IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 17, 2020, accepted January 26, 2021, date of publication February 24, 2021, date of current version March 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062094

Internet of Water Things: A Remote Raw Water Monitoring and Control System

ABÍLIO C. DA SILVA JÚNIOR[1], ROBERTO MUNOZ[2], MARÍA DE LOS ÁNGELES QUEZADA[3], (Member, IEEE), ALOÍSIO V. LIRA NETO[1], MOHAMMAD MEHEDI HASSAN[4], (Senior Member, IEEE), AND VICTOR HUGO C. DE ALBUQUERQUE[5,6], (Senior Member, IEEE)

[1]Department of Computer Science, Universidade de Fortaleza, Fortaleza 60750740, Brazil
[2]Escuela de Ingeniería Informática, Universidad de Valparaíso, Valparaíso 2362905, Chile
[3]Instituto Tecnológico de Tijuana, Tijuana 22414, Mexico
[4]Research Chair of Smart Technologies, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[5]Department of Computer Science, Federal Institute of Education, Science and Technology of Ceará, Fortaleza 60750740, Brazil
[6]Department of Teleinformatics Engineering, Federal University of Ceará, Ceará 60020181, Brazil

Corresponding author: Mohammad Mehedi Hassan (mmhassan@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research at King Saud University through the Vice Deanship of Scientific Research Chairs: Research Chair of Smart Technologies.

ABSTRACT The scarcity of the planet's water resources is a concern of several international entities and governments. Smart solutions for water quality monitoring are gaining prominence with advances in communication technology. This work's primary goal is to develop a new online system to monitor and manage water resources, called Internet of Water Things (IoWT). The proposed system's objective would be to control and manage raw water resources. Thus, it has developed a platform based on the server-less architecture and Internet of Things Architectural Reference Model, in which it is applied in a simulation environment, considering several electronic devices to validate its performance. For this research, there is a system for capturing raw water from tubular wells. Each well has a level sensor, a temperature sensor and a rain gauge. The data is collected every minute by an electronic device and sent every hour to the IoWT system. From data analysis, the amount of memory allocated to functions minimally interferes with efficiency. The IoWT system, applied in a real case, consists of connecting a device installed in a water well to the platform, where the data is transmitted through a 3G network and then processed. Thus, the proposed approach has great potential to be considered a complementary tool in monitoring raw water and assisting in decision-making for the management of water resources.

INDEX TERMS Automation and control, raw water, sensors, water resources.

I. INTRODUCTION

The United Nations (UN) and the United Nations for Education, Science and Culture (UNESC) have been intervening with the shortage of hydrological resources of the planet and its catastrophic effect upon agriculture and human consumption [1]. Researchers have examined the Internet of Things (IoT), considering that it is an infrastructure of the global red and is highly adjustable based on communication protocols in which the physical things have a virtual identity as well as intelligent interphases capable of binding with other information systems [2], [3]. Combining with other computer techniques such as Big Data, Artificial Intelligence and Machine Learning [4]–[6], IoT has been presented as a very promising alternative for divers types of application such as the tracking of environmental and rural resources [7]–[9], urban [10]–[13], industrials [14]–[16] and of health [17]–[19].

The authors in [20] designed one prototype in such a way that it can monitor the number of pollutants in the water. Multiple sensors are used to measure various parameters to assess the quality of water from water bodies. Other research focuses on developing an automation system on a Water Treatment Plant (WTP) using the Raspberry Pi and adafruit.io cloud server as the MQTT server. In this proposed system, Raspberry Pi has a function to facilitate the computational

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

process and communication connection to the internet network [21]. In [22] proposes a prototype system design, implementation and description of required tools and technologies to develop IoT-based water level monitoring system which can be implemented in future smart villages. It is necessary to manage water quality to fulfil the sustainability of water functions as a natural resource. They create an integrated system based on IoT to measure the water quality by developing environmental water management monitoring system using sensors [23]. This research presents a detailed overview of recent works carried out in the field of smart water quality monitoring. Also, a power efficient, simpler solution for in-pipe water quality monitoring is based on IoT technology is presented. The model developed is used for testing water samples and the data uploaded over the Internet are analyzed [24].

Since IoT is a technology that is being increasingly adapted, we are seeing an increasing number of devices, software, and services designed to meet the demand of IoT usage.

There is a great effort in the research and development of hardware, software and protocols used to develop products or services that are intrusive in the IoT paradigm, but that still need to be refined, such as protocols [25]–[28], security [29]–[31], energy consumption [32], [33] and architectural patterns [34], which correspond to the vital part of an information system.

Given the abovementioned situation and combining the capacity to carry out the massive acquisition of data, the IoT is an alternative for the environmental control of water, as a tool to support decision making, resource management, risk management, analysis environmental behavior of the user. For instance, the control of resources to evaluate the quantity and quantity of water in aquifers [35], the monitoring of infrastructure [36], disaster forecasting, forecasting the demand and consumption of water resources [37]. Fortino *et al.* [38] had proposed a model and a simulation system for different IoT infrastructures successfully applied and validated in water monitoring. However, few studies are found in the literature that propose to build an Internet of Things system based on a standardized architecture, as well as a set of requirements created thinking of such a system and using a simulation methodology or tests capable of guaranteeing performance of the IoT application for efficient management of water resources.

Within the scope of global water scarcity, this work aims to model, build and simulate an IoT-based computer system to control and manage water resources (raw water) by monitoring "smart" objects (smart devices). This system is named as the Internet of Things of Water (IoTW).

In the early stages of this work, a series of questions were asked that could answer how IoT technology can be used for water management. The research questions are then predicted.

RQ01 How is an IoT architecture model to monitor the water resources developed that can minimally satisfy the requirements of this type of application?
RQ02 What components of IoT reference architecture should be used for the development of an IoTW system?
RQ03 Which architectural evaluation methods can be used to evaluate the architecture of an IoT system developed for water management?
RQ04 What technologies can be used to develop the prototype of this system?
RQ05 Does the Component x Technologies meet a basic number of requirements for IoT systems? And can you evaluate this same set using architectural evaluation methods?

To answer the questions mentioned above, this work used the DSR (Design Science Research) methods [39]. The present work follows the cycle of said methodology, as shown in Figure 1, where the problem presented and the possible solutions for it were investigated at the beginning. Among them, a candidate solution to the problem was identified and an analysis was carried out before the implementation phase. After evaluating the solution, the proposed solution was implemented and tested.

