An IoT Proposal for the Irrigation Management Exploring Context Awareness

Rogério Albandes[1](✉), Roger Machado[2], João L. B. Lopes[3], Jorge Barbosa[4], and Adenauer Yamin[1,2]

[1] Catholic University of Pelotas, Pelotas, Brazil
rogerio.albandes@sou.ucpel.edu.br, adenauer.yamin@ucpel.edu.br
[2] Federal University of Pelotas, Pelotas, Brazil
rdsmachado@inf.ufpel.edu.br
[3] Federal Institute Sul-Rio-Grandense, Pelotas, Brazil
joaolopes@cavf.ifsul.edu.br
[4] Universidade do Vale do Rio dos Sinos, São Leopoldo, Brazil
jbarbosa@unisinos.br

Abstract. With the changes in technologies and the rapid consumption of natural resources by humans, it is vital to implement smart irrigation techniques to overcome the scarcity of water resources shortly. The amount of fresh water existing on the planet, the use for irrigation corresponds to about 70% of all water consumed in the world, Inadequate water management creates a number of problems, such as lower productivity, lower quality fruits, environmental damage, soil erosion and reduced soil air, which has been motivating the evaluation of alternatives for water supply agricultural crops. One solution that can be exploited is IoT, helping to achieve optimum water-resource utilization. Considering this scenario, this article presents the EXEHDA-SF proposal, which aims to explore IoT solutions and Context Awareness in the decision-making process, with the perspective of minimizing socioenvironmental impacts. For the evaluation of EXEHDA-SF, as an irrigation alternative, without the use of physical sensors, a prototype was developed that integrates open source IoT technologies with the EXEHDA middleware and explores a meteorological forecasting service. The results achieved were considered promising, reaching a success rate of approximately 94% regarding the irrigation decision.

Keywords: Internet of Things · Context Awareness · Watering automatic control · EXEHDA middleware

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and Fundação de Amparo à Pesquisa do Rio Grande do Sul (FAPERGS) - PqG.

A. Casaca et al. (Eds.): IFIPIoT 2019, IFIP AICT 574, pp. 71–85, 2020.
https://doi.org/10.1007/978-3-030-43605-6_5

1 Introduction

Water is considered the most important resource for maintaining life on Earth. Of the amount of fresh water existing on the planet, the use for irrigation corresponds to about 70% of all water consumed in the world. With the need to produce food and knowing the limitation of agriculture, producers seek measures that will allow them to produce more, being irrigation one of them [9].

The design of an irrigation system can be evaluated through several performance indicators, considering the scope of this article, it is possible to highlight its operational efficiency and the costs associated with the water consumed. Inadequate water management creates a number of problems, such as lower productivity, lower quality fruits, environmental damage, soil erosion and reduced soil air, which translates into a high socio-environmental impact. In this sense, the literature has pointed out that the use of irrigation systems has been harmed by inadequate decisions regarding the moment of irrigation [24].

In 2016, the total water demand for irrigation in Brazil was 969 thousand liters per second throughout the year, amount that according to the National Water Agency (ANA[1]), corresponded to 46.2% of all clean water used in Brazil in the period [5].

Excessive use of clean water resources around the globe has generated a need for its use in a smarter way. One solution that can be exploited is IoT, in which the solutions are based on the application specific sensors' data acquisition and intelligent processing.

IoT based smart irrigation management systems can help in achieving optimum water-resource utilization in the precision farming landscape [18]. Besides, we highlight the use of Context Awareness to support the irrigation decision, providing adaptable services. Context Awareness becomes an essential approach in IoT systems to select the most appropriated decision according to the contextual information that is relevant to the user or to change operational parameters of services in execution [29].

The operations of context-aware systems involve a significant amount of context information that needs to be: continuously collected; efficiently interpreted; quickly processed; disseminated to interested applications; stored in context repositories [3,28]. Therefore, to minimize the development efforts, in the literature, we can found approaches exploring middlewares.

The middlewares use can reduce the complexity of context-aware applications development by providing support for context acquisition, modeling, storage, and processing, among other aspects. The use of middleware can reduce the developer's attention from context-related concerns, allowing them to focus on defining business rules and developing application-specific functionalities [20,25].

Being motivated for this scenario, this article discusses a proposal of a system of Smart Farming (SF) based on IoT concepts, called EXEHDA-SF (Execution Environment for Highly Distributed Applications-Smart Farming). This

[1] http://www3.ana.gov.br/.

proposal explores Context Awareness to decide the need for irrigation, aiming to water reduction. Besides, our proposal uses EXEHDA middleware [21] under development in the G3PD/UCPEL (Parallel and Distributed Processing Research Group).

