

Novus-io: An Internet of Things Platform for Academic Projects

Camilo Lozoya¹, Alberto Aguilar-Gonzalez¹, Antonio Favela-Contreras¹ and Arturo Zamora¹

¹School of Engineering and Sciences, Tecnológico de Monterrey
Chihuahua, 31300 - Mexico

[e-mail : camilo.lozoya@itesm.mx, alberto.aguilar@itesm.mx, antonio.favela@itesm.mx,
A01560815@itesm.mx]

*Corresponding author: Camilo Lozoya

*Received April 16, 2018; revised June 22, 2018; accepted July 18, 2018;
published December 31, 2018*

Abstract

Internet of things (IoT) is based on a global dynamic information network with cloud services where a great number of devices (things) exchange data to provide added-value services and products. There are several commercial and open source IoT platforms available in the market to connect devices to internet; however, they have cost and operational constraints that make them not suitable for academic projects. In this work, an IoT platform, known as Novus-io, is introduced in order to support academic projects for undergraduate students. With this platform and proper training, undergraduate students from different majors (not only from information technology and electronics) are capable to upgrade their school projects with IoT functionalities. The objective of this approach is to provide to any undergraduate student skills and knowledge on IoT, so they will be prepared, in their imminent step toward professionalism, to understand the relevance of digital services in today's world.

Keywords: Internet of things, academic projects, open source, performance evaluation, IoT platforms.

1. Introduction

Internet of Things (IoT) is becoming one of the most wide-spread technology available today, its application domain covers areas such as smart homes, smart buildings, wearables, smart cities, smart grids, industrial automation, healthcare, environmental monitoring, transportation and logistics, smart farming among others. The Internet of Things concept considers a group of objects/things/devices connected through a communication infrastructure, and exchanging relevant information in order to provide value-added services, also known as smart systems. The development and research on Internet of Things have focused on aspects such as efficient network bandwidth, communication protocols, data security, data mining, wireless sensor networks, smart sensors and application domains. However, only sporadic efforts have been concentrated on the educational side, specifically on how to train professionals with a non-technical profile in IoT, but with enough knowledge in the application domain area. Train IoT end users to have a basic knowledge on this technology is, without a question, beneficial, since the best innovation ideas in any domain usually come from the process experts when they are capable to understand the scope and services offered by the state of the art technology in their specific fields of knowledge.

The potential impact of Internet of Things in today's economy is indisputable, according to Cisco Internet Business Solutions Group, the number of devices connected through internet will grow up to over 50 billion by 2020 [1]. Furthermore, according to McKinsey Global Institute the added value that IoT may provide to the products and services, is estimated on \$4 to \$11 trillion of dollars globally by 2025 [2]. Therefore, today's information technology and electronics professionals and engineers need to be trained to understand what smart systems are, how existing commercial solutions work, what promises and drawbacks they bring, and how to design and implement such systems [3]. More important, as already denoted by [4], the professionals with non-IoT technical background, who knows the application domain, will play a significant role in the future development of Internet of Things by providing innovative ideas and solutions to current and future process related problems. The work presented in this paper, describes the design and implementation of an IoT platform, which allows undergraduate students, with technical profile and non-technical profile in IoT, to enhance their academic projects by integrating IoT functionalities to their prototypes in a simple and straightforward way.

An IoT platform is a collection of software, hardware and communication components that, when integrated, allows developers to create projects or products that involve a group of connected devices that conforms a smart system, i.e. capable to collaborate and interact autonomously. Several IoT platforms such as ThingSpeak (<https://thingspeak.com/>), Adafruit IO (<https://io.adafruit.com/>), Freeboard (<https://freeboard.io/>), Thinger.io (<https://thinger.io/>) and ThingsBoard (<https://thingsboard.io/>), considered as open-source are partially available for free and fully available for a cost. Others like Xively (by LogMeIn), ThingWorxs (by PTC) and Cayenne (by MyDevices) are considered commercial platforms since they focus on business solutions. Even large companies like Amazon (AWS IoT), Google (Google Cloud IoT), Microsoft (Azure IoT) and IBM (IBM Watson) provide cloud computing services that allow the connection of devices to internet, but these platforms have been designed to be used for well-established companies that can afford the cost and may have an information technology department to support their operation. Therefore, for undergraduate students, commercial platforms are not suitable for their academic projects, mostly due the user cost, the

maintenance complexity and the operational constraints, so just open-source platforms are considered by the work presented in this paper.

The introduced IoT platform, known as Novus-io, has been designed specifically for undergraduate students from different majors with the objective of improving their skills and knowledge on this technology through hands-on workshops and academic projects that includes the connection, visualization and control of devices through internet. Today's established and emergent companies require professionals with IoT knowledge, with this in mind, the proposed platform has been implemented as a technological tool to prepare any undergraduate student with these abilities. The rest of the paper is structure as follows: Section 2 describes different approaches already proposed on how to educate and prepare students on IoT. Section 3, analyzes the characteristics of open source platforms and describes the features of the proposed one. Section 4, analyzes the results obtained after the use of the proposed platform from different groups of undergraduate students. Section 5, provides the conclusions of this work.

2. Related Work

The design of products and services based on connected devices not only require skills on information technology and electronics, but also the understanding of digital services design and business models [5]. Universities should include Internet of Things as part of their curricula of any undergraduate program [6]. Furthermore, IoT should be considered as a transversal competency at the same level of ethics, citizenship, oral and written expression; in order to prepare the students for their imminent professional challenges in a world where internet is a key element for any business. Next, some related works in the area of education and Internet of Things are reviewed.

In [7], an Internet of Things course was designed to develop the technical skills for students, researchers, engineers, and faculties, who desire to address some of the issues faced by the people around the world in the fields of smart sensors and communications with hands-on activities. The participants were from different engineering backgrounds including computing, electronic, electrical and mechanical. The hardware platform was based on Intel Galileo and Intel Edison boards; however, there is no information about the software and communication platform.

An active learning IoT course was implemented by [3], to encourage engineering students from different programs (Computer, Electrical, Mechanical and Industrial) to work in multidisciplinary teams to develop an IoT project in the area of ambient intelligence. The hardware platform was based on Raspberry Pi boards plus other demo kits provided by companies that supported the course.

In [4], an Internet of Things MOOC (massive open online course) is provided by The Open University UK where instead of focusing just on the technical aspects of IoT, the course was designed to construct in the students a new view in the understanding of their own world, helping them to recognize their role in realizing the IoT's potential. This educational aim is achieved by using concrete experiences, creative experimentation, active participation and collaborative learning. The hardware platform is based on Arduino boards, meanwhile Microsoft's .NET Gadgeteer is used as a software platform; however, this platform is more an electronic rapid prototyping framework than an IoT service module.

In [8], analyzes a set of courses proposed by a large group of European universities in order to accelerate the adoption of Industry 4.0, where knowledge and competences on Internet of Things becomes a key element in this domain area.

Besides these works on designing IoT courses to prepare students, other papers analyze the impact of the emergent internet-based technology as an opportunity for companies to obtain competitive advantages from these services [9]. In a similar way, [10] investigates how IoT has reshaped the education business model by changing from a knowledge-transfer model to an active collaborative self-directed model by the disruptive influence of technology in today's educational institutions. In [11], studies how Internet of Things may affect the learning process by combining the use of technology and smart education concepts.