This article is organized as followed. In Section 2, the experimental methods and procedures are presented. The results and discussion are presented in Section 3. Finally, Section 4 presents the conclusions on the proposed system, the contributions to research, the limitations and future work.

II. METHODOLOGY

The investigation presented was based on the DSR methodology, as illustrated in Figure 1, which was developed in five different stages:

1) Investigation of the problem, handled in sections II-A, II-B; 2) Design of the solution, managed in section II-C; 3) Validation of solutions, handled in section II-E1; 4) Implementation of the solution, handled in sections II-D, II-E and 5) The Evaluation of the solution, discussed in section III.

In this work, the initial research for the development of this project is developed. For the purposes of this work, we will treat our problem as a system for capturing raw water from tubular wells. Each well has a level sensor, a temperature sensor and a water meter. The data is collected every minute by an electronic device and sent every hour. A premise used by the current monitoring body which we think is appropriate to use in this article. The idea is that for the sake of saving data on traffic and as a result of the price for operating the system, the devices only sent data every hour and not every minute. To the IoTW system, water resources in the region are processed and stored, as can be seen in figure 2. The system called IoTW is the main objective of study of this work. In the following points, it is explained how the proposed system was developed.

FIGURE 1. Cycle of design science research.

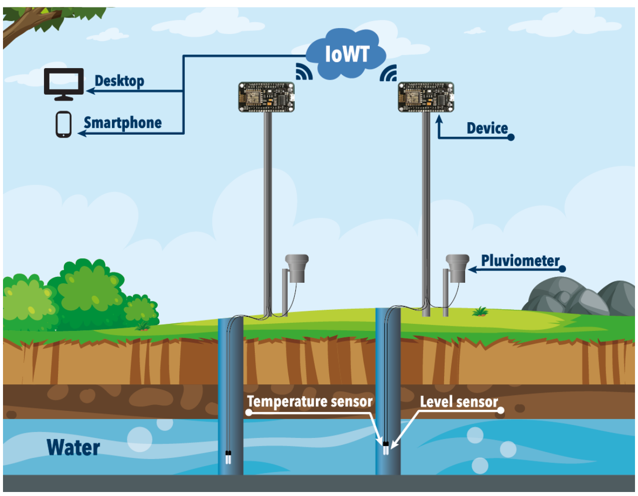


FIGURE 2. High level IoWT architecture.

A. REQUIREMENTS

The main requirements gathering activities were carried out on the basis of the specialized literature. Common requirements were raised for systems that use the IoT paradigm and from these, the most relevant ones were chosen for our research. Table 1 shows the number of requirements found in [3]. This table also classifies the functional and non-functional requirements as already indicated in the aforementioned table and which will form a part of the main needs of our system.

B. ENABLING TECHNOLOGIES

Based on the context and requirements presented above, we move on to the system design phase. The system must be completely built with a cloud-oriented architecture and IoT, which can scale the number of devices, with processing capacity and data storage in the back-end.

TABLE 1. List of requirements.

TYPE	ID	Description	IoWT
RF	R01	Resource Discovery	✓
	R02	Resource Management	✓
	R03	Data Management	✓
	R04	Event Management	✓
	R05	Code Management	✓
RNF	R06	Scalability	✓
	R07	Real-time or Timeliness	X
	R08	Reliability	X
	R09	Availability	X
	R11	Security	X
	R10	Privacy	X
	R12	Deployment, maintenance, and use	X
	R13	Interoperability	X
	R14	Spontaneous interaction	X
	R15	Multiplicity	X
	R16	Adaptability and Flexibility	X

1) COMMUNICATION

It is important that the communication between the devices and the servers where the information is processed and stored. It follows a device standard in the cloud, where these communication devices are intermediary with the cloud through the Internet. Here IoTW must encapsulate everything in the described layers. in IoT-ARM. For our use case, packets with sensor data will be sent from the IoTW devices.

2) PROTOCOLS

All of the devices, physical and simulated, communicate using the Message Queuing Telemetry Transport Protocol (MQTT). This has been widely used in IoT applications. It is a protocol that requires a message agent provided by the communication means mentioned above. The system will also use security protocols for data transmission through MQTT. Transport Layer Security (TLS) will be used for data encryption and JavaScript object notation web algorithms (*JSON Web Algorithm* (RFC 7518)) for authentication based on public/private keys.

C. IOWT- HIGH-LEVEL IOWT ARCHITECTURE

From the data collected with previous data, IoTW was designed. Based on the identified requirements, we map what the necessary requirements for our context are and which layers of the reference model would meet these requirements. This assignment is shown in Table 2.

The set of elements that interact with each other is called the reference architecture of the IoTW system here. Still in Table 2, the column is directly connected to the requirements of the functional layer. Each layer is standardized by the architecture reference and in this way, we hope to give the certainty that the developed architectural framework is based on the main architecture reference.

The functional architecture of IoTW has new layers in which each component is contained inside these layers. The architecture, as you can see, is based in the architecture of IoT-ARM. The only difference is the interface that access the layer of serviced in IO and virtual entity. In our architecture, both layers can only be accessed hierarchically. The architecture can be seen in figure 3.

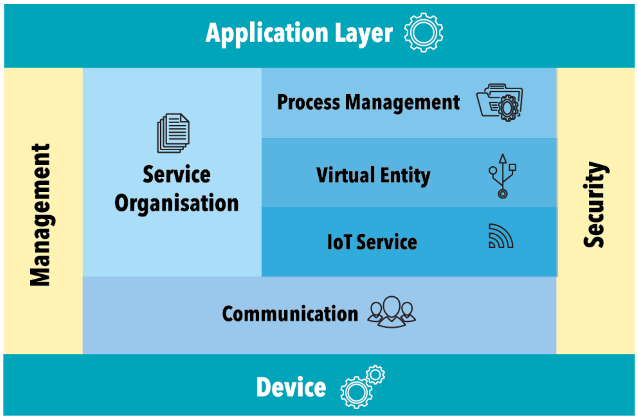


FIGURE 3. Internet of water things architecture.

The description of layers and components refers to respectively to the same IoT-ARM and the instance of system to be built later, the latter, we will take care of in time. For now, it is necessary understand that when the system is developed, it develops only the components they sought to fulfill the requirements raised.