The proposal uses as a data source, the information provided by a weather forecasting service. Thus, EXEHDA-SF explores a logical sensor, without the use of physical sensors, minimizing installation, calibration, and maintenance efforts.

EXEHDA-SF was evaluated through a case study that explored prototyping of the proposal considering the software architecture of the EXEHDA middleware. In this case study, the EXEHDA-SF monitoring the climatic conditions of the region of Pelotas city, obtained from a public service of the forecast, and based on these data it evaluates the need for irrigation. The obtained results were very promising, reaching high levels of correctness when deciding to irrigation or not.

This article is organized in six sections. Section 2 discusses the context awareness design and the characteristics of the EXEHDA middleware. In the third section some related papers are discussed. Section 4 exhibits the proposal EXEHDA-SF, showing its main functions. In the fifth section the prototyping of the proposal is discussed, being addressed the hardware and software used and the evaluation that was done. Finally, the sixth section presents the final considerations and future work.

2 Background

This section introduces the main concepts associated with context awareness and EXEHDA middleware, highlighting the principal features explored in the EXEHDA-SF.

2.1 Context Awareness

Context is any information that can be used to characterize the situation of an entity (person, place, or object) considered relevant to the interaction between user and application, including the user and the application [8,16]. Context Awareness is the ability of a system to use context to provide relevant information and services to the user [8,17].

Some motivations for application of Context Awareness in computational systems are: assisting in the understanding of reality; facilitate the adaptation of systems; contribute to the process of transforming data into information; support the comprehension of events; help identify situations of interest.

Figure 1 presents a vision of the process involved to obtain context awareness, in which we can see that the construction of a context-aware system occurs mainly through four steps [20,25]:

- acquisition: refers to the process of monitoring and capturing contextual information. This step aims at abstracting from context-aware applications the

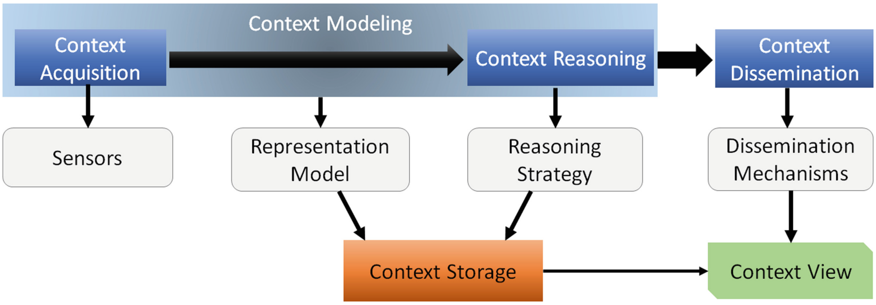

Fig. 1. Steps involved to obtain the context awareness.

complexity of data collection, enabling the reuse of sensors and the separation between obtaining and using contextual information [1];

- modeling: refers to the process of designing a model of real-world entities, their properties, the state of their environment and situations. The purpose of creating a context model is to provide a uniform, machine processable context representation scheme, facilitating context sharing and interoperability between different applications. The uniformity of the model between acquisition, reasoning, and utilization of context information is considered vital [17];
- distribution: refers to the step that allows the injection of context into the context-aware application and its delivery to all entities that have expressed any form of interest in this data [3];
- reasoning: can be defined as a method of deducing new knowledge, and better understanding, based on the available context [4]. It can also be explained as a process of providing knowledge deduction from a set of contexts [12].

In addition to the four steps that can be viewed at the top of the Fig. 1, we highlighted storage and view layers. Context storage is responsible for storing the acquired context data as well as for the contextual information that was inferred by the reasoning step. Context view provides data visualization methods in a way that facilitates access from the context-aware applications to captured data and detected situations.

Regarding the reasoning step, in the literature can be found different strategies for context reasoning, which have advantages and disadvantages considering the distinct domains of application [30]. Among these strategies, we highlight the rule-based, which is the most used strategy to perform the context reasoning in IoT applications [25].

2.2 EXEHDA Middleware

EXEHDA is a middleware designed to manage distributed, mobile, and context-aware applications, available from anywhere, anytime. The EXEHDA includes in its structure a core and services loaded on demand. The G3PD has explored the middleware on research fronts that consider the IoT challenges [31]. The main services provided are organized into subsystems related to Ubiquitous Access, Communication, Distributed Execution, Context Recognition, and Adaptation [2].

The IoT environment provided by EXEHDA is formed by multi-institutional equipment, being composite of user devices and equipment for infrastructure supporting. Each device is instantiated by their respective execution profile in the middleware, which implies on the need to adopt a cellular organizational management in this environment, aimed at ensuring the autonomy of the institutions involved [21].

