A Lightweight Framework for IoT Smart Solutions

Javier Ortiz-Hernandez
The National Technological Institute of Mexico/CENIDET
Cuernavaca, México
javier.oh@cenidet.tecnm.mx

Juan Antonio Miguel-Ruiz
The National Technological Institute of Mexico/CENIDET
Cuernavaca, México
m20ce065@cenidet.tecnm.mx

Manuel Erazo-Valadez
The National Technological Institute of Mexico/CENIDET
Cuernavaca, México
m20ce059@cenidet.tecnm.mx

Leon Torres-Restrepo
The National Technological Institute of Mexico/CENIDET
Cuernavaca, México
leontorres16c@cenidet.edu.mx

Alicia Martinez-Rebollar
The National Technological Institute of Mexico/CENIDET
Cuernavaca, México
alicia.mr@cenidet.tecnm.mx

Manuel Mejia-Lavalle
The National Technological Institute of Mexico/CENIDET
Cuernavaca, México
manuel.ml@cenidet.tecnm.mx

***Abstract*— Successive technological improvements have made possible the development of IoT with a high capacity for communication and data collection, providing numerous opportunities for application. As a result, attention from researchers in both academic and industrial environments has been drawn to the subject matter. Despite all the enhancements till date, there are still several open issues that represent the main challenges for IoT. One of these concerns resides in the management of the devices responsible for collecting environmental data. The functionality and reliability of the services it provides depends on the correct management of the sensor devices. In this article we propose a framework for the development of a lightweight IoT smart solution using the FIWARE Open-Source Platform. This solution scheme is intended for an indoor localization system using Wireless Fidelity (WiFi) and Bluetooth Low Energy (BLE) signals.**

Keywords*— *IoT, lightweight, smart solutions, beacons, FIWARE

I. Introduction

The emerging trends in embedded technologies and Internet has enabled a massive interconnection that lingers behind human interaction. The increase in connected devices and device to device communication has been referred to as Internet of Things (IoT) and sometimes as Internet of Everything (IoE). The number of connected devices is currently estimated at 30 billion, and according to the specialized blog IoT World Online, it is expected to increase up to 75 billion in 2025. In addition, global spending on the Internet of Things will grow at an annual rate of 13.6% during the forecasted period of 2017-2022 [1]. The common aspect in all of these terms is the connection of new kinds of objects to the Internet in order to build a connected world. We can envision a future where Internet-based devices will be invisibly implanted around us, constantly generating enormous amounts of data. These will be gathered, processed, and presented in a simple and understandable form, generating big data, and the special software systems will give information and novel services to their users by analyzing it in servers' side [2].

However, we can observe the continuous and growing offer of IoT solutions in various sectors of economics, socials and even entertainment interests. In most cases, turnkey and scalable solutions are offered, but with very limited configuration margins [3]. Perhaps the most important problem is that these types of solutions are generally very limited to interoperate with other applications that are of interest to us. We consider this situation as a limitation for the expansion of IoT solutions, which are capable of integrating with other information subsystems in an effective and efficient way [4], [5].

Cloud computing platforms deliver virtually endless capabilities as well as specialized services that meet the various IoT demands [3]. Amazon AWS, IBM Watson, Microsoft IoT Suite and Oracle Intelligent Applications have been helpful in the development of such services and applications. In particular, they offer remote management mechanisms, but they are typically specialist software tools that embrace only certain proprietary devices [6], [7], [8]. This can cause the spreading of different management tools, a scenario that creates difficulties in the administration and sometimes increases the costs of the IoT ecosystems. This heterogeneity is a problem for effective device management due to the complexity of obtaining a single, federated management approach. A federated strategy is one that follows a standard corporate architecture aggregating services centrally and characterized by the interoperability of resources and agile service integration [4].