D. IMPLEMENTATION

If we check again the reference architecture, an instance creation of our IoTW system was developed. The application development platform was chosen. What is sought with this is an infrastructure that can be scaled automatically and that meets the satisfaction of the cloud computing approaches. Therefore, the IoTW was implemented using the middleware infrastructure,[1] the use of the tools offered by this platform meets the proposed requirements.

The IoTW prototype is the implementation of the reference architecture using middleware resources. Next, two functions are developed. One of them implements a layer service called IoT service. This function receives a notification from the communication layer and performs data storage. This has been called *IoTService:RecebeTemetria*.

The Virtual Entity layer was also implemented using a serverless function (*VirtualEntity:RecuperaDados*) that through a request for layer process makes a call to the data retrieval function and thus we can access the telemetry data from the sensors. A high-level view of the system to be developed can be seen in Figure 4. In the figure that represents our use case, the devices perform continuous measurements of the water parameters that are sent to the middleware, where the data is processed and they store, and can be viewed from there by an application.

E. VALIDATION

1) ARCHITECTURE EVALUATION METHODS

There are several methods to evaluate the software architectures proposed in the literature [27], these are grouped according to the objectives, the quality requirements, the phase of the project where the method is applied and what artifacts are necessary to carry out the analysis of the method.

The choice of the method used in this work was chosen based on the context of our application. In this work, we will validate the architecture using the Scenario-Based Architecture Reengineering (SBAR) method. This was chosen for the advantages that it is already used in telemetry systems and also for the purpose of evaluating multiple requirements.

In this way, we can validate the architecture using a method that I use in the requirements mentioned in section IIA. The SBAR method consists of three activities: (1) Scenario definition, (2) Architecture analysis and (3) Summary of results. These activities can be carried out interactively and an implemented system, prototype, simulation or mathematical model is required for the execution of the methods.

2) PROTOTYPES VALIDATION

The prototype was validated according to the evaluation methods discussed in the previous section. In this order, a model was developed that simulates the context for which the prototype was developed. The model will not be validated, since it is not the objective of this work. Nor does it intend to fully fill the possible scenarios of water management systems, but rather it is realistic enough to validate the prototype of our architecture.

The simulated context obeys algorithm 1 that creates several tasks on the computer. Each created task performs an algorithm that simulates a device as can be seen in algorithm 2. Each simulated device connects and sends randomly generated data to the prototype of the IoTW system. Due to the repetitive nature of our algorithms, they have a complexity of $\theta(n^2)$.

Algorithm 1 Multitasking Algorithm

input : Number of messages per MSG device,
Number of devices D,
number of messages m
output: File F with published data and errors

//Initialization of variables;
MSG = 600;
D = 500;

for $i \leftarrow 1$ **to** D **do**
  `Simulation` (i, MSG);

3) DATA ACQUISITION AND TREATMENT

Each message sent by the simulated device generates an event. This event triggers the processing of the message, which consists of keeping the simulated values of the sensors in the database. The IoW system registers the execution time of each event for subsequent analysis.

[1]Google IoT Core

TABLE 2. IoWT architecture.

FUNCTIONAL GROUP	FUNCTIONAL COMPONENTS	DESCRIPTION IoWT	REQUIREMENT
IoT Process Management	Process Modeling	x	
	Process Execution	x	
Service Organisation	Service Orchestration	x	
	Service Composition	x	
	Service Choreography	Middleware's tools	Several
Virtual Entity	VE Resolution	Developed using middleware FaaS and application Front-End	Resource Discovery
	VE & IoT Service Monitoring	Developed using middleware FaaS and Application Front-End	Resource Management
	VE Service	x	
IoT Service	IoT Service	Developed using middleware FaaS, application Front-End and Storage PaaS	Data management Service, Event Management, Code Management
	IoT Service Resolution	x	
Communication	Hop To Hop Communication	x	
	Network Communication	x	
	End To End Communication	Middleware's tools	
Securty	Authorization	Developed using middleware FaaS and application Front-End	Security
	Key Exchange & Management		Security
	Trust & Reputation	x	
	Identity Management	Middleware's tools	Security, Privacy
	Authentication	Developed using middleware FaaS e application Front-End	Security, Privacy
Management	Fault	Developed using middleware FaaS and application Front-End	Security, Deployment, maintenance and use
	Configuration	Middleware's tolls	Deployment, maintenance and use
	Reporting	Middleware's tools	Deployment, maintenance and use
	Member	Developed using middleware FaaS and application Front-End	Security, Privacy
	State	Middleware's tolls	Deployment, maintenance and use

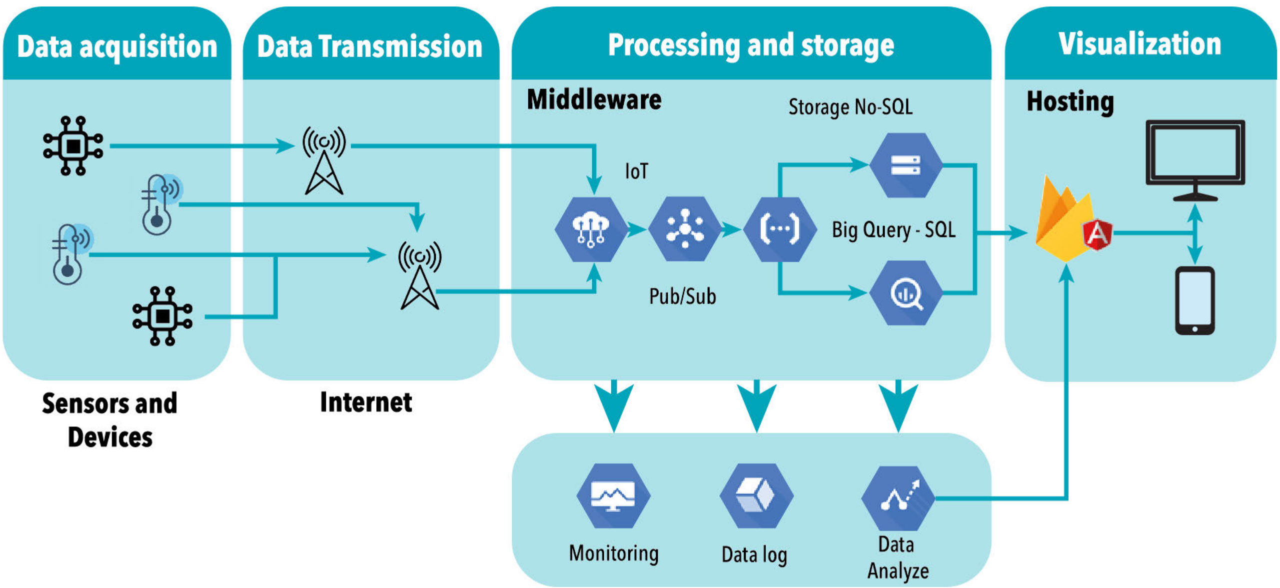


FIGURE 4. IoWT domain model.