Figure 2 shows an IoT environment managed by EXEHDA, wherein each cell, related to the provision of Context Awareness, is formed by a server Context Server, and by various Edge Servers and/or Gateways.

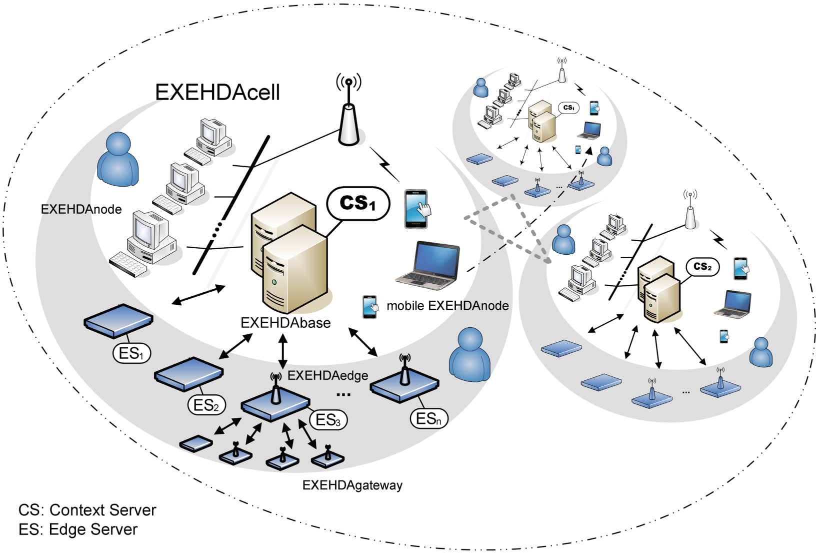

Fig. 2. IoT environment provided by EXEHDA

The gateways collect contextual information from physical or logical sensors, and have the purpose of treating the heterogeneity of the various types of sensors, in aspects of both hardware and protocol; transfer the collected information in a standard way to the Edge Servers.

In EXEHDA the processing of contextual information is distributed, remaining a part with the Edge Server, and another with the Context Server (see Fig. 3). The data received by the several Edge Servers are transmitted to the Context Server that manages them, and performs the storage and contextual processing steps. Context Server can combine the data from the Edge Servers with historical information, which are recorded in the Context Information Repository. A broader discussion about the different functionalities of both the Gateway and the Edge Servers is available in [31]. On the other hand, an approach of the different capabilities of the Context Server can be found in [21].

