ISSN 1870-4069

Testing Environment for Precision Agriculture Supported in IoT

E. A Quiroga[1], S. F. Jaramillo[1], W. Y. Campo Muñoz[1], G. E. Chanchí Golondrino[2], C. L. Burbano G.[3]

[1] Universidad del Quindío, Armenia- Quindío, Colombia
{eaquirogam, sfjaramilloc}@uqvirtual.edu.co
wycampo@uniquindio.edu.co
[2] Institución Universitaria Colegio Mayor del Cauca, Popayán- Cauca, Colombia
gchanchi@unimayor.edu.co
[3] Fundación Universitaria de Popayán, Popayán- Cauca, Colombia
claritaluciab@gmail.com

Abstract. The technification of agriculture from the field of IoT allows to contribute to improve the agricultural productivity, from the provision of meteorological forecasts in an agile way using networks of sensors for the real time monitoring of multiple variables. However, a large number of sensors, capture devices and data processing technologies are available for the implementation of this type of system, which makes it necessary to choose the most suitable technologies for the experimentation and construction of IoT systems focused on agriculture accuracy. In this paper we present a test environment for precision agriculture, which was configured from a set of hardware and free software tools associated with IoT. The proposed environment took into account the Lambda architecture and aims to serve as a guide for the implementation of services in scenarios of precision agriculture.

Keywords: Climate Variables; Internet of Things; Lambda Architecture; Precision Farming; Weka.

1 Introduction

Crop management supported by precision agriculture tools encompasses monitoring activities, decision support tools and actions that automatically control one or more systems (irrigation, frost protection, fertilization, etc.). (GPS), wireless sensor networks, drones, multiple electronic devices, and the application of computer tools from Machine Learning (IoT Simple, 2017). Through this wide range of resources, a farmer can obtain detailed crop information, soil conditions and even more granular climatic variations than traditional farming techniques could not provide; Impacting the quality of the products, the processes that are carried out and the raw materials used in the activity (Solutek, s.f.).

This technification required in precision agriculture environments can be carried out through Internet of Things (IoT). In IoT things, that is, embedded devices can be ideally available anytime, anywhere. In a more technical sense, IoT consists of the integration of sensors and devices into everyday objects so that they are connected to the Internet through fixed and wireless networks (Fundación de la Innovación Bankinter, 2011) (Ruiz, 2016). Cost reduction, improvements in crop processing and care, optimization of the use of material and human resources, increased yield per hectare cultivated, higher quality of final product and reduction of disposal, Compliance with national and international requirements of production and product characteristics, etc., are some of the benefits of opting for a precision agriculture solution supported by IoT.

In order to achieve these objectives in the agriculture sector, nowadays a large number of sensors, capture devices and data processing technologies are available, making it necessary to choose the most suitable technologies for the experimentation and construction of IoT systems focused on the agriculture of precision. However, the problem for a farmer who has not yet implemented some of these precision agriculture tools in his crops is the high cost and complexity in interpreting the information obtained from these technologies. Due to this problem, there is a great demand and need to integrate a set of appropriate and low-cost components, given the large number of technologies that are emerging to simplify the collection, storage and processing of climatological data (Lopez, Chavez, & Sánchez, 2017) (Taffernaberry, Diedrichs, Pérez, Pecchia, & Tabacchi, 2016).

Taking into account the above problem, in this paper we present as main contribution a test environment for precision agriculture supported in open technologies for obtaining, visualization and real time analysis of climatic variables in a small scale test crop. This testing environment is addressed from three high-level views associated with the business model, functionality and implementation of the environment for precision agriculture farming scenarios. Each of the views has taken as reference the layer structure of the Lambda architecture (data capture layer, storage layer, processing layer and query layer) (Deshmane, 2015). This testing environment aims to serve as a reference for the implementation of services in the area of agriculture, in order to improve agricultural productivity, through the flexibility and functionality offered by the combination of free hardware and software tools for implementation of services supported in IoT. The rest of the article is organized as follows: section 2 describes each view of the test environment; section 3 presents the evaluation of the test environment and finally section 4 presents the conclusions and future work derived from the present research.

2 Test Environment Supported in IoT

This section describes the test environment for precision agriculture, which is represented by three views (Jimenez, Hincapié, & Quintero, 2016): business view, functional view and implementation view (see Fig 1). The business view presents the business model of the test environment in the context of precision agriculture. In the functional view the different functional components of the test environment are described and in the implementation view are presented the different hardware and software components that were chosen for the construction of the environment.