Regarding more technical aspects for implementing IoT platforms for academic purposes: Fisher *et al* [12] highlights the benefits of using open-source hardware and software platforms for the implementation of IoT projects for “non-professionals” designers, where Arduino and ThingSpeak are used respectively. In [13], an IoT academic project is presented for monitoring people's health inside their homes, this application has to meet requirements of reliability, robustness, and energy-efficiency; a detailed description of the platform architecture is discussed in the paper. Another platform for academic projects is proposed by [14], based on Raspberry Pi and Arduino boards as hardware platforms and ThingSpeak as a software and communication platform, in this case the developed platform is used for small projects in STEM (Science, Technology, Engineering and Mathematics) courses. Platforms like Arduino and Raspberry Pi have demonstrated their ease of use and they fit for non-expert developers to build their own sensor network [15].

Other works have focused in the design of academic IoT platforms developed specifically for industrial applications. “GreenIoT”, a power-saving IoT platform is proposed by [16], based on a four-layer architecture: device, middleware, applications and cloud computing. An IoT solution for smart cities is presented by [17], which includes a survey of enabling technologies, communication protocols and system architecture designs, the system was deployed in Padova, Italy. “Lysis”, a cloud based platform for the deployment of IoT applications is described by [18], reusability was the main aspect considered in its design. In [19], an IoT architecture, namely “SenSquare” is introduced, with the capability to handle both the heterogeneous data sources coming from open IoT platform and crowd-sensing campaigns, and display a unified access to users. In [20], a practical architecture for the Industrial Internet of Things is proposed considering a design around a middleware protocol and compatible with industrial standards.

Finally, some other works presents extensive reviews on the IoT technology, that provide the current trends in this area. In [21] summarizes the current state-of-the art of Internet of Things in various domains systematically. In [22] analyzes the rol of the middleware and its impact in the IoT services. In [23] provides a brief survey of the state of the art in sensing services over cloud-centric IoT, and present recent research challenges in this area. In [24] surveys popular IoT cloud platforms in light of solving several service domains such as application development, device management, system management, heterogeneity management, data management, tools for analysis, deployment, monitoring, visualization, and research. In [25] introduces an experimental study of a reliable and self-configurable IoT communication gateways.

3. Materials and Methods

Often, each IoT platform provides its own cloud, its own services and its own features, creating a unique framework that forces the developer to learn how the vendor conceive Internet of Things, overcoming this situation is a hard task for people with networking and computer programming skills, and nearly impossible for people without such capabilities [19]. For this reason, the platform presented in this paper is based on a previous analysis of design features, communication patterns and performance parameters from a group of typical open-source platforms. This section analyzes the most common features provided by open-source platforms, then it presents the proposed architecture and introduces the detailed implementation of the Novus-io platform.

3.1 IoT Features

In order to define an IoT platform, different features should be considered as system specifications. For academic projects the design features, communication patterns and performance parameters were selected as the main parameters for a taxonomy analysis for open-source IoT platforms and compare them with the proposed platform implemented in this work. The design features refer to the different aspects considered during the platform design stage, and it is reflected by the quality and quantity of the services offered by the platform. The communication patterns are an indicative on how the data is exchanged between the different elements of the IoT platform, these patterns represent how data flows and the frequency in which the data is processed by the different IoT modules. Finally, the performance parameters represent a set of metrics that reports how well the platform is operating, including delays, latency, throughput and failures measurements. **Table 1**, describes the elements for each one of the main features types.

Table 1. IoT features description

Type	Feature	Concept
Design features	Openness	Supports different hardware platforms and communication protocols.
	Scalability	Supports a very large number of sensor and actuator devices and it is capable to retain data for long time.
	Usability	How easy to manage the IoT platform is for non-experienced users.
	Flexibility	How adaptable the IoT platform is for different types of applications.
Communication patterns	Data monitor	Pull-based data access where server collects data from devices usually at time intervals, also known as upstream data flow.
	Control action	Push-based data access where server sends data to devices immediately upon their availability, also known as downstream dataflow.
	Data stream	Data is sampled and transmitted at specific time rate with real-time requirements.
	Data analyze	Server has the capability to process data and convert them into relevant information for the user.
	Trigger event	Server has the capability to trigger an event automatically after specific conditions are met.

Performance features	Communication Delay	Total time spent from the device to send a message and receive an acknowledge from the processing layer.
	Data propagation latency	Total time elapsed from the moment when data is available on the server to the moment when a user is able to see the relevant visual interpretation of data.
	Message throughput rate	Total of messages sent by the device element to the server by a unit of time.
	Communication failure rate	Percentage of failed messages from the total number of messages sent from the device to the server.

3.2 IoT Architecture Platform

An IoT architecture must take into consideration the interaction between physical things, sensors, actuators and cloud services using specific protocols and communication layers in order to satisfy the application model. For academic projects, the application domain may be very diverse, since it depends on the developer area of interest, so the design approach was to keep the architecture as general as possible.

A typical IoT platform may be comprised by three ([12], [13], [19], [26]) or four layers ([16], [18], [20], [27]) depending of the complexity and scope of the application. For this work, a three level architecture is considered as a reference, which includes the device, computing and application layers, as showed in Fig. 1.

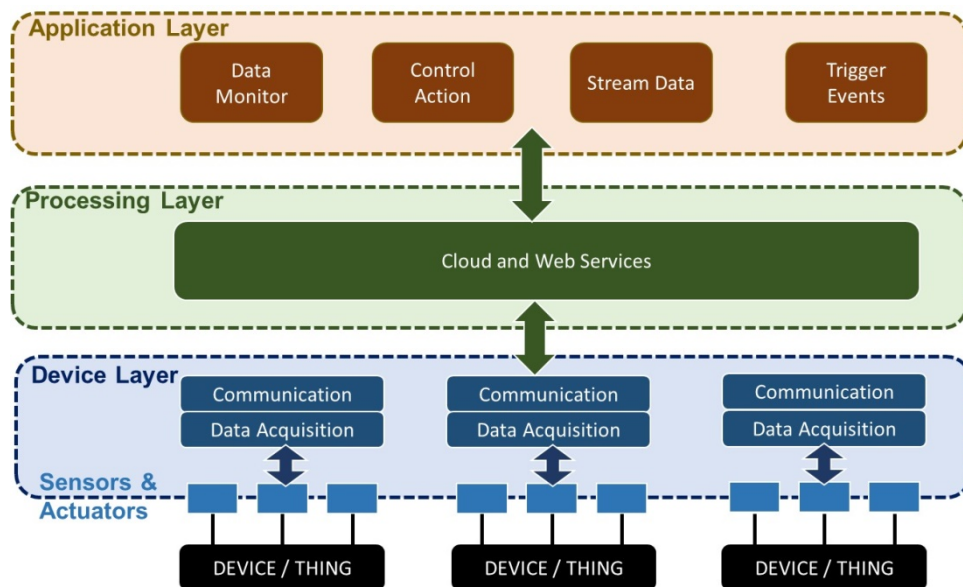


Fig. 1. A three-layer IoT architecture platform.

Each layer conducts specific activities:

- The device layer collects data from the sensors and, if required, activates the actuator devices. The elements in the device layer are close to the objects or things to be monitored and/or controlled. The device layer elements communicate to the processing layer by using a specific communication protocol. In fact, most of the four layer

platforms, divide this layer in two, one that acquire data and another to communicate the data.

- The processing layer contains the cloud and web services, where raw data from the device layer is processed in order to obtain relevant information; it supports different communication protocols and acts as a data server by exchanging messages with both the application and the device layers.
- The application layer is integrated by a collection of web-based or mobile applications from where the user can interact with the system. It usually contains data visualization elements such as charts and gauges, also it may present stream data, e.g., video, and input elements to generate control action messages by the user. Most of the platforms also provides specific events that may be triggered after specific conditions are met, e.g., alarms.