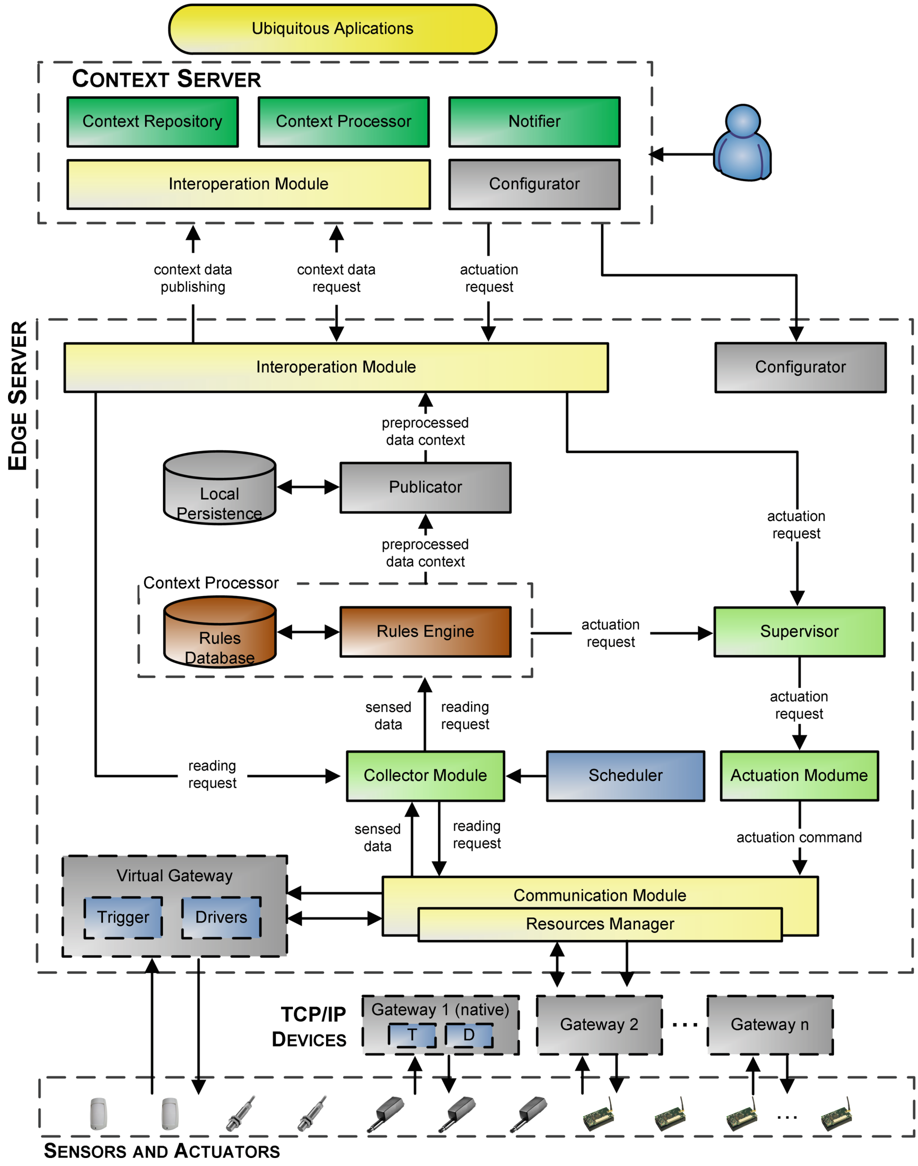

Fig. 3. Software architecture of EXEHDA to provide Context Awareness

3 Related Work

This section discusses papers related to the proposal EXEHDA-SF. A review of the literature was carried out, with the criteria for selection of the papers being its modernity and the use of methodologies associated with SF and IoT.

In paper [11] an intelligent irrigation architecture based on IoT is proposed, along with a hybrid approach based on machine learning to predict soil moisture. The proposed algorithm uses sensor data from the recent past and predicted meteorological data for soil moisture forecast for the following days.

The work [13] shows an automatic water supply system for agriculture using data from 2 sensors (soil moisture and light intensity). In addition, notifications are sent to the farm manager if there is a lack of water, and the interaction between the system and the manager is possible through the exchange of short message service (SMS) messages.

In the paper [18] a cloud-based framework and IoT are proposed to implement an intelligent irrigation system that preserves harvest during unforeseen rains, increases groundwater levels with a competent mechanism and reuses the excessive water generated during the rain for the irrigation of crops.

In the article [19] the linear regression algorithm is used, which helps in the prediction of the amount of water required for daily irrigation based on data provided by sensors scattered around the environment. The proposed system also reduces human and energy efforts. People can gain instant access to data through a mobile application.

The paper [6] uses field-scattered moisture sensors that are connected to a microprocessor (Arduino UNO). Whenever the humidity decreases below the plant wilting point, the sensors report to the system, which in turn triggers the irrigation pump, and when there is sufficient humidity, the shutdown command is sent.

Analyzing the five selected papers, among the several articles identified, it can be highlighted that EXEHDA-SF has two main differentials: (i) it does not use physical sensors, reducing installation, calibration and maintenance costs; and (ii) it explores Context Awareness in the decision of irrigation, considering as contextual variables the volume of recent previous precipitation, as well as the forecast of occurrence of precipitation in the following hours.

4 EXEHDA-SF: Organization and Functionalities

The organization of the software platform designed for EXEHDA-SF is presented in Fig. 4. After in this section, the functionalities of the different modules will be touched, and their operational profiles discussed.

4.1 Environment Interoperation Block

Environment Interoperation Block consists of the Forecast API Module, Communication Module and Actuator Module. This EXEHDA-SF block operates on a Native Gateway of middleware EXEHDA.