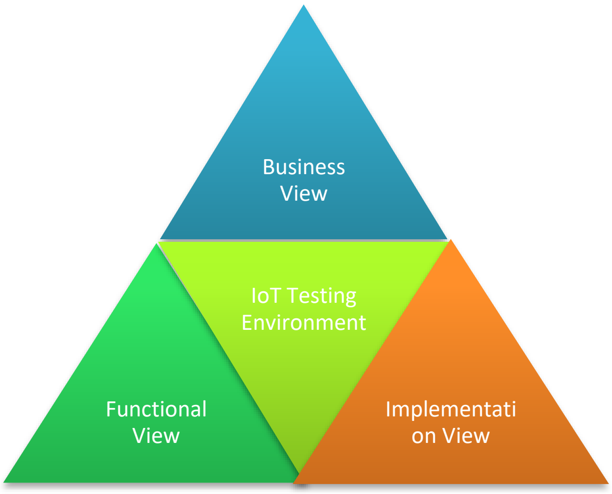

Fig.1. Diagrams of the IoT test environment.

2.1 Business View

This business view presents the proposed IoT testing environment from the four layers of the Lambda architecture: capture layer, storage layer, processing layer and query layer (see Fig. 2).

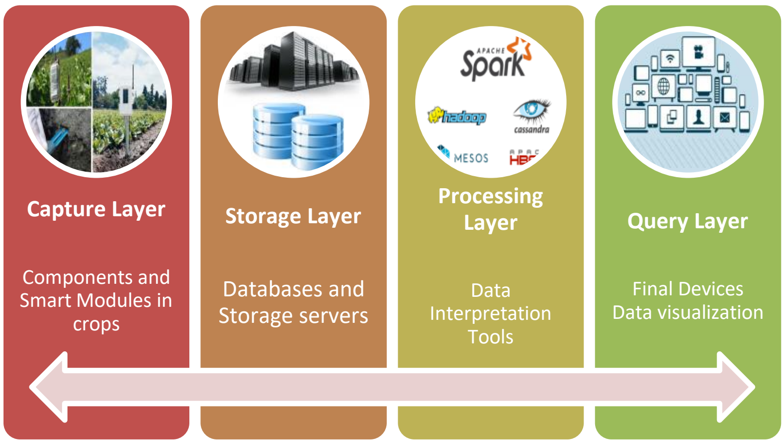

Fig. 2. Business diagram for the IoT test environment.

As shown in Fig. 2, the capture layer includes sensors and data acquisition modules, as well as embedded systems, microcontrollers or meteorological stations for the taking of climatic conditions in the crop. The storage layer consists of platforms, servers and database that allow to save all the data obtained in the capture layer. In the processing layer, these stored data are extracted to perform their analysis and obtain information of interest that can be applied to the crops. The final component in the environment is the query layer, which allows the farmer to visualize in real time the behavior of the variables associated with the crop, as well as the predictions and recommendations to be taken according to the needs of the crop.

2.2 Functional View

In Fig. 3, the different functional modules associated with the different layers of the test environment are presented. In the capture layer, the incoming data stream is taken with the data capture system (sensors-microcontroller) and can be sent in two directions, either the storage layer (online database, cloud, servers) or to the processing layer (machine learning tools). This information can be consulted through internet using final devices (web services, mobile applications) by means of the query layer.

The storage or security layer provides a history file, saving all the data that has been collected. Its storage could be forever, or partially. Storage forever would support advanced analysis and the predictive models to be implemented would have greater accuracy in creating signals on the transmission platform and for general queries. The rate or processing layer is defined as a combination of queue holdings, streaming and operational data by performing real-time analysis. The storage layer bases its estimates on previously stored data, so it is the responsibility of the speed layer to obtain real-time analysis from fast-moving data.