The execution time of each event was the variable studied here. This time can vary for a number of reasons. Each of these reasons can be studied separately. In this work, we are interested in whether the system can carry out all events without errors or if the IoTW system will give indications that it will not support a greater workload.

After the data was acquired, these were summarized. We present the minimum, maximum, the means calculated with equation 1, the standard deviation calculated according to equation 2, the median calculated according to equation 3 and the first and third quartiles. Calculated with the equation 4 and 5.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}; \tag{1}$$

$$s = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1}}; \tag{2}$$

Algorithm 2 Simulation

input : Connection data to server A,
Device identification d,
number of messages MSG.
output: Error prone message *error*

//Variable initialization;

$d \leftarrow$ `Connect` (A);
if *!(d is conected)* **then**
  return *error*;
  for $j \leftarrow 1$ **to** m **do**
    payload($[j]$)$\leftarrow$ `RandomData` ();
    $F \leftarrow F \,\|$`Publish` $(d_i,$ **payload**($[j]$));
else
  `CloseConnect`
(d);

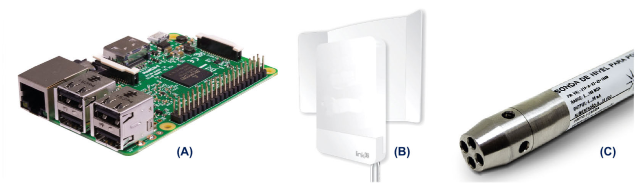


FIGURE 5. Used devices a) Raspberry, b) Modem, c) sensor.

$$x = \begin{cases} x_{(n+1)/2} & \text{if } n \text{ is even} \\ \frac{1}{2}(x_{n/2} + x_{(n/(2+1))}) & \text{if } n \text{ for is odd} \end{cases} \quad (3)$$

$$Q_1 = x_{\frac{x}{4}}; \quad (4)$$

$$Q_3 = x_{\frac{3x}{4}}; \quad (5)$$

As explained by a high value of the studied variable, it could indicate a fragility of the system, generating errors in unexpected telemetry events, which would result in data loss. In this way, we decided to extract and analyze the outliers of our data. For the extraction of outliers, the interquartis amplitude formula 6 was used where IIA is the amplitude given by $IIA = Q_3 - Q_1$. Since we are concerned with only the timeout of the function here, we focus on higher outliers only.

$$\text{upper outliers} = Q_3 + 1,5 * IIA; \quad (6)$$

4) USE CASE METHODS

We tested the prototype in a real environment and used a Raspberry Pi 3 device as shown in Figure 5 a) with the same algorithm adopted in the simulation. A 3G Modem from the company Elsys[2] was also included to provide Internet access to the proposal Figure 5 b).

To finalize, in Figure 6, we can see a schematic of how the system was prepared for the evaluation.

[2]https://www.elsys.com.br/

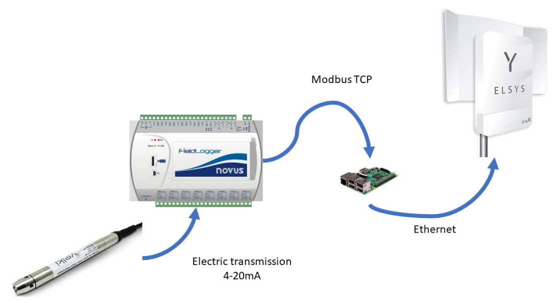


FIGURE 6. Schematic use case.

III. RESULTS AND DISCUSSION

In this section, the results obtained from the analysis is carried out on the architecture and prototype of the solution as described in section II will be discussed. We evaluated the architecture according to the SBAR method described above and the prototype had its evaluated performance of based on memory used by serverless functions.

The architecture analysis was performed analyzing each requirement and examined whether it is being satisfied by the prototype. The results of the evaluation of the artifact architecture can be seen in Table 5.

Here, it was evaluated that the code administration requirement partially meets. The simulation software used did not allow the IoTW prototype to have this proven functionality. This functionality should be tested in a real environment, because this will be done in future work.

The prototype was subjected to the events generated by the algorithm presented in section II. The algorithm was run on an Intel i7 8 GB memory computer, Windows 10 operating system. An explanatory table with the different experiments can be seen in the Table 6.

For the experiments performed and discussed here, no errors in event processing of any kind were detected. NO errors have been detected, such as processing time limit or device connectivity errors, so they will not be part of this discussion.

The variables studied were summarized in Table 3. The data described in the table were the minimum, maximum, mean, median, first quartile, and third quartile standard deviation (SD) values. In this table, we can see that expression number 01 obtained the highest processing time values for the same data, but the location measurements as the variability measurements presented higher values than the others.

Variation in processing time was expected. but it can also be seen in the same table that the values of the other experiments were little changed. Corroborating the values of experiment number 02 is enough to satisfactorily satisfy our context. Table 4 shows the outliers of all the experiments. In this table we observe an absolute number of outliers during the experiments and the percentage in relation to the number of messages sent to the IoWT system.

TABLE 3. Results summary.