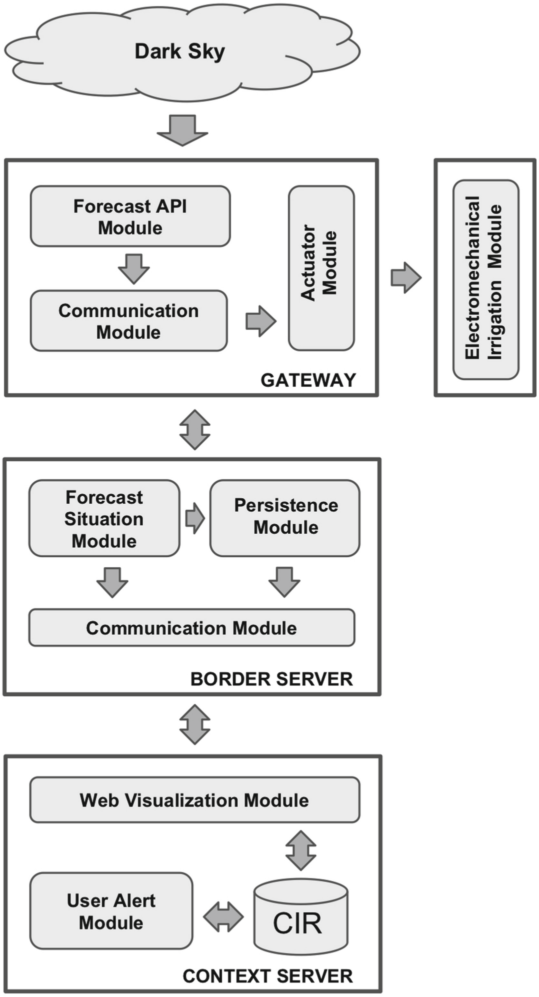

Fig. 4. EXEHDA-SF prototipation

Forecast API Module from the geographical coordinates of the city obtains the weather forecast data provided by the Dark Sky Service [7]. Its execution happens every hour, capturing the climatic data of the last 24 h, and in case it has rained, the volume of precipitated water is also captured. In addition, the climatological forecast is collected for the following 24 h, recording both the probability of occurrence of precipitation and the expected rainfall intensity. The Communication Module is responsible for transferring/receiving information and commands from the Edge Server. Finally, the Actuation Module manages the electromechanical device used to control the irrigation procedure.

4.2 Irrigation Control Block

Three modules form the Irrigation Control Block, which is instantiated on the EXEHDA Edge Server:

- Communication Module, which is responsible for interoperating with the User Interface Block, this functionality is instantiated in the Server Interoperation of the Edge Server;
- Persistence Module, which is intended to perform a temporary persistence if the Internet connection with the User Interface Block is lost. This functionality is instantiated on the Local Persistence module of the Edge Server;
- Forecast Situational Module, which is central to the EXEHDA-SF proposal, and is based on the Edge Server Rule Engine (see Fig. 3), and it has two main responsibilities that are detailed below.

Calculation of Probability of Precipitation (PoP)

It quantifies the possibility of precipitation occurrence in a given area over a specified period of time. The probability of precipitation (PoP) is defined by the product between two percentages:

- **C:** the confidence that precipitation will occur somewhere in the forecast area;
- **A:** the percentage of the area that will receive precipitation if it actually occurs.

Considering that there is 50% confidence (C) that a precipitation will occur and the expectation that measurable rain will occur around 80% of the area considered (A), the PoP will be 40%, that is, $\text{PoP} = \text{C} \times \text{A}$ (0.5×0.8) [14].

Algorithm for Irrigation Decision

In EXEHDA-SF the Irrigation Decision Algorithm is based on the moment of irrigation defined by the user and considers the rain data and its intensity, both from the previous 12 h and from the following 12 h. Based on the average values in the literature [15], the decision to irrigate at EXEHDA-SF will be considered as a 25 mm rainfall reference and as a 60% PoP.

The 25 mm reference for rainfall intensity is an international standard that consider the amount of rainwater that has accumulated in a given location over a period of time. In the International System of Units, the millimeter (mm) is used as the unit. A rainfall intensity of 1 m equals the accumulation of the volume of 1 liter (L) of rainwater over a surface area equal to 1 square meter [10]. An internationally accepted distribution is shown in Table 1.

As indicated in Fig. 5, if in the previous 12 h there was higher volume of precipitation than 25 mm, no irrigation will be triggered. Otherwise, the PoP will be considered for the next 12 h, if less than 60% it will be irrigated. If the PoP of the following 12 h is greater than 60% should be evaluated by the precipitation value; if less than 25 mm the irrigation must happen.

Table 1. Intensity of rain

Intensity (mm)	Description
<5 mm	Light rain
Between 5.1 mm and 25 mm	Moderate rain
Between 25.1 mm and 50 mm	Heavy rain
>50 mm	Violent rain

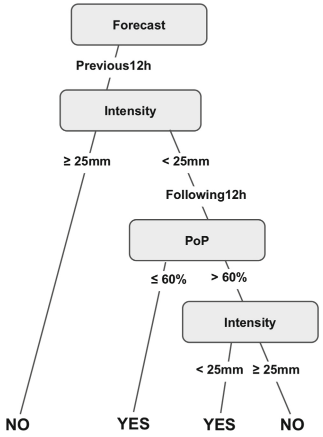

Fig. 5. Hierarchical decision structure

4.3 User Interface Block

User Interface Block was designed by exploring the functionality of the EXEHDA Context Server (see Fig. 3). This EXEHDA-SF block, besides providing information to users through the Internet, also includes persistence and notification functions.