The functions described for each of the layers of the test environment can be broken down into functional sub-modules as presented in Fig. 3, which presents the functional sequence of the environment supported in IoT for precision agriculture, according to the layers provided by the Lambda architecture

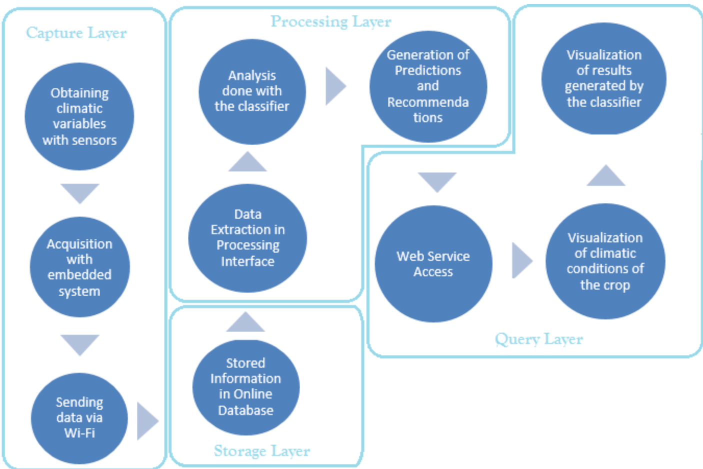

Fig. 3. Functional sequence of the IoT test environment.

2.3 Implementation View

In this section we present in detail the hardware and software components that were chosen for the implementation of the IoT supported test environment described in this article (see Fig. 4), which are organized according to the layers of the Lambda architecture. According to Fig. 4, in "1", sensors of temperature, relative humidity, soil moisture and luminosity are placed in the test culture to realize the corresponding climatic variables. In "2" by means of the IoT Arduino YÚN board (Schwartz, 2014), these sensors are configured with their necessary parameters and wireless communication is established, thus forming the data capture layer. Through the capture board is sent in real time the registration of each of the variables associated with the sensors to be stored in "3", through the online database Ubidots (Torquica & Guzmán, 2016). In "4" using the Spark web development framework, a web service is implemented in Java that allows the online processing of the data captured from the previous layers.

To perform the processing of this data, the Spark framework integrates the API provided by the Weka data mining tool into "5". Using this API, a classifier based on the Naive Bayes algorithm is implemented with the information contained in the file in arff (attribute-relation file format) format, obtaining the Processing layer, which provides predictions of the atmospheric conditions of the crop as well as recommendations for applying inputs. Finally, in "6" the end users access the web service to consult the predictions and recommendations generated by the classifier, thus obtaining the query layer. The following is a detailed description of each layer of the implementation view.

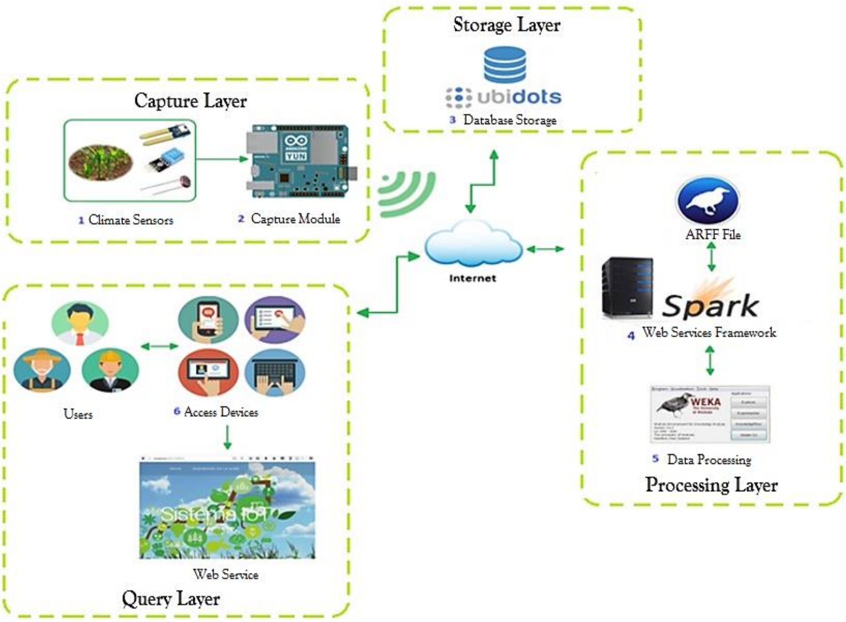

Fig. 4. Implementation diagram of the IoT environment.

Capture Layer

Fig. 5 shows the connection scheme established to operate the sensors of temperature, relative humidity, soil moisture and luminosity connected next to the Arduino YÚN board for the capture of climatic variables.