Experiments	Min	Max	Avg,	SD	Median	1st Quartile	3tr Quantile
Configuration 01	66.0	5374.0	317.5	159.3341	284.0	235.0	360.2
Configuration 02	41.0	5805.0	157.5	164.0322	131.0	103.0	171.0
Configuration 03	38.0	4814.0	122.6	131.0918	104.0	87.0	123.0
Configuration 04	41.0	4340.0	114.6	88.36988	104.0	88.0	123.0
Configuration 05	40.0	7692.0	116.2	114.4186	103.0	86.0	122.0

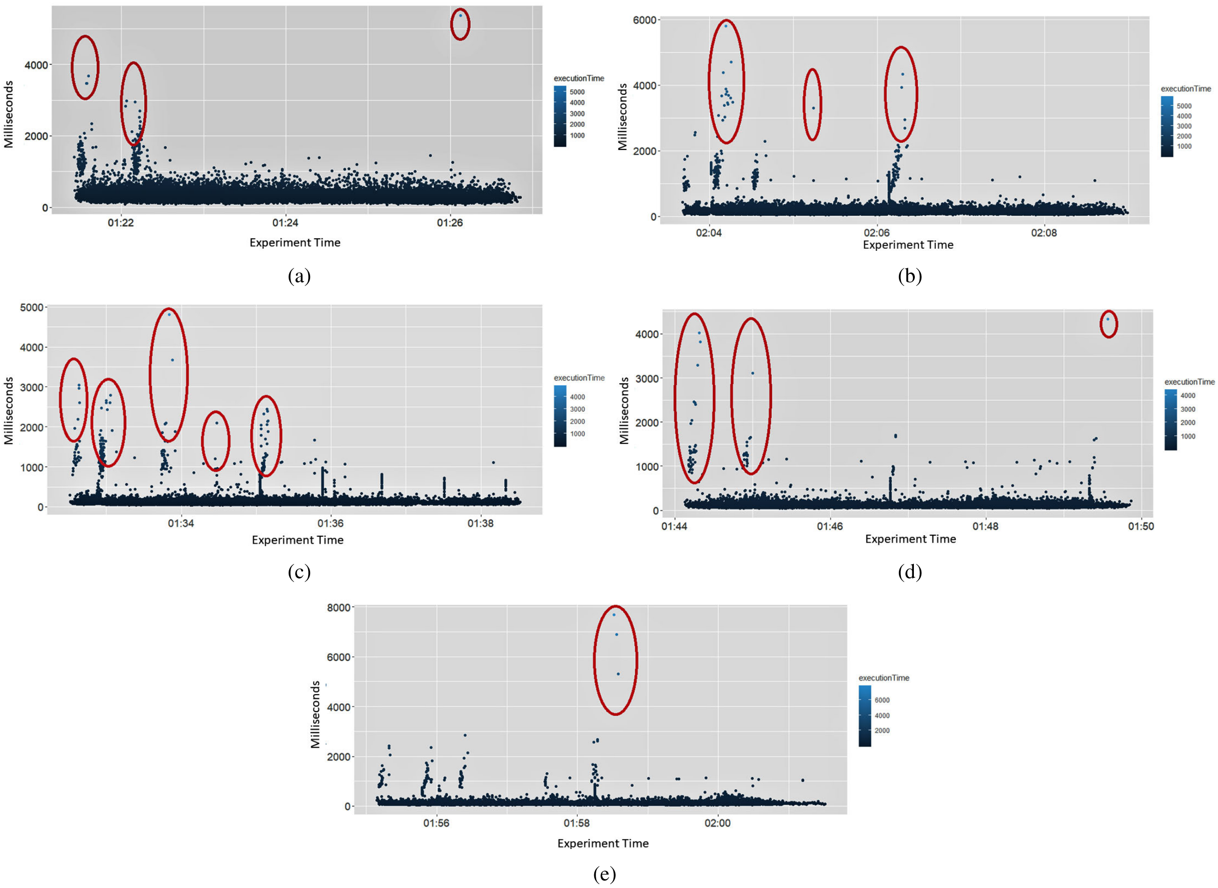


FIGURE 7. Configuration performance evaluation a) 1, b) 2, c) 3, d) 4, e) 5.

It is observed that there is a certain constancy in the percentages of outliers that our method identified, corroborating only the fact that regardless of the memory used in our system, it can meet the requirements for the number of devices and data expected in our context.

Since all the experiments presented a similar amount of outliers, the experiments were extracted and presented, already without the outliers in figure 8 as a graph. Here, the box-type graph was chosen, and it may be possible to check the maximums, minimums, medians, and the first and third quartiles of the data. The performance indextextco is the time required to process the events (30000 events per device) sent to the server. The changed variable is only the memory for processing and the measured variable was the execution time in milliseconds for processing each event. The use of the box-type graph helped to visually check the average, maximum and minimum values of each experiment.

This graph represents well what was said previously. Experiment 01 is remarkably distant from the others. Here is a visual similarity between experiments 03, 04, and 05. All events were placed on the dot plot shown in Figure 7. In these charts, there is a general take of the execution time of all events processed by the system. Here, the distortions of each graph were marked with a red circle.

The points in graphs (a) and (b) are widely scattered at the base of the graph, which is expected because they have the highest standard deviation values between experiments. They also have the highest maximum run time values (also marked in red). Although technically not a problem of any kind, it is a strong indication of a possible system scalability problem.

TABLE 4. Outliers in the processing of experiments.

Experiments	Outliers	% of sample
01	1473	4.91
02	1445	4.81
03	1778	5.92
04	1545	5.15
05	1583	5.27

FIGURE 8. Experiments performance evaluation.

TABLE 5. Software evaluation.

REQUERIMENTS	ACCOMPLISH
Resource Discovery	Yes
Resource Management	Yes
Data Management	Yes
Event Management	Yes
Code Management	Yes with caveats
Scalability	Yes

TABLE 6. Experiments summary.

Experiments	Memory	Devices	Messages	Events
01	128 MB	500	60	30000
02	256 MB	500	60	30000
03	512 MB	500	60	30000
04	1 GB	500	60	30000
05	2 GB	500	60	30000

The other experiments have little or no advantage over each other. Although experiment number 03 presented several observations along its graph (in red), the data from this experiment does not show any significant disadvantage compared to experiments 04 and 05, including having better minimum and maximum values than experiment 05.

Even though a variation in processing time is normal for any computer system, we observe a very large variance here in all experiments. This variation occurs primarily due to a property of systems that use serverless architectures called cold-start.

Cold-start is a property of Function-as-a-ServiceFaaS where, for various reasons, instances of new instances of created functions are created. These new functions are instantiated when we first run it in our experiments and when the FaaS provider scales due to the number of events started.

Therefore, although this behavior may cause the function execution timeout to be observed and the processing may fail, this behavior is what guarantees great scalability for our system. For the IoTW system, where data is sent periodically and there is no need for real-time data processing, this specific behavior is not a system limit.