3.3 Platform Implementation

For the implementation of the Novus-io platform, three main modules were developed corresponding to the three layers of the system architecture used as a reference. The web application, the processing server and data acquisition correspond to the application layer, processing layer and device layer respectively, as shown in Fig. 2.

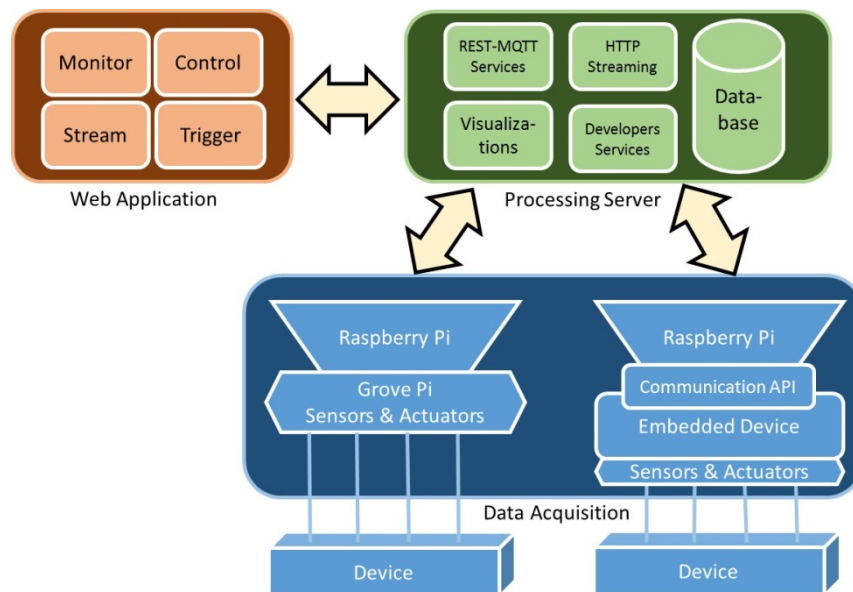


Fig. 2. Novus-io platform implementation.

The data acquisition is based on the Raspberry Pi (<http://www.raspberry.org>), a low-cost single-board computer, commonly used as an open-source IoT hardware module, as mentioned on Section 2. Alternatively embedded devices can be used with the Raspberry Pi.

The processing server provides the communication services in order to exchange messages with the data acquisition module. The processing server supports REST, MQTT and HTTP Streaming services.

- The Representational State Transfer (REST) is a client-server communication protocol built over HTTP (Hypertext Transfer Protocol), the foundation of data communication for the web. REST provides the means to implement a request-acknowledge type of

communication by using operations such as GET, POST, PUT and DELETE. Typically, REST is used to implement upstream messages or pull-based data access for IoT monitor applications.

- The Message Queue Telemetry Transport (MQTT) is a publish-subscribe communication protocol that enables the clients (devices) to subscribe on data associated with certain topic and receive the data provided by the subscriber (server). MQTT is used to implement downstream messages or push-based data access for IoT control applications.
- HTTP Streaming is a push-style data transfer technique that allows a web server continuously sending data to a client over a single HTTP connection that remains open indefinitely. HTTP Streaming provides a persistent message delivery connection that can be used in application such as JPEG video stream transmissions.

In addition, the server includes a set of services for the developers in order to configure the devices that connects to the data acquisition model, indicating the number and types of variables used for each device. Finally, the web application module provides the visualization capabilities for the system, i.e. the creation of charts, controls, indicators and any visual element accessed by the end-user. Detailed explanation for each module is described next.

3.4 Data Acquisition Module

There are two alternatives to implement the data acquisition module. For non-technical profile developers, a Grove Pi (<http://www.dexterindustries.com>) board complements the Raspberry Pi by providing an interface with sensors and actuators, since Raspberry Pi does not have such capability. Grove Pi allows the connection of a large group of sensors and actuators to the Raspberry Pi, and there is a vast group of software examples available at internet on how to read and write values from the sensors and actuators by using programs written in Python that runs on the Raspberry Pi.

For technical profile developers, that are capable to design and implement prototypes based on embedded devices, where a microcontroller directly interact with sensors and actuators to implement an embedded solution, they can enhance their embedded prototype by adding IoT capabilities. In this case, a communication API (application programming interface) is provided for the microcontroller in order to interact with the Raspberry Pi by using a set of subroutines to exchange messages between these two elements. Both solutions for Non-technical IoT profile developers and technical profile developers are shown in [Fig. 3](#).

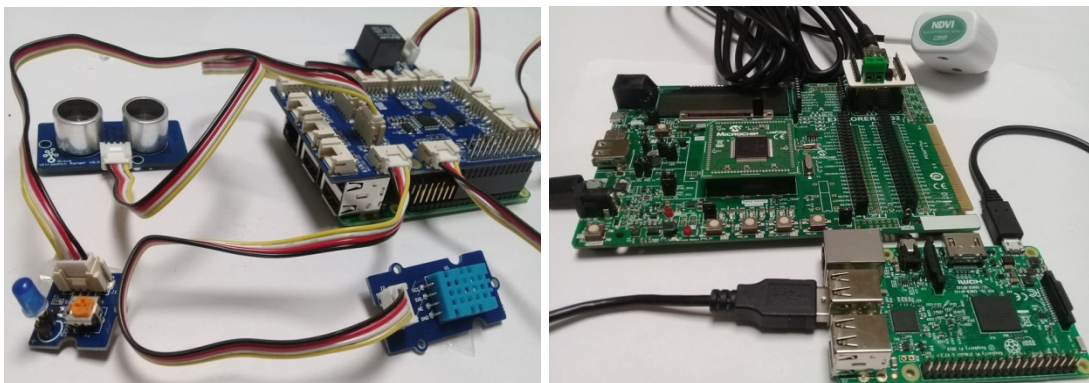


Fig. 3. Hardware platform for the data acquisition module using Raspberry Pi + Grove Pi (left), and Raspberry Pi + embedded device based on Microchip PIC24 microcontroller (right).

The software elements that integrates the data acquisition module are depicted in **Fig. 4**. For the Raspberry Pi + embedded device configuration, the Raspberry is periodically updated with sensor data by the embedded device via serial communication, then the communication interface sends and receives data from the processing server by using REST and MQTT access. For the Raspberry Pi + Grove Pi configuration, sensors and actuators data are accessed by using the Grove Pi interface which provides a set of Python functions to communicate with the Grove Pi Board through I2C protocol. The acquisition application uses REST and MQTT to communicate with the processing server through WiFi. In any case, a Raspberry Pi Camera can be connected to transmit stream video to the server by using the MJPG-streamer services which continuously sends JPEG frames via WiFi.

