas funcionalidades são em geral acionadas através do recebimento de mensagens, que podem ser modeladas como transições de uma máquina de estados (ME) orientada a eventos. No entanto, existe uma dicotomia entre o entendimento intuitivo de uma transição em uma máquina de estados e o seu funcionamento em uma RSSF, uma vez que a máquina de estados deve representar a funcionalidade de cada sensor *individualmente*. Logo, não há necessariamente uma correspondência entre o estado do sensor que envia uma mensagem e o estado do sensor que a recebe. Ou seja, um sensor pode enviar uma mensagem a partir de um estado  $e_1$ , mas o sensor que a recebe pode estar em um estado  $e_2$ , que não é necessariamente o próximo estado do sensor origem da mensagem. Assim, a mudança de estado por evento pode não estar necessariamente relacionada ao estado do sensor emissor e o envio de mensagens não necessariamente determina uma mudança de estado. Em suma, não há correspondência direta entre

transições na máquina e a comunicação entre sensores. No entanto, as mudanças de estado do sensor origem e destino são representadas na mesma máquina, que é executada em todos os sensores, como ilustrado pela Figura 1(b). Experiências anteriores de desenvolvimento de código de simulação para RSSF [Furlaneto et al. 2012, Gonçalves et al. 2012], mostraram que esta dicotomia dificulta o desenvolvimento de programas.

O objetivo da ME proposta neste artigo é oferecer suporte para o desenvolvedor especificar o comportamento do sistema antes de iniciar a implementação, através da criação de dois tipos de transição (por evento e lógica), além de uma notação para o envio de mensagens. A ME combina elementos de máquinas de estados finitos, como estados e transições, com princípios de programação reativa dos simuladores de RSSFs, que associam eventos a um conjunto de ações. As ferramentas visuais de modelagem de máquina de estados que seguem o padrão da *UML StateChart*, podem ser usadas para criar as MEs desenvolvidas neste artigo. Um exemplo desta máquina está ilustrada na Figura 1(a). Cada estado possui um nome, como *Ini* e *Wait\_First\_Sensor\_ID*, e pode possuir uma anotação (entre colchetes) sobre o envio de mensagens ou inicialização de temporizadores. As transições podem ser por evento (representadas em azul) ou lógica (representadas em vermelho):



**Figura 1. Um modelo de máquina de estados para descoberta de vizinhos**

- Mudança de estado por evento: são aquelas que ocorrem quando um sensor recebe algum tipo de mensagem ou então após o tempo de expiração de um temporizador.
- Mudança de estado lógico: é realizada com base no resultado de uma computação.

O exemplo da Figura 1(a) descreve o algoritmo de descoberta de vizinhos por inundação na rede. No estado inicial *INI*, o sensor com identificador (ID) zero envia a mensagem *SENSOR\_ID* para seus vizinhos. O sensor zero e os demais sensores realizam uma mudança lógica do estado *INI* para o estado *Wait\_First\_SENSOR\_ID*. Os sensores que receberem a primeira mensagem *SENSOR\_ID* armazenam o ID contido na mensagem e enviam seu identificador para seus vizinhos. Em seguida, realizam uma mudança por evento do estado *Wait\_First\_SENSOR\_ID* para o estado *Form\_Neighbor\_List*. No estado *Form\_Neighbor\_List*, os sensores continuam armazenando o ID dos demais vizinhos durante um certo tempo  $t_{Flood\_ID}$ . Quando o tempo expirar, os sensores fazem uma mudança para o estado *ACK\_Neighbor\_List* e enviam uma mensagem de *ACK* para os sensores conhecidos. Após o envio da mensagem, os sensores realizam mudança lógica para o estado *FINISH*, armazenando as mensagens de *ACK* recebidas. Ao final da inundação, todos os sensores conhecem seus vizinhos.

Este é um exemplo bastante simples, no qual algumas características de RSSFs não ficam tão evidentes. Elas advêm da natureza distribuída das RSSFs, na qual cada sensor executa a máquina de estados de forma independente. Dessa forma, a modelagem destes sistemas, que frequentemente realizam funções colaborativas assíncronas, na qual múltiplas máquinas podem estar em estados distintos, não é uma tarefa trivial. A próxima seção apresenta um estudo de caso que salienta estas características e mostra a efetividade da máquina proposta para a representação do modelo.

