ISBN: 978-628-95207-4-3. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2023.1.1.729

Cloud-based IoT solution to control and monitor a psychrometric laboratory in a Peruvian University

Stuardo Lucho, M.Sc.[1], Anthony Garcia, B.Sc.[2], Angelo Velarde, M.Sc.[1], and Diego Pichilingue, Ing.[1]
[1]Departamento de Ingeniería, Pontificia Universidad Católica del Perú, Lima, Perú,
stuardo.lucho@pucp.edu.pe, angelo.velarde@pucp.pe, depichilingue@pucp.edu.pe
[2]Facultad de Ciencias e Ingeniería, Pontificia Universidad Católica del Perú, Lima, Perú, garcia.josea@pucp.edu.pe

Abstract*– During the years of the COVID-19 pandemic, university education had to adapt to a virtual context and the laboratories, which made use of specialized equipment, had to stop and move to a simulated context, providing an experience for the student close to reality, but not complete. As the years progressed and the restrictions began to diminish, various alternatives were proposed to facilitate remote access by students to the laboratories through cameras, IoT solutions for the remote control and monitoring of specialized equipment through web environments. The Pontificia Universidad Católica del Perú has a psychrometric equipment for laboratories of undergraduate courses, which could only be used in person. In this article, an IoT solution deployed in the cloud is proposed, with AWS services, to remotely control and monitor all the data generated by the psychometric equipment. These data are obtained through sensors or cameras to be sent to the IoT platform in the cloud through MQTT, which will then allow users to see the result in real time using HTTPs and WebSocket. The deployed solution also allows managing user access by role, by date, assigning sensors to users in real time, saving a history of everything sensed, and exporting all the data generated during the laboratory session. The solution has also been used throughout a period of 12 months in which students have been able to interact with the platform, both in person and remotely.

***Keywords*—IoT, Internet of Things, Remote Laboratories, Psychrometric laboratory, MQTT**

I. INTRODUCTION

The Pontifical Catholic University of Peru has a psychrometric testing bench, which is an educational installation that simulates the air conditioning process and allows the understanding and analysis of psychrometric processes through experiments [1]. This equipment is in the Energy Laboratory of the Mechanical Engineering Department. This equipment is used by approximately 200 students in the laboratory activities, in courses like “General Thermodynamics”, “Heat Transfer” and “Thermodynamics 2”. However, due to the mechanical characteristics of the equipment, students must be present to interact with it.

For this reason, a simulator was built that allowed the students to continue carrying out the laboratory sessions but in a virtual and simulated way. However, according to the head of the Energy Laboratory, the challenge was to stop using simulators and build a solution where the students were able to control the equipment remotely from their homes or from anywhere else connected to the Internet.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

This poses 2 problems, firstly, the equipment needs to be connected to the internet through sensors, software, and other technologies to facilitate the exchange of data with other devices. Secondly, the student needs to be able to monitor and control the equipment remotely. Both problems are solved through an IoT platform.

Thus, the work shown in the present work focuses on designing and implementing a cloud based IoT solution that allows monitoring and control of the psychrometric equipment of an energy laboratory using a web interface in real time. In this process, 36 variables between inputs and outputs were defined, with their respective requirements and ranges, including cases in which a direct output for instrumentation through a sensor could not be provided and therefore required the placement of a camera. (e.g., a camera that allows distinguishing levels of refrigerant fluid), all completely integrated with the platform.

The system allows role management, date and time access control, real-time visualization of the sensors and actuators of the psychometric equipment using the HTTPs, WebSocket [2] and MQTT [3] protocols. For this reason, a customized platform deployed in the AWS cloud was developed and then the corresponding performance tests were carried out.

It is important to mention that, although the solution was proposed for a completely virtual environment, after the return to face-to-face activities, and therefore, the return to the classroom and laboratories, the project continues to be used by students and professors, as it allows not only the variable’s measurement, but also, the historical recording capability and exportation of data in *.CSV format. This last capability is making it easier for the student to resume sessions, analyze the data after the laboratory session ended and in the event of their absence for any reason, they could be able to interact with the equipment remotely and asynchronously using all the capabilities of the psychometric equipment.

The rest of the article is organized as follows: Section 2 describes all the remote laboratories related work and their relationship with the IoT solutions. Section 3 describes the system requirements, the system architecture and the components involved. Section 4 shows considerations for the cloud deployment, testing, and all the associated costs in capex and opex. Section 5 shows a discussion about what was developed and findings, and finally section 6 shows the work conclusions.

II. RELATED WORK

Various publications of educational remote laboratories have been found in the literature, which are managed through a web interface using the WebSocket protocol for real time updates. In [4] and [5] the control of mechanical devices is shown using NodeJS as backend and socket.io for bidirectional and real-time communication between the client (students) and the server. In [6] and [7] remote labs are presented using role access control and MySQL for the persistence layer. In [7] and [8], the use of Arduinos is made for the control and monitoring of remote laboratories and a web server with HTML and JavaScript supported by asynchronous communication through AJAX; while in [9] the created platform uses MQTT for communication between the components and allows remote reservation of a device by schedule, all managed from the cloud, using Arduinos together with Raspberry PI for remote control.