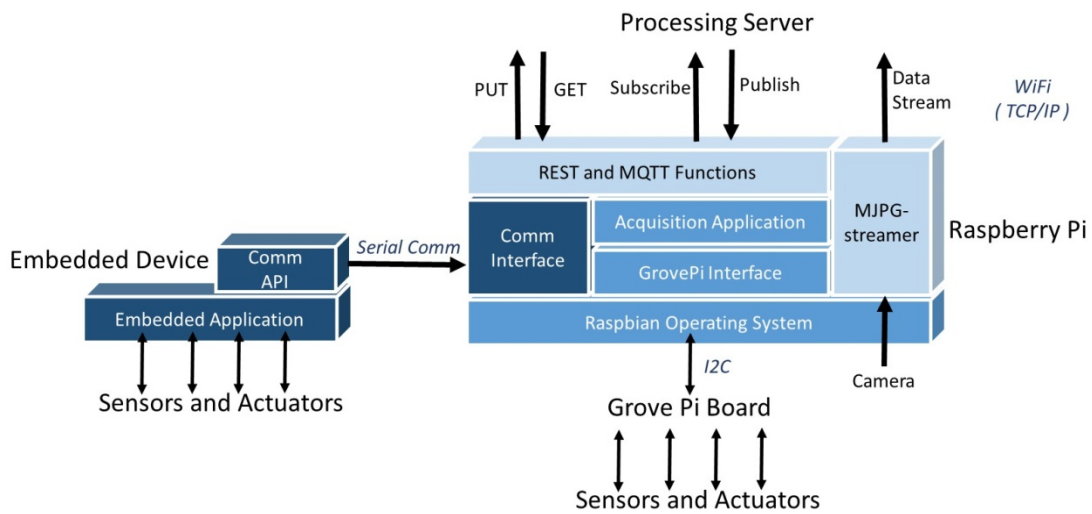


Fig. 4. Software elements for the data acquisition module.

The REST and MQTT functions element provides the communication access to the server by using a client-server pull-data scheme (REST) and a publisher-subscriber push-data mechanism (MQTT). Below there are two pieces of codes illustrating how communication is conducted from the data acquisition module perspective:

Raspberry REST Client Code	Raspberry MQTT Subscriber Code
<code>import requests</code>	<code>import mqtt</code>
<code>requests.post(SERVER,data)</code>	<code>client=mqtt.Client()</code>
<code>...</code>	<code>client.connect(MQTT_SERVER,port)</code>
<code>r=requests.get(SERVER,data)</code>	<code>client.subscribe(topic,callback)</code>
	<code>client.wait()</code>
	<code>client.close()</code>

For REST communication the Raspberry becomes the client that can post data to the server or get updated with information from it, usually the data refers to values provided by the different sensors that are being monitored. For the MQTT communication the Raspberry subscribes to the server for a specific topic, then when the server receives a command, it publishes this information and the Raspberry executes a callback function to process the command, usually in order to update an actuator device.

3.5 Processing Server Module

This module is implemented in a small computer (Dell Optiplex 8020, Core Intel i5-4570 @ 3.2 GHz, 8GB RAM and 457 GB Hard disk) running Ubuntu as an operating system; for portability the open platform Docker (<https://www.docker.com>) is used as a container for the developed software, to provide an independence between applications and infrastructure. Novus-io platform was built over the open-source web framework ASP.NET Core (<https://www.aspnet.net/>). This framework provides all the necessary elements to create an IoT server, which includes the database, the web services, HTTP streaming and the IoT typical communication services like REST server and MQTT broker. As shown in Fig. 5, Novus-io software integrates these IoT services and provides the basis for the web applications available to the end-user, and also it gives to the developer the proper services for account management and channel (device) configuration.

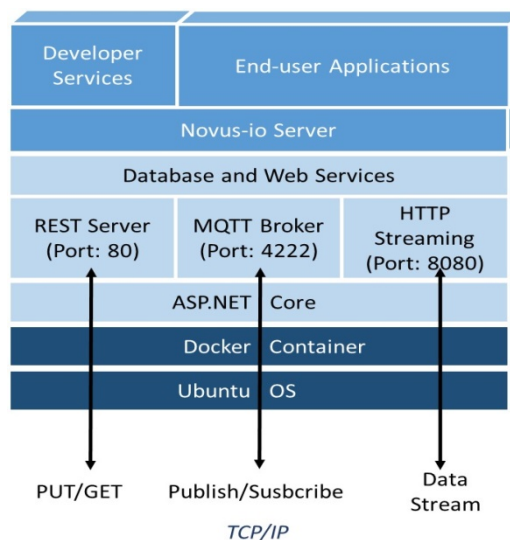


Fig. 5. Software elements for the processing module.

TCP (Transmission Control Protocol) is used to communicate the processing server with the data acquisition module, since it provides a reliable connection, in contrast to UDP (User Datagram Protocol) that uses a best effort connectionless protocol. However TCP may introduce large delays whenever packet loss occurs, these delays are caused by packet retransmissions. REST and MQTT communication are not strongly affected since they are lightweight protocols and as it will be showed in the performance evaluation subsection the communication delays presented by Novus-io are minimum in comparison to the open-source IoT platform. For HTTP streaming, since it requires intensive packets transmissions large delays may occur if network load is high, in order to reduce this problem the MJPG-streamer software can be adjusted to reduce the number of frames per second transmitted.

The detailed sequence of events executed for a data monitor communication task is depicted in Fig. 6. The data monitor is a synchronous data delivery based on a pull data access. The Raspberry Pi as the main element of the data acquisition module periodically initiates the monitor task by autonomously executing a time-based scheduled job (cron process). For the Grove Pi based platform, the Raspberry read the sensors and then post the data via REST services to the processing server; later the visualization element receives the data and update

the monitor application and the database simultaneously. When an embedded device is used, the Raspberry Pi interrupts the microcontroller via serial communication using the API subroutines provided by the platform. Then, the microcontroller’s interrupt service routine reads the sensors specified by the Raspberry Pi and sends a message with their current values. Later, the Raspberry Pi post the data to the processing server meanwhile the microcontroller resumes its normal operation. The implementation of this schema simplifies the access to IoT services for microcontroller-based prototypes, in this way embedded system prototypes can be easily upgraded with IoT functionalities.

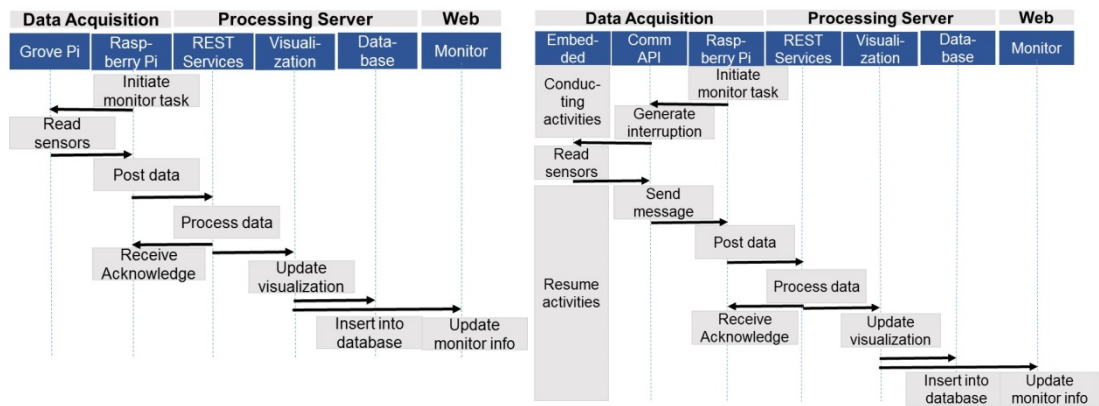


Fig. 6. Data monitor sequence diagram for Grove Pi (left) and embedded device (right).