## 4. Estudo de Caso

Um estudo de caso foi desenvolvido para demonstrar o uso da máquina de estados em outros contextos. Como o foco é em aplicações urbanas, foi desenvolvido um estudo de caso a partir do modelo de armazenamento em RSSFs chamado *Distributed Clustering Scheme based on Spatial Correlation in WSNs* (DCSSC) [Le et al. 2008]. O algoritmo proposto constrói e mantém os *clusters* de forma distribuída e dinâmica. A abordagem de agrupamento por similaridade de dados em redes de larga escala torna o DCSSC um modelo apropriado para o sensoriamento urbano.

### 4.1. O Modelo DCSSC

No modelo DCSSC, cada sensor comunica-se apenas com os sensores que estão à distância de um salto e todos têm os dados relativos ao nível de energia de seus vizinhos. Esta informação é enviada por meio de mensagens do tipo HELLO trocadas periodicamente. O modelo da rede consiste em uma estação-base e N sensores. São atribuídos estados a todos os dispositivos, que determinam seu papel na rede. No início, os sensores estão no estado INI e ao final da fase de construção de agrupamentos eles estarão em um dos seguintes estados: CH (*cluster-head*), GW (*gateway*), EXT (*cluster-extend*) ou MEM (*member*). Os sensores com os estados CH, GW e EXT ficam nesse estado até o fim da fase de construção e são chamados de nodos de *backbone*. Existem também estados temporários, como o INI, GWR (*gateway-ready*) e CHC (*cluster-head candidate*).

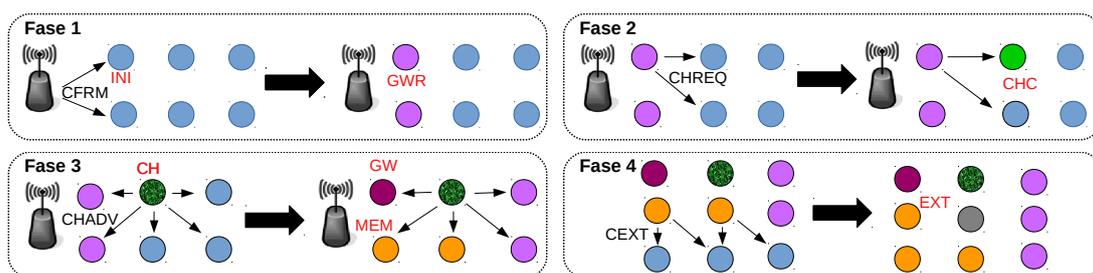


Figura 2. Fases do algoritmo DCSSC

As mensagens de formação de agrupamento incluem sempre a média de valores das leituras aferidas no tempo que precede o envio da mensagem. Baseado no tipo da mensagem, o receptor irá mudar seu estado, criar uma nova mensagem e propagar para seus vizinhos. A nova mensagem também deve conter o identificador (ID) do emissor original da mensagem recebida. A Figura 2 ilustra as fases da simulação que inicia com a construção de agrupamentos, determinada pelos passos que seguem:

**Fase 1:** A estação-base inicia a formação de agrupamentos fazendo *broadcast* de uma

mensagem do tipo  $CFRM$  (*Cluster Formation Message*). Cada sensor que recebe essa mensagem e está no estado  $INI$  muda seu estado para  $GWR$ , criando dois *timers*,  $t_{req}$  e  $t_{wait}$ , escolhidos aleatoriamente. Quando o tempo definido por  $t_{req}$  expira, sensores no estado  $GWR$  fazem o envio de uma mensagem  $CHREQ$  (*Cluster Head Request*) para seus vizinhos a fim de encontrar um *cluster-head*.

**Fase 2:** Um sensor  $INI$  ao receber a mensagem  $CHREQ$  calcula sua similaridade com o sensor  $GWR$  que enviou a mensagem. Os sensores que forem altamente correlacionados mudam seu estado para  $CHC$  (candidatos a *cluster-head*). Os nodos  $CHC$  criam um novo *timer*  $t_{adv}$ , baseado no seu nível relativo de energia. Um sensor  $CHC$  se declara como  $CH$  fazendo *broadcast* da mensagem  $CHADV$  (*Cluster Head Advertisement*) quando seu *timer*  $t_{adv}$  expirar. Como  $t_{adv}$  é inversamente proporcional ao seu nível de energia, sensores com maiores níveis de energia têm maior chance de serem eleitos como líder.