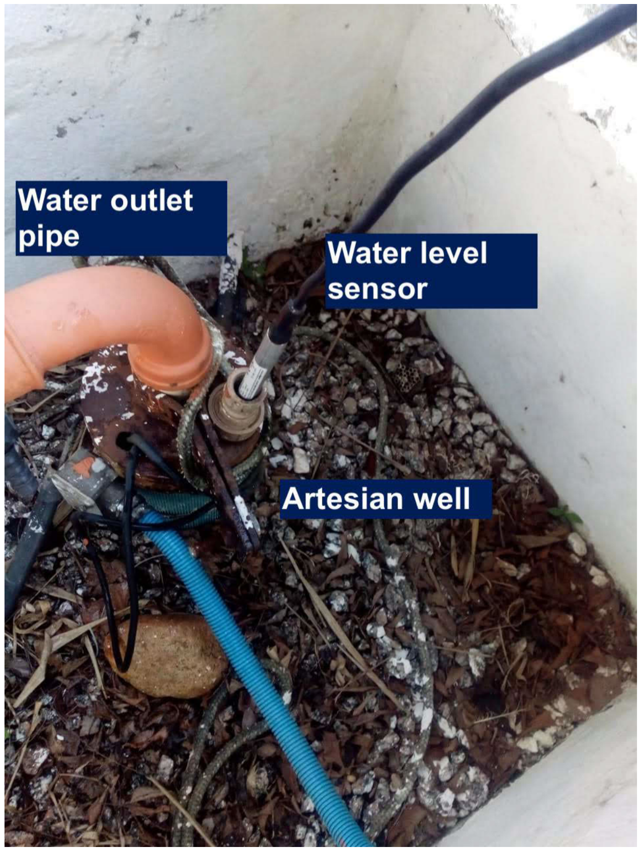


FIGURE 9. Developed prototype in a real environment.

For other specific systems or for a system that has a larger amount of data, this statement may not be entirely true and is not part of the study conducted here.

A. CASE STUDY

The prototype was tested in a monitoring system of deep tubular wells installed in protective tubes (Figure 9) where water levels are measured and stored in electronic equipment every hour. The sensor used was a piezoelectric level sensor, a data storage device and a 3G communication modem.

The data that resulted from this last experiment are the values of the levels measured and processed by the developed prototype and can be seen in the Figure 10 and Figure 11. The first graph shows a month of data transmitted to the system in its entirety. The graphs show the distance from the top of the well to the water depth. You can see that there is a cut at the top of the graph and another at the bottom. The constancy in this cut shows what the sensor is getting out of the water due to the lowering of the well and the cut at the bottom of the graph shows the dynamic level of the well.

In the daily graph (Figure 11), it is easier to check the dynamic level of the well, which varies with time. We found that the cut at the top of the graph is constant, due to the level sensor being out of the water for a more accurate

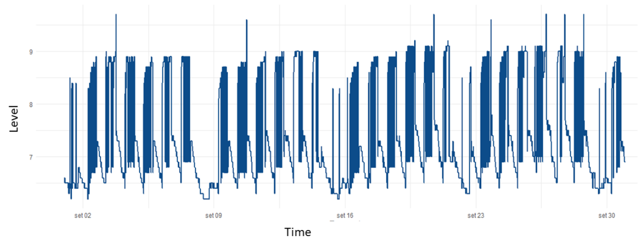


FIGURE 10. **Well monthly graph.**

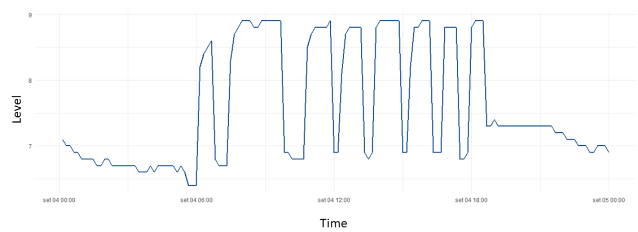


FIGURE 11. **Well daily graph.**

measurement. In this way, it would be necessary to lower the sensor closer to the water. There were no processing failures in the developed prototype, although it may have occurred between the device and the system prototype due to the communication failures of the 3G network.

This system can be used for data acquisition for monitoring purposes, decision-making regarding the use of water, prediction of levels or even use of water and any others that require data collected as input parameters. As shown in the article, there is an optimum amount of resources for using the system. This amount of resources refers to the memory used for processing, a key parameter for making the price of this type of solution. The use of this architecture can be then used with a low cost of implementation and operation.

The results presented here show that the architecture developed on the basis of the reference architecture is sufficient to satisfy most of the basic requirements of the studied context. The results of the evaluation of our implemented artifact acting in our context have demonstrated that within this, our architecture requires few hardware resources and can be rapidly developed using market tools.

The main contribution of this work is to demonstrate that IoT computational systems can be developed to help the management of water resources reliably, using a standardized reference architecture that can be understood by several researchers and developers, in addition to presenting a cheap and efficient alternative. We hope that the system as shown here can be used by researchers and by society as an alternative and inexpensive way of acquiring data, and with this, assist these institutions to predict water scarcity in a region, verify losses, fraud, guarantee quality water, and so on.

IV. CONCLUSION

This work developed and tested architecture for IoT systems for efficient and scalable monitoring of water resources. For validation purposes, we used the architecture validation methods and developed an algorithm that simulates the data sent to test the architecture prototype. This has been tested in various scenarios. The contributions of the work are derived directly from the answers obtained for the research questions listed above.

The contributions of the work come directly from the answers obtained for the research questions listed above.

RQ01 *How to develop an IoT architecture model to monitor water resources that can meet a minimum range of requirements?* To define this architecture, we identify a set of architectures, along with some requirements and from there we built our architecture. The validation of this stage of the work was given by the criteria adopted in section II-C.

RQ02 *Should the components of an IoT reference architecture be used for the development of an IoWT system?* We present the components of the architecture developed in accordance with the proposed research. In each section II, we raise the hardware and software components for the development of the IoWT system.

RQ03 *What architectural evaluation methods can be used to evaluate the architecture of an IoT system developed for water management?* No satisfactory methods were found for evaluating a specific IoT system. Analogous methods were then used and had to be adapted to evaluate the architecture, as seen in section II-E1.

RQ04 *What technologies can be used to deflect the prototype of this system?* In the section II-D, a series of technologies used for the development of the system prototype were presented.