The Fig. 7 shows the detailed sequence of events executed by the control action task. End-user initiates the control action by providing input-data in the web application, as an event-driven task, with the aim to modify either an actuator or other relevant device parameter. As an asynchronous data delivery using a push data access model the MQTT services are used. The Raspberry Pi subscribes to the processing server, then the processing server delivers the message containing the command to modify an actuator or a parameter. When the Raspberry Pi receives the action order, it can either updates the actuator connected to the Grove Pi board or generate an interrupt signal for the embedded device, depending on the type of configuration used by the developer. Later, the visualization model and the database are updated with the modified value. Finally, the sequence is completed with an acknowledge message.

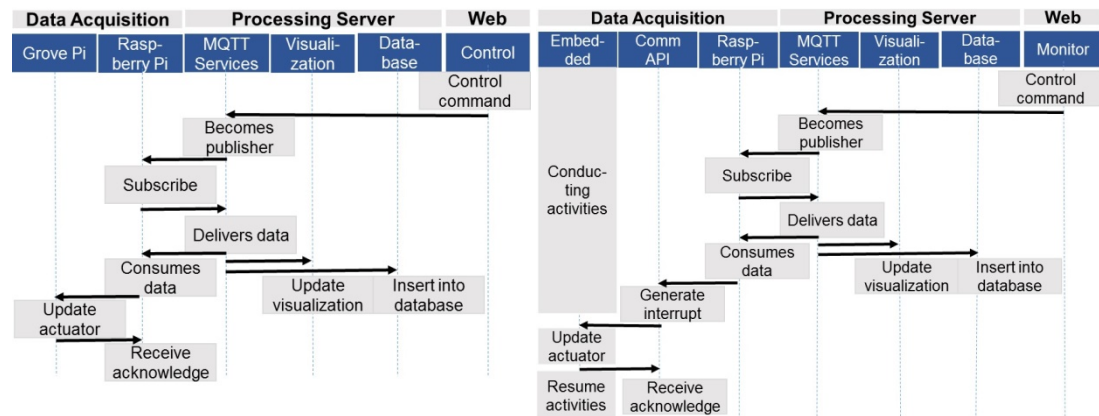


Fig. 7. Control action sequence diagram for Grove Pi (left) and embedded device (right).

3.6 Web Application Module

Novus-io provides a web interface in order to manage the developer’s access accounts. Each developer can create one or more channels to create an IoT application, each channel contains a group of variables that correspond to a specific device or thing, see Fig. 8. From the end-user perspective, the web applications offered by the platform are grouped in four categories: data monitor, control action, stream data and trigger events. Screenshot samples for monitor and control applications are shown in Fig. 9.

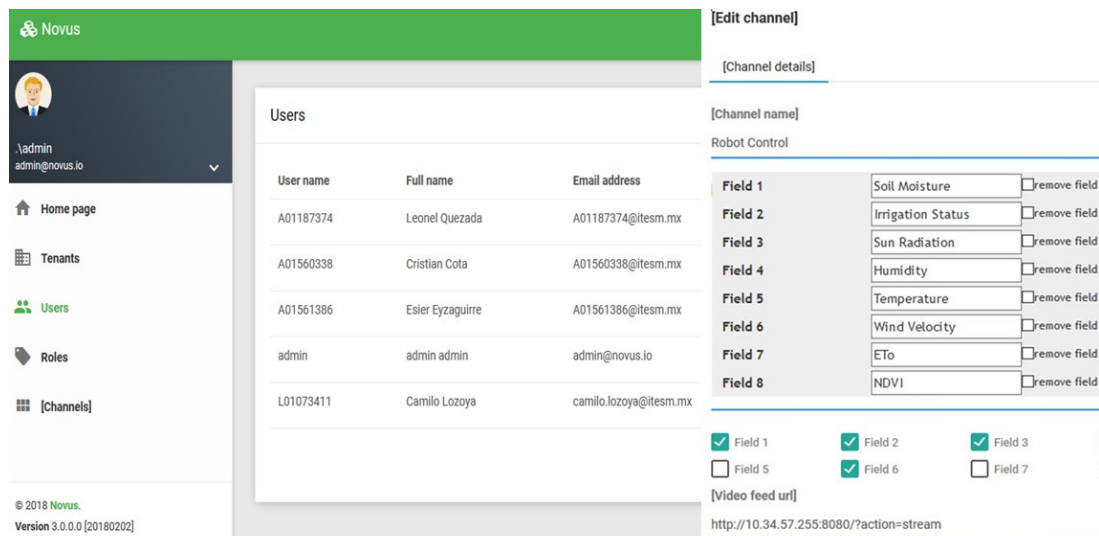


Fig. 8. Developers access account (left) and channel configuration (right) provided by Novus-io

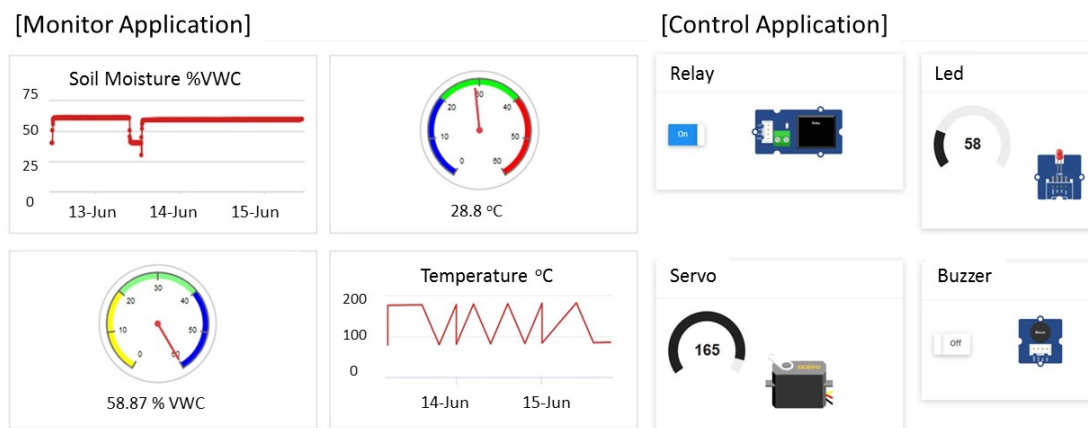


Fig. 9. Monitor application (left) and control application (right) provided by Novus-io

- Monitor application: refers to the visualization elements such as time-based charts (line, bar, column), gauges, text, buttons, and indicators in order to visualize the main variables of the IoT application.

- Control application: refers to the data-entry elements, which allows the end-user to send downstream commands to the IoT application in order to update actuators or other device variables. The data-entry can be conducted by using knobs, text, slider and button elements.
- Stream application: refers to the applications that requires data at a specific time rate, for the implementation of the Novus-io platform, a stream video visualization is included in order to have real-time access for video broadcast.
- Trigger application: refers to the activation of a specific event, after certain conditions are met. In Novus-io, a trigger event in the form of an alarm is implemented, where an event is dispatched a value of a specific variable is either above or below certain threshold.

4. Results and Discussion

Novus-io platform is currently supporting 35 developed applications for an approximate scope of 150 end-users. The university campus has a total population of 2,000 students, according to the performance evaluation conducted, the current implementation may support more than 500 applications and 2,000 devices running simultaneously, to cover student population. For network security reasons, Novus-io is running in the campus intranet, any external access is only possible through VPN (Virtual Private Network) authentication. The platform was implemented over current network infrastructure, the only additional cost is the computer that hosts Novus-io which is approximately \$900 US dollars. The cost per individual license for the full versions of the open-source platforms is in the range of \$60 to \$100 US dollars per year, therefore 15 licenses covers the full cost of the Novus-io computer. No additional software licenses were required since only open-source software was used for Novus-io implementation.