II. IoT for lightweight smart solutions

Although, as mentioned above, there is an extensive growth in the development of IoT applications. We observe that the majority of them come from the main cloud providers which offer a wide range of tools for the development of horizontal applications [9], [10]. In all cases, it is a very wide and specialized offer that seeks to cover the needs of the industrial, commercial and consumer sectors, sometimes hand in hand with technological partners. Also, most of these providers offer documented use cases and very didactic user guides that aim to facilitate the selection of services. In the case of light IoT applications, we consider this as a great advantage to have this commercial alternative. At the same time, we promote the

This work has been partially funded by the TecNM project 8819.20-P

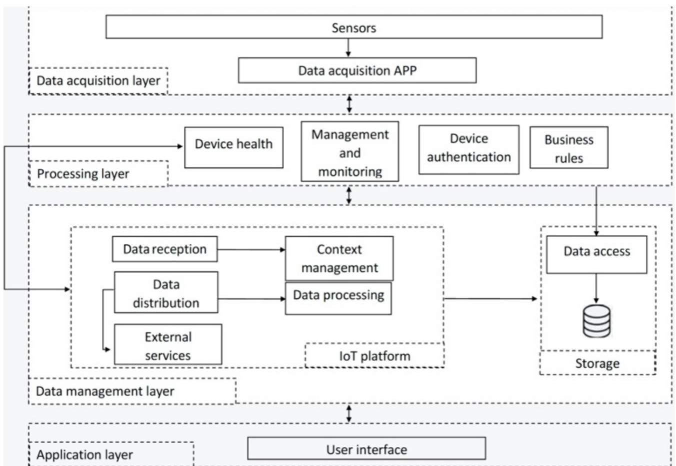
XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

development of open architectures based on free software, which has the following advantages: a) low-cost solutions based on available resources and in open IoT platforms b) customized solutions that allow personalizing their interfaces and optimizing their performance according to their own parameters, and c) didactic prototypes that seek to promote technological entrepreneurship, in particular to solve public interest problems [6], [7], [8].

III. A LIGHTWEIGHT IOT FOR SMART SOLUTIONS FRAMEWORK

This section describes the framework that is proposed for the development of IoT applications. The structure is shown in Figure 1, in which the four layers that make up the framework are observed.

FIGURE 1. THE LIGHTWEIGHT IOT FRAMEWORK PROPOSED



Each layer is explained below:

Data acquisition layer: consists in a specialized mobile application to perform data acquisition. The application must be configurable and capable of receiving heterogeneous data from various devices and sensors and even various communication protocols. In other words, it must allow the framework to interact with different types of IoT devices without having the need for all sensors to handle the same protocol. The application would be the intermediary between the sensors and the data processing layer.

Data processing layer: It is composed of different modules that integrate a REST service through the HTTP protocol. Its main objectives are to manage the previously collected information and to maintain integrity over the data that will be sent to the data management layer. The internal modules of the data acquisition layer are the following:

• Device health: This module is in charge of monitoring the battery and the functional status of the sensor devices from which information is being collected.

• Management and monitoring: Module in charge of managing the different devices and users registered in the storage system.

• Device authentication: This module is in charge of verifying and taking care of monitoring the information registered in the previous module. Authentication is completed through a request to the devices.

• Business rules: It is a module that exploits the data obtained from the sensing devices and provide the functionality that is shown in the application layer. The inference engine must be able to generate useful information from the raw data received.

Data management layer: It is responsible for the storage of all the information captured in the previous modules, as well as the distribution of the information to the application layer and the processing layer. This can be composed of multiple databases. You can have one or more IoT components, which facilitate the management of context data and the management of timelines.

Application layer: It is responsible for graphically presenting the collected information, and through its user interface, manage the devices and show their status interacting with the data management and processing layers.

IV. THE FIWARE IOT OPEN SOURCE PLATFORM

FIWARE has the mission of promoting, augmenting, protecting, and validating the FIWARE technologies as well as the activities of the FIWARE community, empowering its members including end-users, developers and the rest of the stakeholders in the entire ecosystem. The FIWARE Foundation is open, anybody can join contributing to transparent governance of FIWARE activities and based on their merits, rise through the ranks. It is supported for the FIWARE Foundation, a non-profit organization that guides the definition and encourages the adoption of open standards (implemented using open-source technologies) that ease the development of smart solutions across domains such as Smart Cities, Smart Energy, Smart Agri Food and Smart Industry, based on FIWARE technology [11].