RQ05 *Does the x Technologies component pair meet a basic number of requirements for IoT systems? And can this same set be evaluated using architectural evaluation methods?* The requirements were successfully met and assessment method architectures can be used with warnings.

It's also worth mentioning that the prototype could be tested more extensively. Although the simulated number of sensors for the presented context was sufficient, it would not be sufficient to monitor the measurement of a treated water supply system in a city with millions of inhabitants.

Analyzing only the processing time of the functions may also have introduced a method bias and this has masked the need to evaluate other requirements. An example of this could have been security tests on the system for security breaches. Due to the complexity of this type of tests, this effort will be reserved for exclusive works for this purpose.

The architecture that was tested in this experiment can be used as a simple means to monitor water resources of any nature, but the simulation and case study were built with groundwater monitoring in mind. A prototype was designed which was validated according to the proposed architecture. It was observed that some of the experiments showed very little variation in the processing time. The results presented here demonstrate that the architecture developed on the basis of the reference architecture is sufficient to satisfy most of the basic requirements of the context studied.

As a future project, we intend to implement the system for operation in a real environment, using real physical devices and not just simulations. In this manner, we would be able to verify the operation of the system in production using real data over a long period of time. With the real data from the environment, we hope to develop an algorithm that can predict demand and consumption needs.

REFERENCES

[1] R. Connor, A. Renata, C. Ortigara, E. Koncagül, S. Uhlenbrook, B. M. Lamizana-Diallo, and D. Brdjanovic, "The United Nations world water development report 2017. Wastewater: The untapped resource," United Nations World Water Develop. Rep., 2017.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[3] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, and V. H. C. de Albuquerque, "A reference model for Internet of Things middleware," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 871–883, Apr. 2018.

[4] M. Aly, F. Khomh, Y.-G. Gueheneuc, H. Washizaki, and S. Yacout, "Is fragmentation a threat to the success of the Internet of Things?" *IEEE Internet Things J.*, vol. 6, no. 1, pp. 472–487, Feb. 2019.

[5] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of big data and machine learning in smart grid, and associated security concerns: A review," *IEEE Access*, vol. 7, pp. 13960–13988, 2019.

[6] C. Wang, G. Yang, G. Papanastasiou, H. Zhang, J. Rodrigues, and V. Albuquerque, "Industrial cyber-physical systems-based cloud IoT edge for federated heterogeneous distillation," *IEEE Trans. Ind. Informat.*, early access, Jul. 7, 2020, doi: 10.1109/TII.2020.3007407.

[7] A. M. Elmisery, M. Sertovic, and B. B. Gupta, "Cognitive privacy middleware for deep learning mashup in environmental IoT," *IEEE Access*, vol. 6, pp. 8029–8041, 2018.

[8] F. Montori, L. Bedogni, and L. Bononi, "A collaborative Internet of Things architecture for smart cities and environmental monitoring," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 592–605, Apr. 2018.

[9] B. Keswani, A. G. Mohapatra, A. Mohanty, A. Khanna, J. J. P. C. Rodrigues, D. Gupta, and V. H. C. de Albuquerque, "Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms," *Neural Comput. Appl.*, vol. 31, no. S1, pp. 277–292, Sep. 2018.

[10] L.-M. Ang, K. P. Seng, A. M. Zungeru, and G. K. Ijemaru, "Big sensor data systems for smart cities," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1259–1271, Oct. 2017.

[11] J. Santos, T. Vanhove, M. Sebrechts, T. Dupont, W. Kerckhove, B. Braem, G. V. Seghbroeck, T. Wauters, P. Leroux, S. Latre, B. Volckaert, and F. D. Turck, "City of things: Enabling resource provisioning in smart cities," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 177–183, Jul. 2018.

[12] K. Pardini, J. J. P. C. Rodrigues, O. Diallo, A. K. Das, V. H. C. de Albuquerque, and S. A. Kozlov, "A smart waste management solution geared towards citizens," *Sensors*, vol. 20, no. 8, p. 2380, Apr. 2020.

[13] A. H. Sodhro, S. Pirbhulal, Z. Luo, and V. H. C. de Albuquerque, "Towards an optimal resource management for IoT based green and sustainable smart cities," *J. Cleaner Prod.*, vol. 220, pp. 1167–1179, May 2019.

[14] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[15] Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps, "Industrial Internet of Things: A systematic literature review and insights," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4515–4525, Dec. 2018.

[16] P. H. F. Sousa, N. M. M. Nascimento, J. S. Almeida, P. P. R. Filho, and V. H. C. Albuquerque, "Intelligent incipient fault detection in wind turbines based on industrial IoT environment," *J. Artif. Intell. Syst.*, vol. 1, no. 1, pp. 1–19, 2019.

[17] J. H. Abawajy and M. M. Hassan, "Federated Internet of Things and cloud computing pervasive patient health monitoring system," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 48–53, Jan. 2017.

[18] P. Verma and S. K. Sood, "Fog assisted-IoT enabled patient health monitoring in smart homes," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1789–1796, Jun. 2018.

[19] C. Arcadius Tokognon, B. Gao, G. Y. Tian, and Y. Yan, "Structural health monitoring framework based on Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 619–635, Jun. 2017.

[20] A. Jerom B., R. Manimegalai, and R. Manimegalai, "An IoT based smart water quality monitoring system using cloud," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE)*, Feb. 2020, pp. 1–7.

[21] R. H. Laluma, R. Giantara, B. Sugiarto, C. A. Siregar, and S. Risnanto, "Automation system of water treatment plant using raspberry Pi.3 model B+ based on Internet of Things (IoT)," in *Proc. IEEE 13th Int. Conf. Telecommun. Syst., Services, Appl. (TSSA)*, Oct. 2019, pp. 72–76.

[22] T. Malche and P. Maheshwary, "Internet of Things (IoT) based water level monitoring system for smart village," in *Proc. Int. Conf. Commun. Netw.*, N. Modi, P. Verma, B. Trivedi, Eds. Singapore: Springer, 2017, pp. 305–312.

[23] R. P. N. Budiarti, A. Tjahjono, M. Hariadi, and M. H. Purnomo, "Development of IoT for automated water quality monitoring system," in *Proc. Int. Conf. Comput. Sci., Inf. Technol., Electr. Eng. (ICOMITEE)*, Oct. 2019, pp. 211–216.