In order to validate Novus-io as a capable platform to support academic projects, the following analysis were conducted. First, Novus-io was technically compared with the open-source platforms commonly used students in their projects and that are used as references. Then, a performance evaluation was conducted to compare Novus-io with these other platforms. Finally, student's feedback information was gathered in order to analyze the perception from the developers regarding the proposed platform. It is important to mention that the open-source platforms considered for the analysis were only the free versions, that contains several operational constraints but are available to the developers.

4.1 Comparative Analysis

An analysis of IoT features was conducted in order to compare Novus.io with some relevant open-source IoT platforms. **Table 2** summarizes the obtained results after inspecting the available platforms with Novus-io. The features selected were: available communication protocols, supported hardware boards, data retention time and common communication patterns.

Every evaluated platform supports REST and MQTT protocols, since they are the most commonly used in IoT applications. In addition Novus-io, Freeboard and Thinger.io offers HTTP Streaming to provide video broadcast capability in the application. This is a well-valued feature by the users, since it allows them to observe the device dynamics.

Table 2. Comparative analysis of evaluated open source IoT Platforms

Platform	Supported Protocols	Supported Hardware	Data Retention	Communication patterns				
				Monitor	Control	Stream	Analysis	Trigger
ThingSpeak [28]	REST, MQTT	ESP8266, Raspberry Pi, Arduino	Unlimited	X	X		X	X
Adafruit IO [29]	REST, MQTT	ESP8266, ESP32, Raspberry Pi, Arduino	30 days	X				X
Free board [30]	REST, MQTT, HTTP Stream	ESP8266, Raspberry Pi, Arduino	30 days	X		X		X
Thingier.io [31]	REST, MQTT, Websockets, HTTP Stream	Inte Edison, ARM Mbed, Raspberry Pi, Arduino	365 days	X	X	X		X
Things Board [32]	REST, CoAP, MQTT	ESP8266, Raspberry Pi, Arduino	30 days	X	X			X
Novus-io	REST, MQTT, HTTP Stream	Microchip PIC24, Raspberry Pi, Arduino	Unlimited	X	X	X		X

For the supported hardware, Raspberry Pi and Arduino are the most popular boards that can be integrated straightforward by every evaluated IoT platform. This confirms, that these two hardware platforms as the most popular for IoT applications. For Novus-io, an API is provided to support embedded systems, specifically the ones based on Microchip PIC24 microcontroller, in this way academic projects based on this board can be easily enhanced with IoT features.

Data retention is a major constraint on current IoT open source platform (free versions), this is why Novus.io was designed with no restriction in this aspect. It is understandable that open source platforms has limitation on data storage since they are used by thousands of developers, so by implementing a local solution for a small group of developers (students) the constraints in this matter can be easily eliminated.

Finally, for the communication patterns, the monitor and trigger patterns are offered by every platform, however control action is only supported by a few. Control actions gives the user the possibility to modify and interact directly with the decisions taken by the IoT application. Freeboard and Adafruit IO does not support this feature, maybe in order to keep the simplicity in these platforms. The stream data pattern provides the stream video capability not very often available on IoT platforms. In general, it can be observed that Novus-io and Thingier.io offers the most relevant communication patterns. Novus.io, in this first release is not offering analytical capability, although it is a very important feature, for academic projects it not used very often.

4.3 Performance Evaluation

A performance evaluation was conducted on the IoT platforms in order to measure the communication delay, the data propagation latency, the communication failure rate, and the message throughput. These parameters were identified as the main performance features by an IoT application in Subsection 3.2. The aim of this evaluation is to identify if the proposed Novus-io platform provides an acceptable performance for the students in comparison to the commercial open-source IoT platforms, i.e. the students when selecting Novus-io overcomes the constraints presented by these available platforms.

It is important to highlight that this evaluation platform cannot be used as a general benchmark evaluation for the IoT platforms, since the performance was conducted using the university network infrastructure and by a relatively small group of developers. The Novus-io platform is a local solution for the university students and does not pretend to be a global

solution for any academic project from any university. The performance evaluation was executed using only REST messages.

On [Fig. 10](#), the communication delay and data propagation latency results are presented. As observed, Novus-io platform presents the lowest communication time delay, as expected since the server that hosts the platform is just dedicated to a small group of students from the university, however it is important to mention that the communication delay in any case is using internet, no intranet network was used. The data propagation delay for Novus-io although is acceptable, it is not the lowest one, this means that graphics render process developed in the platform can be improved. All other platforms, with the exception of ThingSpeak, have similar latencies.

On [Fig. 11](#), communication failure rate and message throughput results are presented. Novus-io has a significant low failure rate, this confirms that the proposed platform uses a robust communication mechanism and with the advantage of using a dedicated server, the developers may not be worried about losing data for their academic projects. Message throughput is probably the most important constraint of the commercial open-source IoT platforms, in fact every reference platforms, with the exception of ThingsBoard, imposes limits in either the number of messages or the messages update interval, this is in order to offer their services to a great amount of developers, but also to obtain a profit when a developer needs more resources. Novus-io provides to students a platform with a very large message throughput, since no restrictions are imposed by having a host dedicated to them.

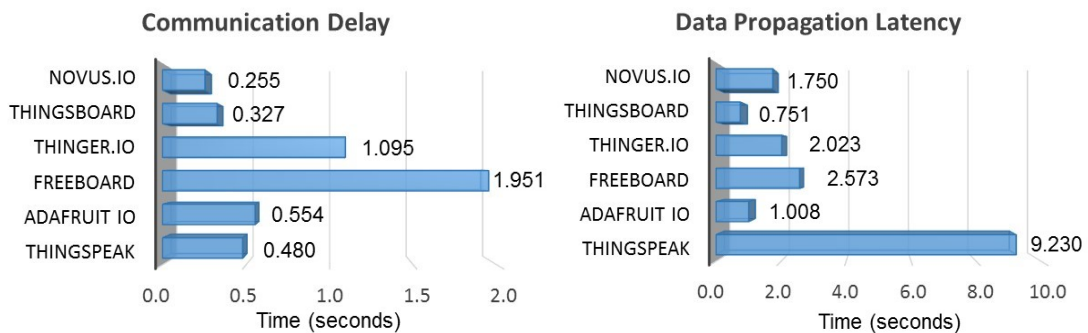


Fig. 10. Performance evaluation on communication delay (left) and data propagation latency (right) for the evaluated IoT open source platforms.

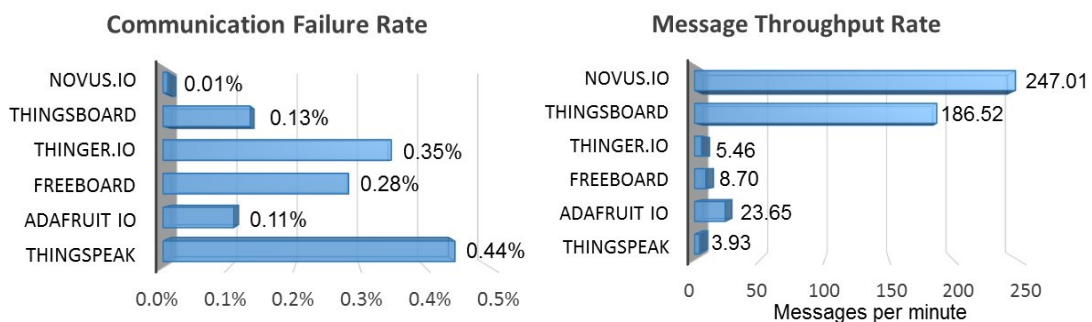


Fig. 11. Performance evaluation on communication failure rate (left) and message throughput rate (right) for the evaluated IoT open source platforms.