- CIR Module is implemented in the Context Information Repository. Particularly on the relational database functionalities. All information regarding irrigation decision making is stored;
- Web Visualization Module provides the user with information on weather forecasts and irrigation decisions. The user is allowed to have the history records, with the possibility of indicating the occurrence and totals considering a certain period of time;
- User Alert Module scans the functions of the Notifier, and sends alerts to users using a public message sending platform, in this case Telegram.

5 EXEHDA-SF: Prototyping and Results

This section presents the decisions regarding the choice of hardware/software technologies for the EXEHDA-SF prototyping. The fact that the technology is open-source was the basic criterion for its selection.

5.1 Hardware Adopted

Gateway in the EXEHDA-SF architecture explores the ESP32 NodeMCU module (see Fig. 6a). This module consists of an IoT-oriented System-on-a-chip (SOC), formed by a 32-bit Tensilica Xtensa dual core microprocessor with built-in support for the Wi-Fi network (802.11) and bluetooth version 4.2, and with a 16 Mb integrated flash memory [22]. Its choice is because it supports multi-threading programming in Python, and because it is an open source hardware with excellent cost-benefit relation.

Edge Server in EXEHDA-SF explores Raspberry PI III b (see Fig. 6b). The used model offers network connection and for its operation an operating system called Raspbian was installed, which is a variant of the Linux's Debian distribution. The different functionalities of the Edge Server were designed through the Python Programming Language [27].

As an electromechanical actuator to control the water flow, the Hunter PGV-101G valve was used, which is supplied at 24 V and offers a flow rate ranging from 0.7 to 150 l/min (see Fig. 6c). This valve is triggered by a solenoid, and its choice was a consequence of both its large use in irrigation projects and its ease of interfacing with different microcontroller platforms.

5.2 Software Frameworks Adopted

Besides the software platforms used in the other modules of the EXEHDA middleware that are discussed in [21,31], the following software artifacts stand out for the design of EXEHDA-SF:

- **MicroPython:** besides implementing a selection of the main Python language libraries, MicroPython includes specific features for use with microcontrollers. In the case of the EXEHDA-SF Gateway, the "machine" library stands out, which is used to access different hardware features such as input and output ports [23].
- **Picoweb:** also in the Gateway of EXEHDA-SF a webserver, named Picoweb [26], was installed, under which algorithms were developed to interpret REST commands sent via the GET command by the Edge Server, using the JSON notation. Based on these functionalities, the Communication Module has been designed, through which it is possible to implement performance commands, particularly the activation of the Hunter PGV-101G valve, which activates the irrigation.
- **Dark Sky API:** it is an API that provides hourly weather conditions for a given region [7]. In EXEHDA-SF the Dark Sky API behaves as the source of the contextual data used to infer a possible irrigation situation.

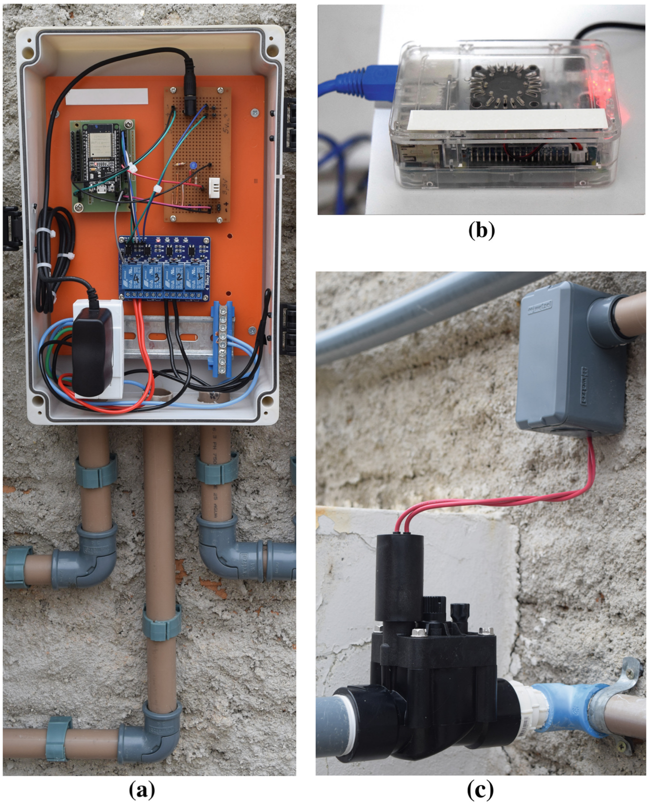

Fig. 6. Prototype developed for EXEHDA-SF evaluation

5.3 Evaluation

In order to evaluate the EXEHDA-SF, data about its operation were collected over a period of 90 days. Data generated at 6 am and 6 pm, on consecutive days, totaling 180 sensing procedures. The tests were carried out in the neighborhood Las Acacias in Pelotas, RS, which is part of the forecasting task offered by Dark Sky. Its API has coverage for areas with a radius of approximately 50 km. Table 2 shows the total errors that were observed. Precipitation volumes were estimated with the aid of a rain gauge.