FIWARE brings a curated framework of open source platform components which can be assembled together and with other third-party platform components to build Smart Solutions faster, easier and cheaper. It is an open-source API platform, which, to date, has released 42 standardized software components aimed at helping startups and enterprises build the next generation of smart applications and services for cities, industries, e-health or agribusiness. It offers a set of tools for different functionalities and it is also an innovation ecosystem for the creation of new applications and internet services. It is especially useful in terms of Smart Cities, as it ensures the interoperability and the creation of standard data models [12].

The platform provides enhanced OpenStack-based Cloud capabilities and a set of tools and libraries known as Generic Enablers (GEs) with public and open-source specifications and interfaces. The use and management of data coming from "things" (i.e., sensors, actuators and other devices) is also a complex process, as there are many different protocols in the IoT sphere, but FIWARE provides a set of GEs allowing access to the relevant information through only one API (NGSI). It allows the reading of sensor information, while also the operation of some elements. Therefore, the Orion Context Broker is an essential part in the management of the entire lifecycle of context information including updates, queries, registrations and subscriptions. Using the Orion Context Broker, you are able to create context elements and manage them through updates and queries. In addition, you can subscribe to context information so that when some condition occurs (e.g., the context elements have changed) you receive a notification.

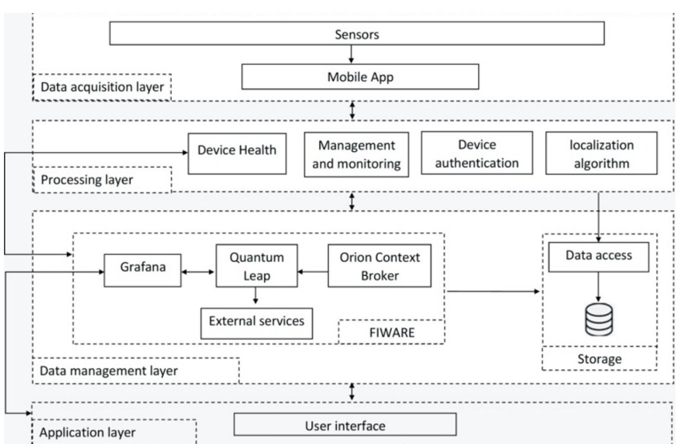
Other functionalities such as business intelligence, Web interfaces and advanced interfaces allow the creation of very powerful applications and solutions [12].

V. A CASE STUDY

Next, we present an experimental study that seeks to localize people in an interior space, using as input the intensity of the signal received from the Bluetooth Low Energy (BLE) beacons as well as the WiFi signal of the equipment installed inside of the buildings. It is an application that monitors the elderly by reading the activity in the interior of a determined space. This information will permit the identification and prediction of various symptoms of physical and mental health.

The development of this system was carried out following the architecture proposed in point III and adapted to the features and tools offered by the FIWARE platform mentioned in point IV [11], [12]. The selection of these tools was carried out according to criteria of functionality, simplicity and flexibility. Once the objectives of the application to develop were established and the three criteria mentioned above were taken into account, the minimum set of functions necessary for its incorporation into the system was mapped. This results in the possibility of being able to establish changes to the application in order to adjust it to a dynamic context, as presented in the architecture of four layers. See Figure 2.

FIGURE 2. THE LIGHTWEIGHT IOT FRAMEWORK OF THE CASE STUDY



The layers proposed are described below:

Data acquisition layer: In this layer, a mobile application was implemented to receive signals from the beacon devices and the considered access points. The mobile application runs in the smartphone of the people subjected to monitoring.

• Sensors: For this case study were used two types. The first one corresponds to Estimote beacons, which are devices that constantly emit a Bluetooth signal containing information about their identification and their configuration parameters [13]. The intensity of the Bluetooth signal emitted allows us to infer the distance from it to the receiver. The second one is the Access Point devices pre-installed in the buildings to provide internet. The Access Points also transmit a WiFi signal with a power that allows calculating the distance from it to the receiver. In addition to the intensity of the signals, the identification and configuration data of the mobile device is received.

• Mobile App: It is composed of two blocks, which are the acquisition of data and the transfer of data. This mobile application has the objective of capturing information that will be sent to the processing layer through the volley library using the HTTP protocol. This library allows the application to communicate with the data processing layer [2]. It can use up to 7.75 Megabytes of storage and it also has a user authentication method using email and password. See Figure 3.