In general, Novus.io provides at least similar performance results in comparison with the open-source IoT platforms, this validates that a small local IoT platform can be capable to support students developers for academic projects. ThingsBoard platform also obtained good performance results.

4.4 Students' Feedback

A total of 107 students were trained into the Novus-io platform and were prepared to develop IoT projects in teams from 3 to 5 members. Two main group of students were conformed, one with technical profile in IoT, and the other with non-technical profile. Students with good knowledge on information technology and/or electronics formed the IoT technical profile group; these students are enrolled in the Mechatronics and Computer Engineering majors. The non-technical profile group was conformed by students with none to basic knowledge on information technology and/or electronics. These students are enrolled in other engineering majors such as Industrial and Mechanical Engineering, but also in Humanistic and Business majors. A total of 65 technical profile students and 42 non-technical profiles students participated in the experiment.

The technical group was trained on advanced IoT, where they learn how to program the Raspberry Pi to communicate with the server using different protocols, the advantages and disadvantages of each protocol; they exercise the different communication patterns. Also, they learned how to use the communication API to exchange message with an embedded device. At the end, they took an academic project previously developed by them in the embedded course, and integrated the IoT features learned in the workshop. The workshop aim was to give the students the tools to transform an embedded application into an IoT application.

The non-technical group was trained on basic IoT where they learn how to use the Grove Pi board with simple sensors and actuators, they do not get into the deeper details of the protocols, they just learned how to make a basic program in the Raspberry Pi and send data to the server. They use very basic control and monitor applications. In this case, the workshop aim was to show the students how to create a basic IoT application. They were encouraged to develop an application related to their major.

Once the workshop was concluded, the students were asked to provide feedback regarding their experience on this process, the questions asked to the students were:

- How much the workshop helps them to improve your knowledge on Internet of Things?
- How easy to use and accessible was the hardware platform presented in the workshop?
- How easy to use and accessible was the software platform presented in the workshop?
- How satisfied are you with the result of the IoT application that you developed?
- How useful do you consider the academic project that you developed in the workshop was?
- How much impact do you consider Internet of Things have in your career?

The result of this poll is presented in [Fig. 12](#), where the scale used considers a 5 as the maximum grade for the question, and zero as the lowest one. In general, the technical profile students considered the Novus.io platform as a very useful tool to enhance an academic project and they stated that IoT gives value-added to their projects. Although, the project satisfaction was well qualified, it obtained the lowest grade among the other questions. According to the student's comments, this was caused because they still feel that the platform can be improved with more functionalities, mostly on the monitor aspect of the platform, i.e. charts can be more dynamic, and data should be presented using more different types of charts.

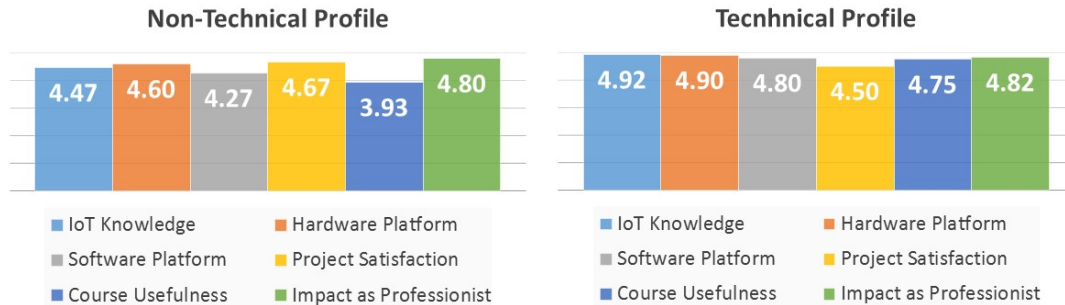


Fig. 12. Students' feedback after using the Novus.io platform.

The non-technical profile students, also liked the workshop, they thought the platform was easy to use and they were very satisfied with their academic project developed. However, they still feel that the usefulness of the course is not very clear. Going deeper into this matter, the students commented that the IoT project developed in the workshop was interesting, but it was not very related to their major or area of interest, specially the students with humanistic and business background. One suggestion on this issue was that a technical profile instructor gave the workshop, so the examples and activities were on the area of electronics and information technology, so probably if the instructor had a better knowledge of business and humanistic aspects, they could apply IoT to their specialization areas.

5. Conclusions

The work presented in this paper introduces Novus-io, an open-source Internet of Things (IoT) platform, designed specifically for undergraduate students where they can turn their academic projects into IoT applications providing value-added to their final product. The proposed platform was evaluated against other IoT platforms available on internet. The comparative analysis and the performance evaluation conducted, indicates that Novus.io is a suitable option for academic purposes since overcomes cost and operational constraints while it provides the common features offered by the other platforms. Students with IoT technical profile and students with non-technical profile in IoT, were trained on the Novus.io and then encouraged to develop an IoT project or to convert their current project into an IoT application. From the students' feedback, we may state that they like this technology and worldwide business trends indicate that IoT is a major feature in today's products and services, therefore IoT should be considered as a transversal competency for any undergraduate major.

Future works includes a continuous improvement of the Novus.io platform in order to keep the pace with the state of the art technology and provide the features that other platforms offer, for instance analytical capabilities. Also, it is important to continue involving undergraduate students from any major on IoT related projects with a background in their interest area. The student involvement can also include high schools and middle schools, but it is also important to train teachers in this technology.