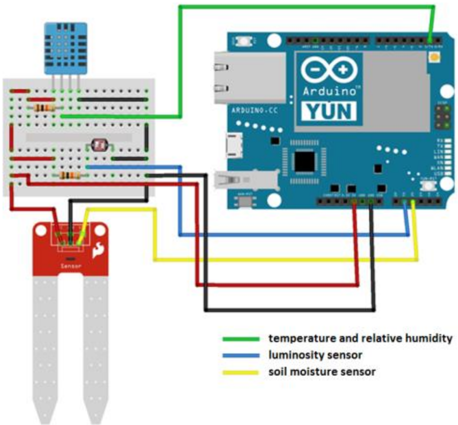

Fig. 5. Schemes of connection of sensors with microcontroller.

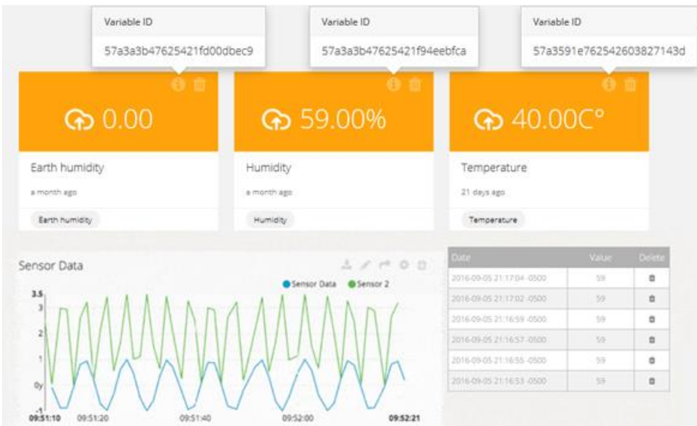

Fig. 6. Associated climatic variables in online database and their storage.

For this research the IoT Arduino YÚN board was used as the appropriate embedded system in the data capture layer and for use in the design of connected devices and in IoT projects. This board combines the power of Linux with the ease of use of Arduino since having communication and configuration via Wi-Fi. In adition it allows to send the data of climatic variables wirelessly to the other layers of the proposed environment. The sensors used were the DHT11, SEN92355P and a LDR, that were chosen for their low cost to meet the requirement of economic components for the implementation of the test environment.

Storage Layer

To establish the Lambda-based IoT testing environment, you must have a storage manager that serves as the storage layer. For this, we selected the online database Ubidots which is a specialized tool for IoT applications, which offers a platform for developers. Ubidots allows you to easily capture sensor data to become useful information, as well as support for the Arduino YUN capture module. Fig. 6 shows the Ubidots administration interface with the different climatic variables captured through the Arduino YÚN board.

Processing Layer

This layer comprises two stages of processing: the generation of a file in the specific format of the Weka software and the processing of the generated file making one of the Weka data mining software:

ARFF file generation. To process climate variables with the Weka data mining tool, information must be organized in a specific way, this is done by an ARFF file. The list of attributes or meteorological variables to be considered (humidity, relative humidity, temperature, etc.) and the class in which the attributes of these attributes are classified are used in the header of the ARFF file, which is used to generate irrigation predictions. In the body of the document are located the different instances of these attributes, which are fed with the values obtained from the sensors of the crop.

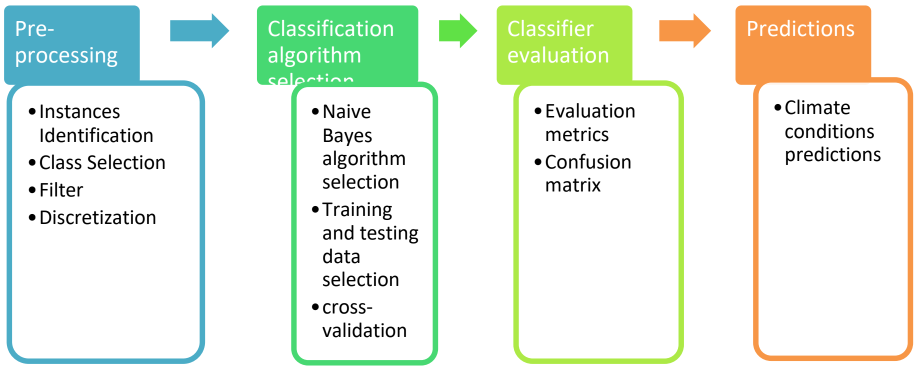

Fig. 7. Weka data mining software operation process.