In the universe of 180 decision making situations made by EXEHDA-SF, a total of 11 non confirmed cases by the observation of what actually occurred, which translates into a correctness rate of approximately 94%. These non confirmed cases happened due to errors in the forecast API. For example, inconsistent values regarding rain intensity of the previous 12 h, or incorrect predictions of probability of precipitation (PoP) and/or rain intensity of the following 12 h.

Table 2. Decision errors by EXEHDA-SF

Period of measurement	≥25 mm	<25 mm
Previous 12 h	2	1
Following 12 h (PoP ≥ 60%)	4	2
Following 12 h (PoP < 60%)	1	1

6 Conclusions

The study and research effort related to the conception and development of EXEHDA-SF pointed out that the combined exploration of Context Awareness and the Internet of Things is a promising way for computational systems to adjust their behavior regarding modifications in their contexts of interest.

On the other hand, it can be observed that the middleware-based approach can reduce the complexity of the development of context-aware applications by providing support for the acquisition, modeling, storage and processing of the context, among other aspects. Thus, the use of middleware may release developers of context-related concerns, allowing them to focus on defining the operational rules and developing the application-specific functionality.

The high percentage of clean water used in irrigation, in the case of Brazil, 46.2% in 2016, highlights the importance of optimizing irrigation procedures for world agriculture, with the aim of improving their efficiency, contributing to the farmers who seek for alternatives that provide better use of water to meet their water requirements.

From this perspective, this paper presented the EXEHDA-SF proposal, which aims to explore the Context Awareness in decision making for irrigation, seeking to minimize socio-environmental impacts. EXEHDA-SF aims to explore the Context Awareness in decision making for irrigation, seeking to minimize socio-environmental impacts. In addition, the proposal does not employ physical sensors, thus reducing installation, calibration and maintenance costs. In EXEHDA-SF the decision to perform irrigation is based on rainfall data and its intensity, both from the previous 12 h and from the following 12 h.

The proposal EXEHDA-SF was evaluated based on data collected and information regarding its operation over a period of 90 days. In this evaluation, a success rate of 94% was obtained in the decision making, achieving promising results pointing to the continuity of the research. Among the expected future work is the expectation of observing the behavior of EXEHDA-SF throughout the 4 seasons of the year, as well as to use a machine learning algorithm to optimize the parameters used for the irrigation decision, including the difference in water consumption with the use of EXEHDA-SF.

References

1. Alegre, U., Augusto, J.C., Clark, T.: Engineering context-aware systems and applications. J. Syst. Softw. **117**(C), 55–83 (2016). https://doi.org/10.1016/j.jss.2016.02.010
2. Augustin, I., Yamin, A., Silva, L.: Building a smart environment at large-scale with a pervasive grid middleware. In: Grid Computing Research Progress, pp. 182–186. Nova Science Publishers, Inc. (2008)
3. Bellavista, P., Corradi, A., Fanelli, M., Foschini, L.: A survey of context data distribution for mobile ubiquitous systems. ACM Comput. Surv. **44**(4), 24:1–24:45 (2012). https://doi.org/10.1145/2333112.2333119
4. Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D.: A survey of semantics-based approaches for context reasoning in ambient intelligence. In: Mühlhäuser, M., Ferscha, A., Aitenbichler, E. (eds.) AmI 2007. CCIS, vol. 11, pp. 14–23. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85379-4_3
5. Brasil: Governo do brasil (2018). http://www.brasil.gov.br/noticias/meio-ambiente/2018/03/. Último acesso 02 Dezembro 2018
6. Chavda, R., Kadam, T., Hattangadi, K., Vora, D.: Smart drip irrigation system using moisture sensors. In: ICSCET, pp. 1–4. IEEE (2018)
7. Darksky (2018). http://darksky.net/dev. Último acesso 2 Dezembro 2018
8. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001). https://doi.org/10.1007/s007790170019
9. Erthal, E.S., Berticilli, R.: Sustentabilidade: Agricultura irrigada e seus impactos ambientais. Ciência Tecnologia **2**(1), 64–74 (2018). http://revistaeletronica.unicruz.edu.br/index.php/CIENCIAETECNOLOGIA/article/view/6940
10. Glickman, T.S., Zenk, W.: Glossary of meteorology. American Meteorological Society (2000)
11. Goap, A., Sharma, D., Shukla, A., Rama Krishna, C.: An IoT based smart irrigation management system using machine learning and open source technologies. Comput. Electron. Agric. **155**, 41–49 (2018). https://linkinghub.elsevier.com/retrieve/pii/S0168169918306987
12. Guan, D., Yuan, W., Lee, S., Lee, Y.: Context selection and reasoning in ubiquitous computing. In: The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), pp. 184–187, October 2007. https://doi.org/10.1109/IPC.2007.102
13. Imteaj, A., Rahman, T., Hossain, M.K., Zaman, S.: IoT based autonomous percipient irrigation system using Raspberry Pi. In: 2016 19th International Conference on Computer and Information Technology (ICCIT), pp. 563–568. IEEE (2016)
14. Joslyn, S., Nadav-Greenberg, L., Nichols, R.M.: Probability of precipitation: assessment and enhancement of end-user understanding. Bull. Am. Meteorol. Soc. **90**(2), 185–194 (2009)
15. Kamienski, C., et al.: SWAMP: an IoT-based smart water management platform for precision irrigation in agriculture. In: 2018 Global Internet of Things Summit (GIoTS), pp. 1–6. IEEE (2018)
16. Khattak, A.M., et al.: Context representation and fusion: advancements and opportunities. Sensors **14**(6), 9628–9668 (2014). https://doi.org/10.3390/s140609628
17. Knappmeyer, M., Kiani, S., Reetz, E., Baker, N., Tonjes, R.: Survey of context provisioning middleware. IEEE Commun. Surv. Tutor. **15**(3), 1492–1519 (2013). https://doi.org/10.1109/SURV.2013.010413.00207