- Data acquisition block: The open-source Beacon Library version 2 [13], and the API developed by Google called WiFi Manager were implemented [14]. The following image shows the interface of the mobile application for the acquisition of data from the Bluetooth and WiFi sensors.

- Management and data transfer block: The objective of this block is to allow communication between the data management layer and the data acquisition layer, which belong to the proposed framework. The data transfer is completed using the HTTP protocol to send and receive information.

FIGURE 3 . MOBILE APP

Processing layer:

The objective of this layer is to authenticate all connected devices, monitor the health of all sensors and manage all the information sent by the mobile application.

• Sensor health: This module performs a sensor battery level analysis to detect when the sensor or battery needs to be changed. The objective of this is to prevent the absence of the signal due to a flaw in the sensor affecting the functionality of the system.

• Manage and monitor sensors: This module aim is to take control over the sensors available, to know the location of these sensors and the characteristics of the device, for example, its identification number, the brand to which it belongs and its MAC Address. The monitoring is focused on the interaction of the sensor with the system, allowing us to know if, at any time, information is stopped from any sensor.

• Sensor authentication: This module allows only the information received by the sensors that are registered to be taken into account. Authentication is completed using the MAC Address and its universal unique identifier. This process is carried out with the help of the data sent by the mobile application and the information already stored in the database.

• Localization algorithm: This module integrates an algorithm to localize people in indoor spaces, using as an input the signals from beacons and access points previously captured by the smartphone device that the person subjected to monitoring carries with them. There are several indoor localization methods, which combined with the correct technology have demonstrated acceptable results: Triangulation, Trilateration, Fingerprinting. In our case, we use the Fingerprinting method. We are not specifying more details about the implementation of this method since it is not the purpose of this article. So, we are going to assume this module as a black box that continuously receives signals from a mobile application integrated into a Smartphone and outputs the location of the person who carries the device [15], [16].

Data management Layer: This layer is made up of two blocks: The IoT component and the Data Access component.

IoT component: integrates all the functionality related to the management of information from different data sources in three basic aspects: reception, distribution and visualization. The first one receives data providing from different devices, services or applications, in this case from the mobile application installed on the Smartphone devices of the people subject to monitoring [17], [18]. Distribution refers to a task in which the received data is sent to the different services and applications that require it to perform some specific functionality. Visualization seeks to supply the need that some applications have to visualize the behavior of data in real time.

For this case study, some free software modules were used to encompass the three aspects mentioned above. The first module used is the Orion Context Broker, which is a generic component of the FIWARE platform [19]. This module allows the reception and distribution of data from different data sources. In addition, the Orion Context Broker allows the management of context information, for which it relies on a MongoDB database in which the last state of all registered objects is saved. Likewise, the Quantum Leap component is used, being a part of the generic FIWARE components, it specializes in the management of time series. Additionally, QuantumLeap integrates a Grafana module to cover the visualization of data. In addition, it integrates the CrateDB database which was developed for the management of time series, which is very frequent when working with data from sensors. The inclusion of these modules allows us to integrate a functional, flexible and robust component on the server side for the reception of the data and its distribution to the different consumers, as well as visualizing and monitoring the data over time.

• Data access: This is the layer in charge of providing access to the different storage media available, as well as in charge of data security. Three types of databases were used, the first one is in charge of storing the information of the sensors and the access information of the users, the second database is in charge of storing the general information of the sensors (these data are called context data) and the third database is in charge of storing the historical information of the context data. The data storage was made through the following database managers:

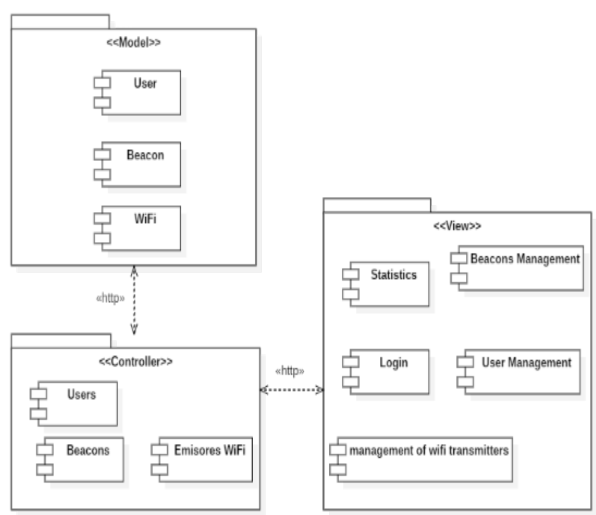
• MariaDB: This open source relational database has the virtues of performance and stability [20]. Its objective was the storage of data pertaining to the characteristics of the sensors and user data, as well as the communication with the data processing layer.

• MongoDB: It is a distributed database, based on documents, typical of the cloud era [21], [22]. The Orion Context Broker component of the FIWARE platform is based on the open source MongoDB technology [19]. The need for this non-relational database was for the persistence of the context data that is collected by the mobile application.

• CrateDB: It is a high-performance database which allows managing the context data providing persistence in the data [24], [24]. This database stores any changes to the context information that are made in the Orion Context Broker component, which allows the information to be queried by the processing layer and the application layer.

Application layer: This layer is the one that contains the interface for the user to interact with the system. The Web component was built following the model-view-controller architecture [26], illustrated in Figure 4.

FIGURE 4. THE INTERFACE MODEL-VIEW-CONTROLLER



Model: The models are responsible for representing each of the entities with which the system works. In addition to representing entities, it contains the mechanisms to access information and also update its status. The models used in the Web component are Users, Beacons and WIFI's. Their objective is to make the connection with the database, as well as to consult, insert, update and delete information from the tables corresponding to each model, connecting with the management layer of data. There is also a model that is in charge of consulting the information captured from the sensors to send it to its controller to be processed.

Controller: Its main objective is to maintain the connection between models and views. They contain the necessary mechanisms to manipulate the information depending on the need, creating a data flow between the model and the view [25]. The four controllers implemented are Users, Beacons, WiFi transmitters and Location Algorithm. This last controller is in charge of calculating the approximate location using the information from the sensors and the information captured by the mobile application.

View: Its objective is to allow the user to interact with the system. The views implemented in the Web component allow the management and monitoring of health in the devices, also the graphic observation of the context data and the user's managing features. Another Web component interface allows consulting the approximate location of the smartphone device, in this case from the beacons and WiFi transmitters.

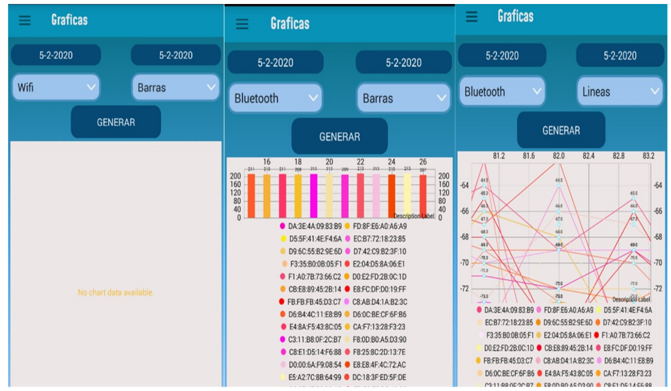
- Mobile application and Web application User interface

This section shows some of the interfaces implemented in a Web application and in the mobile version, where you can manage all the registered sensors and the authorized users, as well as display the information of the people being monitored.

FIGURE 5. MOBILE APPLICATION USER INTERFACE

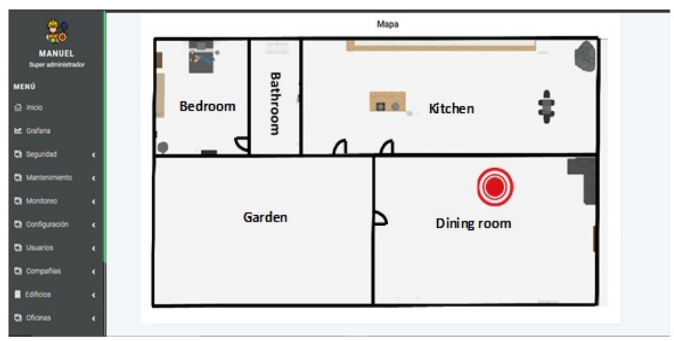
In Figure 6 we can see the interface of the graphs corresponding to the data captured by the application. These graphics were made using the MP Android Chart library [26].