[24] S. Geetha and S. Gouthami, "Internet of Things enabled real time water quality monitoring system," *Smart Water*, vol. 2, no. 1, pp. 1–19, Dec. 2016.

[25] I. Tomic and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1910–1923, Dec. 2017.

[26] N. Javaid, A. Sher, H. Nasir, and N. Guizani, "Intelligence in IoT-based 5G networks: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 94–100, Oct. 2018.

[27] L. Oliveira, J. Rodrigues, S. Kozlov, R. Rabêlo, and V. Albuquerque, "MAC layer protocols for Internet of Things: A survey," *Future Internet*, vol. 11, no. 1, p. 16, Jan. 2019.

[28] M. A. A. da Cruz, J. J. P. C. Rodrigues, P. Lorenz, P. Solic, J. Al-Muhtadi, and V. H. C. Albuquerque, "A proposal for bridging application layer protocols to HTTP on IoT solutions," *Future Gener. Comput. Syst.*, vol. 97, pp. 145–152, Aug. 2019.

[29] H. Derhamy, J. Eliasson, and J. Delsing, "IoT Interoperability—On-demand and low latency transparent multiprotocol translator," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1754–1763, Oct. 2017.

[30] M. M. Hassan, M. R. Hassan, S. Huda, and V. H. C. de Albuquerque, "A robust deep learning enabled trust-boundary protection for adversarial industrial IoT environment," *IEEE Internet Things J.*, early access, Aug. 24, 2020, doi: 10.1109/JIOT.2020.3019225.

[31] S. Pirbhulal, W. Wu, K. Muhammad, I. Mehmood, G. Li, and V. H. C. de Albuquerque, "Mobility enabled security for optimizing IoT based intelligent applications," *IEEE Netw.*, vol. 34, no. 2, pp. 72–77, Mar. 2020.

[32] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive IoT data and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1457–1477, 3rd Quart., 2017.

[33] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: An energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.

[34] G. Gardašević, M. Veletić, N. Maletić, D. Vasiljević, I. Radusinović, S. Tomović, and M. Radonjić, "The IoT architectural framework, design issues and application domains," *Wireless Pers. Commun.*, vol. 92, no. 1, pp. 127–148, Oct. 2016.

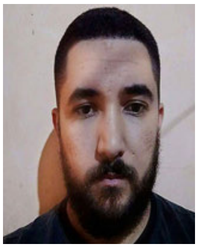
[35] J. Pan, Y. Yin, J. Xiong, W. Luo, G. Gui, and H. Sari, "Deep learning-based unmanned surveillance systems for observing water levels," *IEEE Access*, vol. 6, pp. 73561–73571, 2018.

[36] R. Martac, N. Milivojevic, V. Milivojevic, V. Cirovic, and D. Barac, "Using Internet of Things in monitoring and management of dams in serbia," *Facta Universitatis Ser., Electron. Energetics*, vol. 29, no. 3, pp. 419–435, 2016.

[37] M. K. Dhobale, P. S. Gorgile, J. P. Gunjal, A. K. Hirve, and U. A. Prof Mande, "IOT based smart water supply management system," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 6, no. 1, pp. 297–299, Jan. 2017.

[38] G. Fortino, R. Gravina, W. Russo, and C. Savaglio, "Modeling and simulating Internet-of-Things systems: A hybrid agent-oriented approach," *Comput. Sci. Eng.*, vol. 19, no. 5, pp. 68–76, 2017.

[39] J. R. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*. Berlin, Germany: Springer, 2014.



ABÍLIO C. DA SILVA JÚNIOR received the B.Sc. degree in information systems from the Ceará Estácio University Center in 2014 and the master's degree in applied informatics from the University of Fortaleza in 2020. He works with Automation and Control Systems, based on Internet of Things for water resource management and control.

ROBERTO MUNOZ received the master's degree in computer engineering, engineering science, and education, and the Ph.D. degree in computer engineering. He is currently the Head of the School of Informatics Engineering, Universidad de Valparaíso. He has authored over 90 scientific articles. His research interests include computers and education, multimodal learning analytics, human–computer interaction, and health informatics. He is a TPC member in refereed international conferences and highly reputed journals.

MARÍA DE LOS ÁNGELES QUEZADA (Member, IEEE) received the master's degree in computer science and the Ph.D. degree in computer science. She is currently a Professor with the Department of Systems and Computation, Technological Institute of Tijuana. Her research areas are computers and education, human–computer interaction, and health informatics.

ALOÍSIO V. LIRA NETO received the degree in public safety management from the University of Southern Santa Catarina. He is currently pursuing the master's degree in applied informatics with the University of Fortaleza. He is currently a Special Agent with the Federal Highway Police. He has experience in strategic intelligence, intelligent systems, pattern recognition, artificial intelligence, and sociology, with emphasis on violence, criminality, and police action.

MOHAMMAD MEHEDI HASSAN (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2011. He is currently a Professor with the Department of Information Systems, College of Computer and Information Sciences, King Saud University (KSU), Riyadh, Saudi Arabia. He has authored and coauthored more than 260 publications, including refereed journals (more than 218 SCI/ISI-indexed journal articles, more than four ESI highly cited papers, and one hot paper), conference articles, books, and book chapters. His research interests include cloud/edge computing, Internet of Things, artificial intelligence, body sensor network, big data, mobile computing, cyber security, smart computing, 5G/6G network, and social network. He was a recipient of several awards, including the Best Conference Paper Award from CloudComp Conference in 2014, the Excellence in Research Award from CCIS, KSU, in 2015 and 2016, the Best Journal Paper Award from the IEEE Systems Journal in 2018, the Distinguished Research Award from CCIS, KSU, in 2020, and the Best Conference Paper Award from the IEEE International Conference on Sustainable Technologies for Industry 4.0 in 2020. He is listed as one of the top 2% scientists of the world in the networking and telecommunication field.

VICTOR HUGO C. DE ALBUQUERQUE (Senior Member, IEEE) received the degree in mechatronics engineering from the Federal Center of Technological Education of Ceará, the M.Sc. degree in teleinformatics engineering from the Federal University of Ceará, and the Ph.D. degree in mechanical engineering from the Federal University of Paraíba. He is a specialist, mainly, in Internet of Things, machine/deep learning, pattern recognition, and robotics.