Analysis with Weka software. Once the ARFF file is generated, it is analyzed through the API provided by the Weka tool, following the sequence of steps of Fig. 7. The first stage is the pre-processing, which consists of the manipulation of vectors with discrete content of the sensed values, where loading the file with the necessary data, identifies the class to predict and can be filtered attributes, instances or discretization. The second stage consists of the generation of the classifier using a learning algorithm to which a set of data is given which correspond to the contents of the ARFF file. For this investigation, the Naive Bayes algorithm was used and the data provided to generate the classifier were selected as a training data set and with cross validation. In the third stage the evaluation of this classifier is done, which can be defined as the degree of agreement between the values of the class assigned by the classifier and their correct locations according to the data of the ARFF file.

Query Layer

To complete the final stage of the testing environment, a graphical GUI access interface was developed by the Spark framework. The interface developed allows real-time visualization of the climatic data of temperature, relative humidity, soil moisture and illumination being taken by the data capture component, see Fig. 8. In addition, this interface makes it possible to consult the history online with the Data stored in the Ubidots database, as well as the predictions and recommendations associated with the crop.

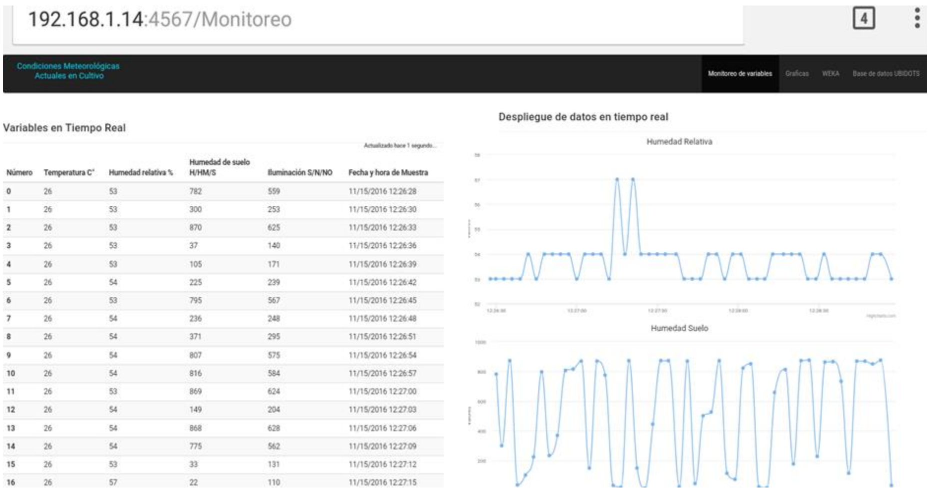

Fig. 8. Visualization of the web service for the query of the information.

3 Evaluation of the IoT Test Environment

Finally, by way of evaluation of the testing environment, the main component of the environment was validated, which corresponds to the processing module, which is responsible for extracting and processing information from the Ubidots database. This component was subjected to load tests in order to determine its capacity and response time. These tests make it possible to obtain the limits under which the processing module will behave appropriately and let you know when different optimization strategies need to be implemented. Thus, Fig. 9 shows the response times for the stress tests performed on the processing module, which were carried out by incremental variation from 100 to 1000 concurrent requests.