**Fase 3:** Qualquer sensor em um estado temporário ( $INI$ ,  $GWR$ ,  $CHC$ ), ao receber uma mensagem  $CHADV$ , calcula a similaridade com o sensor que enviou a mensagem. Se eles forem altamente correlacionados, o receptor se torna membro (estado  $MEM$ ) do *cluster* formado por aquele  $CH$ . Do contrário, ele vai para o estado  $GWR$ , seguindo os mesmos passos definidos anteriormente para um sensor no estado  $GWR$ . Todo sensor altamente correlacionado compara o  $ID$  da fonte original da mensagem recebida com o seu próprio  $ID$ . Se forem iguais, ou seja, o  $CHADV$  recebido foi uma resposta ao seu próprio  $CHREQ$ , o sensor no estado  $GWR$  altera seu estado para  $GW$ . Quando o *timer*  $t_{wait}$  expira, o sensor no estado  $GWR$  muda seu estado para  $CHC$  se não houver nenhuma mensagem  $CHADV$  proveniente dos seus vizinhos.

**Fase 4:** Os sensores nos estados  $GW$  e  $MEM$  propagam a mensagem de formação enviando mensagens de *broadcast* do tipo  $CEXT$  (*Cluster Extend*), que incluem a média das leituras do  $CH$  que originou a mensagem, e não do sensor atual. Após receber uma mensagem  $CEXT$ , todo sensor em estado temporário ( $INI$ ,  $GWR$ ,  $CHC$ ) calcula a sua similaridade com relação ao  $CH$ . Se forem altamente correlacionados, ele integra o *cluster* formado e se torna um membro (estado  $MEM$ ), fazendo com que o transmissor da mensagem  $CEXT$  vá para o estado  $EXT$ . Caso não seja altamente correlacionado, ele vai para o estado  $GWR$  e repete os passos atribuídos a ele. O modelo  $DCSSC$  prioriza como líderes de agrupamentos os sensores com maior nível relativo de energia dentre os que apresentam leituras similares, prolongando a vida útil da rede. Os nodos com os papéis de  $CH$ ,  $GW$  e  $EXT$  (nodos de *backbone*) têm a função de coletar dados das leituras dos sensores, que enviam seus dados através do *backbone* a cada intervalo de tempo segundo o esquema de escalonamento *Round-Robin*. As leituras recebidas são armazenadas no  $CH$ , e podem ser comprimidas a fim de reduzir o espaço necessário de armazenamento. Os dados agregados são transmitidos do  $CH$  para o  $GW$  do respectivo *cluster*, que encaminha para um sensor vizinho que pertença a outro *cluster*. Dessa forma os dados trafegam na rede através de nodos intermediários, até chegar à estação-base, com menor custo de comunicação.

## 4.2. A Máquina de Estados do modelo $DCSSC$

A máquina de estados foi aplicada para descrever a especificação formal da coordenação do fluxo de execução do modelo  $DCSSC$ . O fluxo de execução do  $DCSSC$  segue uma máquina de estados como representado na Figura 3(b). Todos os nodos iniciam no estado  $INI$ , e modificam seu estado ao longo da execução. Neste exemplo, as características distintas da máquina proposta ficam mais evidentes. Por exemplo, todos os sensores iniciam no estado  $INI$  e a mensagem  $CFRM$  é transmitida por *broadcast* pela estação base. Aque-

les que a recebem passam para o estado GWR, que esperam por um tempo aleatorizado  $t_{req}$  para passar para o estado TREQ e enviar uma mensagem CHREQ. Os sensores que recebem esta mensagem estão no estado INI e passam ao estado selectCH, no qual a similaridade de dados com o sensor emissor é calculado para determinar se ele passa ao estado CHC. Caso as condições para mudança de estado para CHC não sejam satisfeitas, o sensor retorna ao estado INI para que possa receber mensagens de outros sensores.