FIGURE 6. MOBILE APPLICATION USER INTERFACE



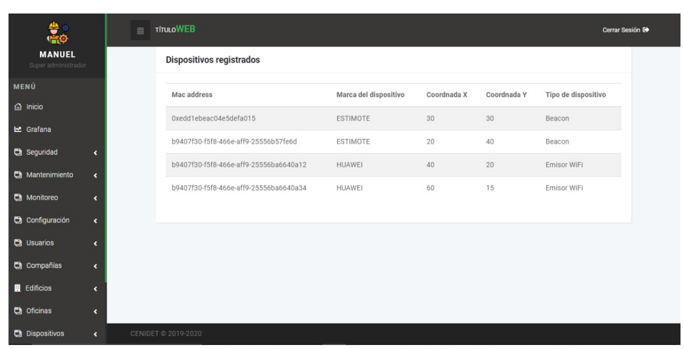
In Figure 7 we can see the location of a person being monitored, shown by a red point marked on the map. The map shows the circulation area in an interior space and the phone carried by the person has been registered in the location platform.

FIGURE 7. WEB APPLICATION MONITORING



The Figure 8 shows presents the information corresponding to the sensors registered.

FIGURE 8. WEB APPLICATION SENSORS REGISTRY



The Figure 9 shows the Grafana component embedded in the Web application, this component allows graphing the behavior of the context data that has been captured by the mobile application and has been stored in the CrateDB database.

FIGURE 9. WEB APPLICATION GRAPHING

VI. CONCLUSIONS AND FUTURE WORK

One of the main challenges faced by software entrepreneurs interested in developing solutions to meet needs in the IoT domain for small and medium-sized companies is the cost associated with purchasing licenses from cloud IoT providers, as well as the difficulty of adapting the solutions that these providers offer given the complexity to integrate the tools they include.

Through the development of a case study that integrates free software tools with the tools offered by the FIWARE open software platform, the architecture of a lightweight IoT application for monitoring the activity of the elder is stated.

The proposed architecture can allow the agile and low-cost development for innovative applications in various areas, particularly those where there is a need to monitor people or objects, in order to generate analytics, as well as the execution of the business rules required.

It is sought that, in a next version, we will seek a better flexibility for the incorporation of user services, to strengthen the corresponding part to device management, as well as to integrate greater functionalities for data analysis. In all cases, evaluating the use of free software tools, as well as those offered by the FIWARE platform.

REFERENCES

[1] IoTWorldOnline. Available in: https://www.iotworldonline.es/las-grandes-estadisticas-del-internet-de-las-cosas-iot/. Accessed in: 10 Oct. 2020.

[2] Farzad Kiani, "A Survey on Management Frameworks and Open Challenges in IoT", Wireless Communications and Mobile Computing, vol. 2018, article ID 9857026, 33 pages, 2018.

[3] Gonçalo Marques, Rui Pitarma, Nuno M. Garcia and Nuno Pombo, "Internet of Things Architectures, Technologies, Applications, Challenges, and Future Directions for Enhanced Living Environments and Healthcare Systems: A Review", Electronics 8, 1081, 2019.

[4] R. Luis de Moura, T. Monteiro Brasil, L. de Landa Ceotto, A. Gonzalez, L. Paulo Barreto and L. Bassini Werner, "Industrial Internet of Things: Device Management Architecture Proposal," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 1174-1178.

[5] R. Moura, L. Ceotto, A. Gonzalez and R. Toledo, "Industrial Internet of Things (IIoT) Platforms - An Evaluation Model," *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2018, pp. 1002-1009.

[6] Microsoft. Microsoft Azure. Available in: https://azure.microsoft.com. Accessed in: 10 Oct. 2020.

[7] AWS. Amazon Web Service. Available in: https://aws.amazon.com. Accessed in: 10 Oct. 2020.