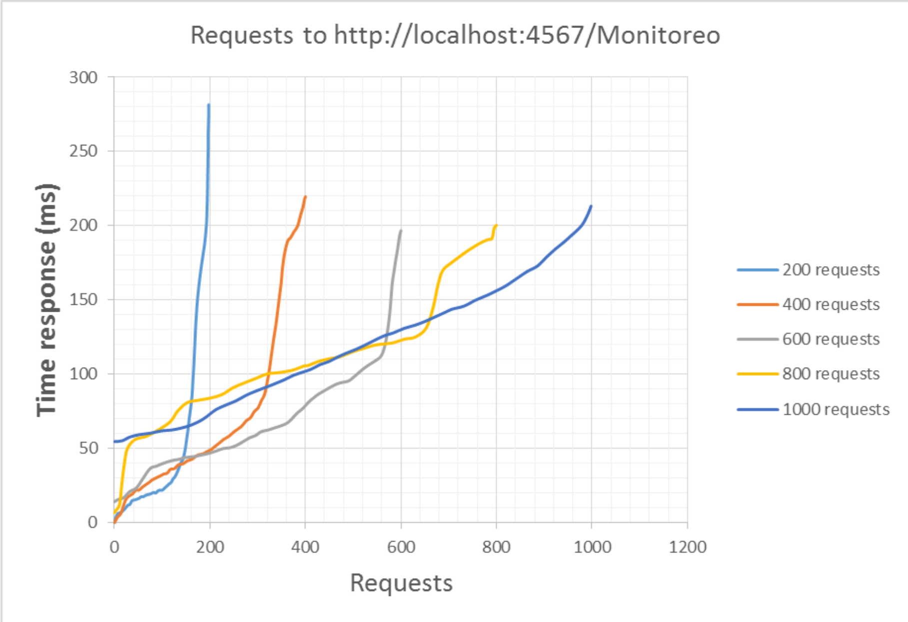

Fig. 9. Response times for stress test in processing layer.

Fig. 9 shows that the maximum time obtained in each of the concurrent request series (200,400,600,800,1000) varies between 200 and 260 milliseconds. In the same way, in each of the series as the number of requests increases, the time grows proportionally; being the series of 200 requests the greatest slope, and the series of 1000 petitions the smallest one, which allows to conclude that the processing module presents an appropriate behavior as the number of concurrent requests grow. Thus, for the particular case of 1000 concurrent request series, the response time increases from 160 ms to 220 ms between 800 and 1000 requests, which represents an average of 0.3 ms for each additional concurrent request.

4 Conclusions and Future Work

Taking advantage of the great flexibility and functionality offered by the combination of low cost hardware tools and open source software, we have achieved the design, implementation and deployment of an IoT supported test environment, which is addressed from three design views. In addition, four layers of operation are deployed, including the capture of climatic variables, the storage of this information, its processing and its consultation. This environment is intended to serve as reference for farmers and the community interested in using new technologies, techniques and services based on IoT that seek to improve agricultural productivity.

It has been possible to integrate all the components proposed in this research to be based on the Lambda architecture, considering the Arduino YÚN capture module as a data acquisition component, the Ubidots online database in the storage stage, the Weka data mining tool used as the processing stage and the Java Spark web design framework for the query stage; resulting in a testing environment for IoT.

As a future work, it is intended to evaluate this IoT test environment in other application contexts where the capture of high volumes of data is required to perform a real-time processing of these, so that contexts such as education, marketing, smart cities, etc.

References

1. Deshmane, S.: Using the Lambda Architecture on a Big Data Platform to Improve Mobile Campaign Management. Talentica Software (2015)
2. Fundación de la Innovación Bankinter: El Internet de las Cosas - En un mundo conectado de objetos inteligentes (2011)
3. IoT Simple S. A.: Agricultura inteligente (2017)
4. Jimenez, J., Hincapié, J.A., Quintero, J.: MDSD Multi-Plataforma: Más Allá de la Vista Funcional. Revista Ingenierías - Universidad de Medellín (2016)
5. Lopez, J., Chavez, J., Sánchez, A.: Modelado de una red de sensores y actuadores inalámbrica para aplicaciones en agricultura de precisión. In: Humanitarian Technology Conference (MHTC), IEEE Mexican. Puebla: IEEE (2017)
6. Ruiz, D.: Diseño de un sistema en cloud para controlar dispositivos IoT a través de Internet. Facultad de Ingeniería. Universidad Militar Nueva Granada (2016)
7. Schwartz, M.-O.: Wireless Security Camera with the Arduino Yun. Adafruit Learning System (2014)
8. Internet de las cosas (IoT) para la agricultura Colombia (2017)
9. Taffernaberry, J.C., Diedrichs, A., Pérez, C., Pecchia, M., Tabacchi, G.: Gateway 6LoWPAN para red de sensores inalámbricos. In: XLV Jornadas Argentinas de Informática e Investigación Operativa. Buenos Aires-Argentina (2016)
10. Torquica, L., Guzmán, M.: Desarrollo de Sistema de Seguridad para Automóviles con IOT y Smartphone. Facultad de Ingeniería. Bogotá: Universidad Francisco José de Caldas (2016)