References

- [1] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything," April 2011. [Online]. Available: [Article \(CrossRef Link\)](#). [Accessed 05 2018].
- [2] McKinsey Global Institute, "Internet of Things: How to capture the value of IoT," 2017. [Online]. Available: [Article \(CrossRef Link\)](#). [Accessed 04 2018].
- [3] F. Corno, L. De Russis and D. Bonino, "Educating Internet of Things Professionals: The Ambient Intelligence Course," *IT Professional*, vol. 18, no. 6, pp. 50-57, 2016. [Article \(CrossRef Link\)](#)
- [4] G. Kortuem, A. K. Bandara, N. Smith, M. Richards and M. Petre, "Educating the Internet-of-Things Generation," *Computer*, vol. 46, no. 2, pp. 53-61, 2013. [Article \(CrossRef Link\)](#)
- [5] T. Svane, Z. Minling and L.-O. Johansson, "Enchanting education from student input preparing students to envision and develop in an internet of things world," in *Proc. of IEEE Frontiers in Education Conference (FIE)*, Indianapolis, USA, 2017. [Article \(CrossRef Link\)](#)
- [6] D. Namiot, M. S.-S. Ventpils and Y. I. Daradkeh, "On internet of things education," in *Proc. of 2017 20th Conference of Open Innovations Association (FRUCT)*, St. Petersburg, 2017. [Article \(CrossRef Link\)](#)
- [7] T. Islam, S. C. Mukhopadhyay and N. K. Suryadevara, "Smart Sensors and Internet of Things: A Postgraduate Paper," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 577-584, 2017. [Article \(CrossRef Link\)](#)
- [8] J. Rios, F. Mas, M. Marcos, C. Villa, D. Ugarte and T. Chevrot, "Accelerating the Adoption of Industry 4.0 Supporting Technologies in Manufacturing Engineering Courses," *Materials Science Forum*, vol. 903, pp. 100-117, 2017. [Article \(CrossRef Link\)](#)
- [9] I. K. Song and J. Chun, "Evolving Internet Information & Technology as Enablers for Creating Shared Values," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 1, pp. 309-3017, 2015. [Article \(CrossRef Link\)](#)
- [10] M. Bagheri and S. H. Movahed, "The Effect of the Internet of Things (IoT) on Education Business Model," in *Proc. of 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Naples, 2016. [Article \(CrossRef Link\)](#)
- [11] V. Uskov, A. Pandey, J. P. Bakken and V. S. Margapuri, "Smart engineering education: The ontology of Internet-of-Things applications," in *Proc. of 2016 IEEE Global Engineering Education Conference (EDUCON)*, Abu Dhabi, 2016. [Article \(CrossRef Link\)](#)
- [12] D. K. Fisher, R. S. Fletcher, S. S. Anapalli and H. Pringle III, "Development of an open-source cloud-connected sensor-monitoring platform," *Advances in Internet of Things*, vol. 8, pp. 1-11, 2018. [Article \(CrossRef Link\)](#)
- [13] A. Elsts, G. Oikonomou and X. Fafoutis, "Internet of Things for smart homes: Lessons learned from the SPHERE case study," in *Proc. of 2017 Global Internet of Things Summit (GIoTS)*, Geneva, Switzerland, 2017. [Article \(CrossRef Link\)](#)
- [14] J. He, D. C.-T. Lo, Y. Xie and J. Lartigue, "Integrating Internet of Things (IoT) into STEM Undergraduate Education: Case Study of a Modern Technology Infused Courseware for Embedded System Course," in *Proc. of 2016 IEEE Frontiers in Education Conference (FIE)*, Erie, PA, USA, 2016. [Article \(CrossRef Link\)](#)
- [15] F. Montori, L. Bedogni and L. Bononi, "On the integration of heterogeneous data sources for the collaborative Internet of Things," in *Proc. of 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, Bologna, 2016. [Article \(CrossRef Link\)](#)
- [16] Y.-W. Ma, J.-L. Chen, Y.-S. Lee and H.-U. Chang, "GreenIoT Architecture for Internet of Things Applications," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 2, pp. 444-461, 2016. [Article \(CrossRef Link\)](#)
- [17] A. Zanell, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, 2014. [Article \(CrossRef Link\)](#)

- [18] R. Girau, S. Martis and L. Atzori, "Lysis: A Platform for IoT Distributed Applications Over Socially Connected Objects," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 40-51, 2017. [Article \(CrossRef Link\)](#)
- [19] F. Montori, L. Bedogni and L. Bononi, "A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1-1, 2017. [Article \(CrossRef Link\)](#)
- [20] I. Ungurean, N. C. Gaitan and V. G. Gaitan, "A Middleware Based Architecture for the Industrial Internet of Things," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 7, pp. 2874-2891, 2016. [Article \(CrossRef Link\)](#)
- [21] P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University - Computer and Information Science*, 2016. [Article \(CrossRef Link\)](#)
- [22] A. Farahzadi, P. Shams, J. Rezazadeh and R. Farahbakhsh, "Middleware technologies for cloud of things - a survey," *Digital Communications and Networks*, 2017. [Article \(CrossRef Link\)](#)
- [23] B. Kantarci and H. T. Mouftah, "Sensing Services in Cloud-Centric Internet of Things: A Survey, taxonomy and challenges," in *Proc. of 2015 IEEE International Conference on Communication Workshop (ICCW)*, London, 2015. [Article \(CrossRef Link\)](#)
- [24] P. P. Ray, "A survey of IoT cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1-2, pp. 35-46, 2016. [Article \(CrossRef Link\)](#)
- [25] B. Kang and H. Choo, "An experimental study of a reliable IoT gateway," *ICT Express*, 2017. [Article \(CrossRef Link\)](#)
- [26] D. Amaxilatis, O. Akrivopoulos, G. Mylonas and I. Chatzigiannakis, "An IoT-Based Solution for Monitoring a Fleet of Educational Buildings Focusing on Energy Efficiency," *Sensors (Basel)*, vol. 17, no. 10, 2017. [Article \(CrossRef Link\)](#)
- [27] Z. B. Babovic, J. Protic and V. Milutinovic, "Web Performance Evaluation for Internet of Things Applications," *IEEE Access*, vol. 4, no. 1, pp. 6974-6992, 2016. [Article \(CrossRef Link\)](#)
- [28] "IoT Analytics - ThingSpeak Internet of Things," The MathWorks, Inc., 2018. [Online]. Available: [Article \(CrossRef Link\)](#) [Accessed June 2018].
- [29] "Welcome to Adafruit IO," Adafruit IO, [Online]. Available: [Article \(CrossRef Link\)](#). [Accessed June 2018].
- [30] "freeboard - Dashboards for Internet of Things," Bug Labs, Inc., [Online]. Available: [Article \(CrossRef Link\)](#). [Accessed June 2018].
- [31] "Thingier.io – Open Source IoT Platform," Thingier.io, 2018. [Online]. Available: [Article \(CrossRef Link\)](#). [Accessed June 2018].
- [32] "ThingsBoard - Open-source IoT Platform," The ThingsBoard Authors, 2018. [Online]. Available: [Article \(CrossRef Link\)](#). [Accessed June 2018].



Camilo Lozoya received the degree in Electronics Engineering in 1993 from Chihuahua Institute of Technology (Chihuahua, Mexico). He obtained a M.S. degree in 1996 in Electronic Engineering from the Tecnológico de Monterrey (Monterrey, México). He obtained his Ph.D. degree in 2011 in Automatic Control and Robotics from the Technical University of Catalonia (Barcelona, Spain). He is a Professor in the Department of Mechatronics, and member of the Sensors and Devices Research Group at the School of Engineering and Science of the Tecnológico de Monterrey. His research interests include real-time control systems, wireless sensor networks, internet of things, embedded systems and industrial automation.



Alberto Aguilar-Gonzalez is Dean of the School of Engineering and Sciences, and researcher at Tecnológico de Monterrey Campus Chihuahua. He has a doctorate in computer science and is involved in several technology and mobile related projects in the institution, working as leader, supporter or advisor. His research interests include P2P, IoT and sensor networks. Also, he advises students on the creation of Startups using the Lean Startup Methodology; he is as a founder of two startups. In 2012 and 2013, he was awarded with an Innovation Award from Tecnológico de Monterrey for best practices on using technologies to enforce learning.



Antonio Favela-Contreras received the B.S. and M.E. degrees in electronics and communications engineering and the M.S. degree in control engineering, all the three degrees from the Tecnológico de Monterrey, Campus Monterrey, Mexico in 1986, 1995 and 1991 respectively. In 1999, he received the Ph.D. degree in automatic control from the National Polytechnic Institute of Grenoble, France. He is a Professor in the Department of Mechatronics and Electrical Engineering, member of the Sensors and Devices Research Group at the School of Engineering and Science of the Tecnológico de Monterrey, member of the Mexican National Research System SNI-1 and member of the Mexican Academy of Science. His main research interests include adaptive and predictive control, analysis and control of hybrid systems, embedded control systems, control of mechatronics systems and industrial process control.



Arturo Zamora received the B.S. degree in Information Technologies and Communications Engineering in 2017 from Tecnológico de Monterrey. He founded in 2018, the software solutions company named BuscaMed. His professional interests are computer networks and internet of things.