[8] Oracle. Oracle Integrated Cloud. Available in: https://cloud.oracle.com. Accessed in: 10 Oct. 2020.

[9] Sarada Prasad Gochhayat, Pallavi Kaliyar, Mauro Conti, Prayag Tiwari, V.B.S. Prasath, Deepak Gupta, Ashish Khanna, "LISA: Lightweight context-aware IoT service architecture", Journal of Cleaner Production, Volume 212, 2019, pp 1345-1356.

[10] Marques, G.; Pitarma, R.; M. Garcia, N.; Pombo, N. Internet of Things Architectures, Technologies, Applications, Challenges, and Future Directions for Enhanced Living Environments and Healthcare Systems: A Review. Electronics 2019, 8, 1081.

[11] FIWARE. Available in: https://www.fiware.org. Accessed in: 10 Oct. 2020.

[12] Alicia Martinez, Fernando Ramirez, Hugo Estrada, L.A. Torres "A Generic Module For Collecting Data in Smart Cities", The International Archives of the Photogrammetry, Remote Sensing And Spatial Information Sciences, Volume XLII-4/W3, 2017, 2nd International Conference On Smart Data And Smart Cities, 4–6 October 2017, Puebla, Mexico.

[13] Estimote, «Technical specification of Estimote Beacons and Stickers, » Available:https://community.estimote.com/hc/en-us/articles/204092986-Technical-specification-of-Estimote-Beacons-and-Stickers, https://github.com/AltBeacon/android-beacon-library/. Accessed in: 11 Oct. 2020.

[14] Google, «developer.android,» GOOGLE, 27 09 2017. Available: https://developer.android.com/training/volley. Accessed: 11 Oct. 2019.

[15] Gu, F., Hu, X., Ramezani, M., Acharya, D., Khoshelham, K., Valaee, S., y Shang, J., "Indoor localization improved by spatial context - a survey", ACM Computing Surveys, Vol. 52, No. 3, Article 64, 2019.

[16] Zekavat, R., y Buehrer, R. M. "Overview of global positioning systems. En Handbook of position location: Theory, practice, and advances", pp 655-705, IEEE.urlhttps://ieeexplore.ieee.org/document/8633807, 2019

[17] Radius Networks, «https://altbeacon.github.io/android-beacon-library/» 2014. [On line]. Available: https://github.com/AltBeacon/android-beacon-library/, Access in: 10 Oct. 2019.

[18] Google, «developer.android.com,» Google. Available: https://developer.android.com/reference/android/net/WiFi/WiFiManage. Accessed in: 10 Oct. 2019.

[19] Telefonicaid, «WELCOME TO ORION CONTEXT BROKER,» telefonicaid, 22 0220 16. Available: https://fiware-orion.readthedocs.io/en/master/, Accessed in: 11 Oct. 2019.

[20] M. Foundation, «MariaDB Server: The open source relational database,» MariaDB Foundation, 26 03 2020. Available: https://mariadb.org/. Accessed in: 11 Oct 2019.

[21] Mongodb, «mongodb,» mongodb. Available: mongodb.com/es. Accessed in: 11 Oct. 2020.

[22] Fiware, «FIWARE Getting Started,» Fiware. Available: https://documenter.getpostman.com/view/513743/fiware-getting-started/RVu5kp1c. Accessed in: 16 Oct. 2020.

[23] Telefonicaid, «CRATEDB,» Fiware. Available: https://quantumleap.readthedocs.io/en/latest/admin/crate/. Accessed in: 4 Oct. 2019.

[24] Telefonicaid, «GRAFANA,» Fiware, Available: https://quantumleap.readthedocs.io/en/latest/admin/grafana/. Accessed in: 12 Oct. 2019.

[25] C. Á. Caules, «El patrón MVC, arquitectura cliente vs servidor,» https://www.arquitecturajava.com/, 12 02 2016. [On line]. Available: https://www.arquitecturajava.com/patron-mvc-arquitectura-cliente-vs-servidor/. Accessed in: 07 Oct. 2019.

[26] PhilJay, «MPAndroidChart,» PhilJay, 29 06 2016. Available: https://github.com/PhilJay/MPAndro. Accessed in: 07 Oct. 2019.