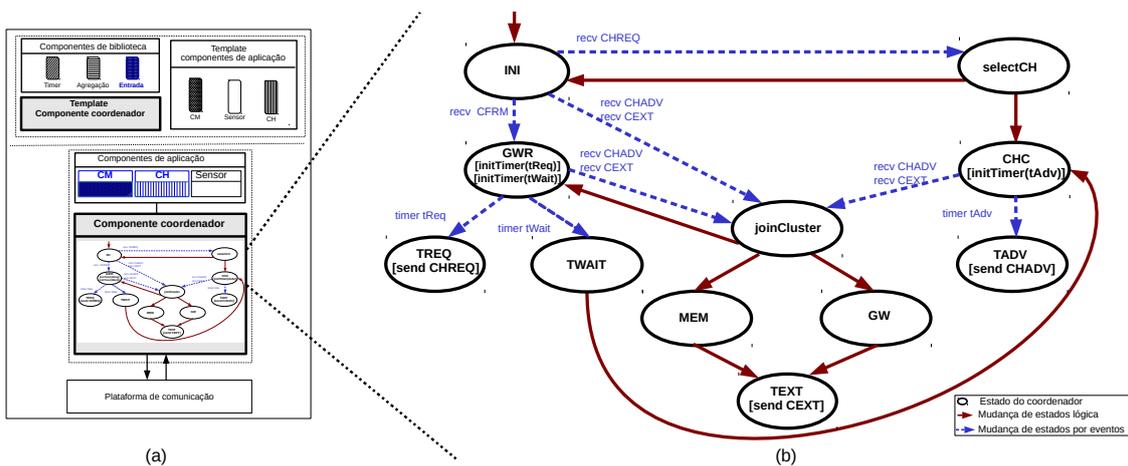


Figura 3. Visão do *framework* RCBM e da máquina de estados do DCSSC

A próxima seção descreve a implementação da máquina de estados do modelo DCSSC. O *framework* RCBM [Carrero et al. 2017], ilustrado pela Figura 3(a), descreve uma metodologia para o desenvolvimento de sistemas de armazenamento para RSSFs orientado a componentes. O RCBM oferece suporte para atender à grande demanda por serviços e aplicações pois possibilita a reusabilidade de código, facilitando o desenvolvimento de novos modelos. O RCBM é um metamodelo que utiliza entidades, propriedades e funções para descrever o que seriam as instâncias desse metamodelo, ou seja, os diversos modelos de armazenamento. No RCBM, cada entidade é associada a um conjunto de componentes que implementam as funcionalidades comuns aos modelos. Por exemplo, para a máquina de estados ilustrada pela Figura 3(b), as entidades são os próprios sensores, os agrupamentos formados por *cluster members* (CM) e os líderes do grupo (CH).

## 5. Implementação

O RCBM é um modelo baseado em componentes reutilizáveis para a simulação de modelos de armazenamento para RSSFs. O RCBM foi implementado no simulador NS2 com o objetivo de promover a reutilização de códigos de simulação a partir de componentes existentes. A arquitetura do RCBM, ilustrada pela Figura 3(a), é composta por três camadas: especificação, implementação e comunicação. O nível superior descreve a camada de especificação, composta pelos componentes de biblioteca, *template* do coordenador e *templates* dos componentes de aplicação. Esses elementos são utilizados da forma como foram especificados, sem a necessidade de alterações realizadas pelo usuário. No nível inferior, estão os componentes de aplicação e o componente coordenador, que de fato serão construídos pelo desenvolvedor, seguindo as definições da camada de especificação. A camada de comunicação é a responsável por interligar as camadas de especificação e de implementação, fornecendo a infraestrutura de comunicação entre os componentes.

O *framework* RCBM já possui a definição de interfaces de funções para os componentes CH e CM, nos quais os estados `selectCH` e `joinCluster` da Figura 3(b), foram adaptados para o modelo DCSSC. No entanto, o desenvolvimento do código de coordenação entre os componentes é responsabilidade do usuário. A partir da máquina de estados ilustrada na Figura 3(b), o componente coordenador, que implementa os diversos estados e transições do modelo RCBM, foi desenvolvido através de duas funções principais no NS2: `recv` e `TimerHandle`. A função `recv` é responsável pelo recebimento de pacotes pelos sensores e trata essas mensagens. É a parte do código que faz as transições do estado do tipo “mudança por evento” e também faz a chamada para os componentes externos. O `recv` é composto por um comando *switch*, com vários *cases*, cada um equivalente a um estado que pode ser atingido pelas mudanças de estado por evento. Cada mensagem possui um ID, e dentro dela existem alguns parâmetros como o ID do emissor, suas coordenadas e leituras, além do ID do emissor da última mensagem recebida pelo sensor. A função `TimerHandle`, é executada quando existe uma transição ativada quando um *timer* expira e uma nova rodada (ou *round*) é iniciada. O `TimerHandle` cria pacotes e os envia conforme o tipo da rodada atual. Os detalhes da implementação das funções `recv` e `TimerHandle` podem ser verificados no código fonte<sup>1</sup> da implementação do DCSSC.