18. Koduru, S., Padala, V.G.D.P.R., Padala, P.: Smart irrigation system using cloud and Internet of Things. In: Krishna, C.R., Dutta, M., Kumar, R. (eds.) Proceedings of 2nd International Conference on Communication, Computing and Networking. LNNS, vol. 46, pp. 195–203. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1217-5_20
19. Kumar, A., Surendra, A., Mohan, H., Valliappan, K.M., Kirthika, N.: Internet of Things based smart irrigation using regression algorithm. In: International Conference on Intelligent Computing, Instrumentation and Control Technologies (2017)
20. Li, X., Eckert, M., Martinez, J.F., Rubio, G.: Context aware middleware architectures: survey and challenges. Sensors **15**(8), 20570 (2015). https://doi.org/10.3390/s150820570
21. Lopes, J.L., de Souza, R.S., Geyer, C., da Costa, C., Barbosa, J., Pernas, A.M., Yamin, A.: A middleware architecture for dynamic adaptation in ubiquitous computing. J. Univ. Comput. Sci. **20**(9), 1327–1351 (2014)
22. Node MCU (2018). https://nodemcu.readthedocs.io/en/dev-esp32/. Último acesso 15 Dezembro 2018
23. Micropython (2018). https://micropython.org. Último acesso 2 Dezembro 2018
24. Penteado, S.R.: Manejo da Á gua e irrigação – em propriedades ecológicas. Via Orgânica (2010)
25. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the Internet of Things: a survey. IEEE Commun. Surv. Tutor. **16**(1), 414–454 (2014). https://doi.org/10.1109/SURV.2013.042313.00197
26. Picoweb (2018). https://github.com/pfalcon/picoweb. Último acesso 3 Dezembro 2018
27. Raspberry (2018). https://www.raspberrypi.org/. Último acesso 15 Dezembro 2018
28. Sahu, C.G., Adane, D.S.: A survey on context-aware middleware. Int. J. Adv. Res. Comput. Commun. Eng. **4**(5), 650–656 (2015). https://doi.org/10.17148/IJARCCE.2015.45138
29. Sánchez Guinea, A., Nain, G., Le Traon, Y.: A systematic review on the engineering of software for ubiquitous systems. J. Syst. Softw. **118**(C), 251–276 (2016). https://doi.org/10.1016/j.jss.2016.05.024
30. Sezer, O.B., Dogdu, E., Ozbayoglu, A.M.: Context-aware computing, learning, and big data in Internet of Things: a survey. IEEE Internet Things J. **5**(1), 1–27 (2018). https://doi.org/10.1109/JIOT.2017.2773600
31. Souza, R., Lopes, J., Geyer, C., Cardozo, A., Yamin, A., Barbosa, J.: An architecture for IoT management targeted to context awareness of ubiquitous applications. J. Univers. Comput. Sci. **24**(10), 1452–1471 (2018)