Analisando o código do NS2, verifica-se que há uma correspondência direta entre os tipos de transição propostos na máquina de estados e o seu detalhamento no componente coordenador do programa de simulação. Esta distinção conceitual de tipos de transição facilita o desenvolvimento de código, mantendo a reutilização promovida pelo *framework* RCBM, como mostram os experimentos detalhados na próxima seção. É importante ressaltar que o componente coordenador deve ser implementado pelo programador e que sua função é fazer a junção dos componentes, coordenando as interações entre eles. O RCBM é mais flexível do que modelos baseados em componentes anteriores para RSSFs, porém possui como limitação a falta de uma especificação formal para a coordenação do fluxo de execução dos componentes. A máquina de estados proposta na Seção 3 supre esta deficiência, com uma proposta para especificar o fluxo de operações do coordenador, facilitando sua implementação.

## 5.1. Avaliação

Essa seção mostra a avaliação do sistema DCSSC, desenvolvido a partir de técnicas que integram reuso de componentes de software e máquina de estados que descrevem o comportamento dinâmico do sistema. Foram realizados dois experimentos. No primeiro foi usada a métrica de contagem de linhas de código (LOC) para analisar a porcentagem do código que foi possível reutilizar do modelo RCBM. No segundo, analisou-se a métrica CBO para avaliar o grau de acoplamento entre os componentes do sistema.

**Reutilização de código:** Para analisar a eficácia do modelo RCBM durante o desenvolvimento de código, foram contabilizadas as linhas de código necessárias para implementar o DCSSC. A Tabela 1 apresenta o número total das linhas de código, quantas delas foram criadas, quantas foram reutilizadas e a porcentagem equivalente à reutilização. Para implementar o DCSSC, foi adicionada uma nova biblioteca ao RCBM, que inicializa as leituras dos sensores. O número de linhas equivalentes à implementação do componente

<sup>1</sup>O código completo está disponível em [www.inf.ufpr.br/macarrero/SBCUP2018](http://www.inf.ufpr.br/macarrero/SBCUP2018).

que inicializa as leituras dos sensores foi inserido na contagem A como linhas novas e na contagem B como linhas reutilizadas. Considerando que o componente que inicializa as leituras dos sensores não existia no modelo RCBM e foi criado para o desenvolvimento deste estudo, seu código pode ser classificado como novo. Entretanto, levando-se em consideração implementações futuras, esse componente pode ser considerado reutilizável, justificando as linhas de sua implementação inseridas como linhas reutilizadas.

| Contagem | Total de linhas | Linhas reutilizadas | Linhas novas | Porcentagem de reuso |
|----------|-----------------|---------------------|--------------|----------------------|
| A        | 1333            | 905                 | 428          | 67,9%                |
| B        | 1333            | 954                 | 379          | 71,6%                |

**Tabela 1. Proporção de linhas reutilizadas do RCBM**

**CBO:** Na avaliação do coeficiente de dependência entre os componentes foi aplicada a métrica *Coupling Between Object Classes* (CBO), cujo valor está entre 0 e 1. Quanto mais próximo de 1 for o CBO de um certo componente, mais dependente ele é de outras classes. O resultado da avaliação é apresentado na Tabela 2. Percebe-se que o coordenador é altamente dependente de outros componentes, o que é esperado uma vez que ele é o responsável por fazer a junção de todos os componentes. Os componentes de biblioteca por sua vez não são dependentes de nenhum outro componente, e isso garante que seu uso seja independente da implementação do usuário. Os outros componentes utilizados possuem um baixo grau de acoplamento, favorecendo a depuração do código e a correção de erros nos módulos, além de facilitar a inclusão de novos componentes ao RCBM.

| Componente | Coordenador | Componentes de biblioteca | Sensor | CM   | CH   |
|------------|-------------|---------------------------|--------|------|------|
| <b>CBO</b> | 1           | 0                         | 0,33   | 0,33 | 0,33 |

**Tabela 2. Análise da métrica CBO para cada componente do modelo DCSSC**

## 6. Conclusão

Este trabalho apresentou uma proposta de máquina de estados para fazer o detalhamento do fluxo de operações em um simulador baseado em eventos. Através de um estudo de caso e a sua codificação no simulador NS2, mostrou-se que há uma correspondência direta entre a máquina proposta e o programa desenvolvido. Os experimentos demonstram que o percentual de reutilização de código utilizando o RCBM para a implementação do DCSSC foi de até 71,6%, considerando o componente de biblioteca como reutilizável, e de até 67,9%, considerando o componente de biblioteca criado como linhas de código novas. Os valores são significativos considerando a complexidade dos códigos de simulação para RSSFs. Como trabalhos futuros podem ser citados: o desenvolvimento de uma linguagem baseada em máquina de estados para a geração automática do coordenador, a realização de estudos experimentais adicionais envolvendo outras métricas e utilização da abordagem de reutilização em outros simuladores, como por exemplo o *Network Simulator 3* (NS3).

## Referências

Araújo, G., Tostes, A., de LP Duarte-Figueiredo, F., and Loureiro, A. (2014). Um protocolo de identificação e minimização de congestionamentos de tráfego para redes vei-

- culares. *XXXII Simp. Bras. de Redes de Computadores e Sistemas Distribuídos*, pages 207–220.
- Cañete, E., Chen, J., Díaz, M., Llopis, L., and Rubio, B. (2011). A service-oriented approach to facilitate wsn application development. *Ad Hoc Networks*, 9(3):430–452.
- Carrero, M. A., da Silva, R. I., dos Santos, A. L., and Hara, C. S. (2015). An autonomous in-network query processing for urban sensor networks. In *20th IEEE Symp. on Computers and Communications (ISCC)*, pages 968–973.
- Carrero, M. A., Musicante, M. A., dos Santos, A. L., and Hara, C. S. (2017). A reusable component-based model for WSN storage simulation. In *Proceedings of the 13th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 31–38.
- Cecílio, J. and Furtado, P. (2012). A state-machine model for reliability eliciting over wireless sensor and actuator networks. *Procedia Computer Science*, 10:422–431.
- Cremonesi, B. M., Vieira, A. B., Nogueira, M., and Nacif, J. A. (2017). Um protocolo de alocação dinâmica de canais para ambientes médicos sob múltiplas estações base. In *XXXV Simp. Bras. de Redes de Computadores e Sistemas Distribuídos*, pages 272–285.
- Desai, A., Gupta, V., Jackson, E., Qadeer, S., Rajamani, S., and Zufferey, D. (2013). P: safe asynchronous event-driven programming. *ACM SIGPLAN Notices*, 48(6):321–332.
- Filho, G. P., Ueyama, J., Faiçal, B. S., Guidoni, D. L., and Villas, L. A. (2015). Residi: Um sistema de decisão inteligente para infraestruturas residenciais via sensores e atuadores sem fio. In *XXXIII Simp. Bras. de Redes de Computadores e Sistemas Distribuídos*, pages 53–66.
- Furlaneto, S. S., dos Santos, A. L., and Hara, C. S. (2012). An efficient data acquisition model for urban sensor networks. In *13th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 113–120.
- Gonçalves, N. M., dos Santos, A. L., and Hara, C. S. (2012). Dysto-a dynamic storage model for wireless sensor networks. *Journal of Information and Data Management*, 3(3):147–162.
- Krämer, M., Bader, S., and Oelmann, B. (2013). Implementing wireless sensor network applications using hierarchical finite state machines. In *10th IEEE Int. Conf. on Networking, Sensing and Control*, pages 124–129.
- Le, T. D., Pham, N. D., and Choo, H. (2008). Towards a distributed clustering scheme based on spatial correlation in wsns. In *International Wireless Communications and Mobile Computing Conference*, pages 529–534.
- Malavolta, I. and Muccini, H. (2014). A study on mde approaches for engineering wireless sensor networks. In *40th EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA)*, pages 149–157.
- Muñoz, C. and Leone, P. (2017). A distributed event-based system based on compressed fragmented-iterated bloom filters. *Future Generation Computer Systems*, 75:108–127.
- Taherkordi, A., Johansen, C., Eliassen, F., and Römer, K. (2015). Tokenit: Designing state-driven embedded systems through tokenized transitions. In *2015 Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, pages 52–61.

## Patrocinador Diamante



**GOVERNO**  
DO RIO GRANDE DO NORTE

---

## Patrocinadores Bronze



## Apoio Financeiro



MINISTÉRIO DA  
EDUCAÇÃO



nic.br  
Núcleo de Interligação  
& Coordenação de  
Porto BR

cgi.br  
Comitê Gestor de  
Internet no Brasil