On the other hand, although it is not used for an educational environment, in [10] an interesting architecture is proposed, since it controls and monitors a mechanical device using NodeJS as a server, Arduino to control the electronic/mechanical part and the MQTT protocol for communication between the Arduino and the server.

It is important to mention that 4 works [4], [7], [11], [12] have been found for educational environments, where they complement the remote laboratory experience with cameras, so that students can see what is happening in real time, in some cases the environment, the equipment or a particular sensor.

All the cited investigations propose architectures using different components at the software level; however, other articles propose solutions using existing software as tools for remote laboratories implementations, such as [13], where LabView is used to control and monitor a DC motor, or [14], where Anydesk is used to connect to a local server and control robotic arms from it, including, the use of cameras to view what is happening in the laboratory in real time.

No papers have been found in the literature where there is a solution that integrates: Access management based on dates and roles, use of the same equipment concurrently by multiple students, real-time assignment of control variables to students, history of values from past experiences, and availability anywhere in the world. world through a cloud provider.

It is also good to consider that there are commercial solutions whose price is much higher or more industrial components such as Portenta [15], Revolutionpi [16] or even PLCs, but whose cost is higher in components than the solution proposed in this research.

III. SYSTEM DESIGN

This research work was developed to meet the requirements shown in Table I, considering that before the project, the only possible way to interact with the psychrometric equipment was in person. To meet these requirements, the architecture used (and its communication protocols) is shown in Fig. 1.

TABLE I
SYSTEM REQUIREMENTS

Requirements
Global access: in Peru there were still restrictions for attending universities in 2022, so the student had to be able to access the platform from anywhere.
Control and monitor all inputs and outputs of the psychrometric equipment.
Role-based access management: the course teachers had to be able to see all the elements to be controlled in the psychrometric laboratory; while the students could only see and control certain parts of the laboratory.
Date-based access management: students could only access the platform in certain ranges of days and hours (when it was their turn to carry out their laboratory experience in their courses).
Visualization in real time: the student must see the result of her actions and the rest of the students participating in the laboratory in real time.
Complementary visualization by cameras: some measurement points, such as a nanometer, which does not have a digital or electrical connection, had to be visualized by a camera.
Attendance: each laboratory session has 8 people connected simultaneously, 6 students and two teachers.
Management and export of the data generated: the solution should allow storing a history of the data generated by the equipment and be able to export it for offline analysis.

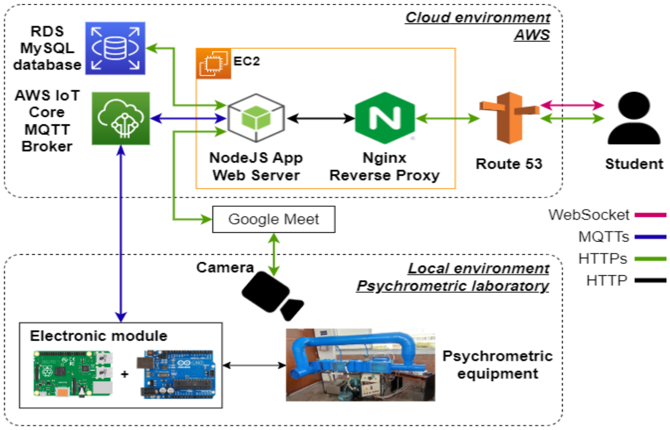


Fig. 1 Deployed solution architecture.

A. *Local environment - Psychrometric lab*

1) Psychrometric equipment:

The psychrometric equipment conditions air by increasing and decreasing temperature, as well as increasing and decreasing humidity. To achieve this, the equipment has 4 processes connected in series, measuring the air conditions before and after each process. The equipment has been logically divided into 5 measurement zones, as shown in Fig. 2, where each zone has subsystems, which are integrated and controlled with the proposed solution:

- Zone A: air at room temperature, which is driven by a fan.

- Zone B: air with a higher temperature than the initial one, due to the heat transfer of two heating resistors of 1 kW each.

- Zone C: air with a higher temperature and humidity than the previous one, due to an injection of steam, which is generated by the heater (with internal immersion heaters).

- Zone D: saturated air at a lower temperature than the previous one, due to the cooling of the refrigeration system.

- Zone E, air with a higher temperature compared to the previous one, again due to two heating resistances of 500 W each. An orifice plate type flowmeter is in this section.

Fig. 2 Psychrometric equipment.

To integrate the manual operation of the equipment, "variables" were defined that represent the values associated with the processes in the psychometric equipment, which will then be integrated into the platform through the Arduino as input and output variables.

It must be considered that, in this research work, the name "input" refers to a measured value and that it is used as input to the Arduino; while "output" is a value that the Arduino sends and enters the psychrometric equipment. 36 variables were defined (25 input and 11 output) as shown in Table II.

TABLE II

VARIABLES DEFINED IN THE SYSTEM

Type	Variable	Num. Variables
Input	Ambient temperature - Zone A, B, C, D and E	5
Input	Ambient humidity – Zone A, B, C, D, and E	5
Input	Electric current - Resistance 1, 2, 3 and 4	4
Input	Water level – heater	1
Input	Coolant Temperature - State 1, 2, 3 and 4	4
Input	Airflow - Differential Pressure at Orifice Plate	1
Input	Electrical voltage - General (general voltage entering the system).	1
Input	Water level - Level of the water tank of the steam generator.	1
Input	End of career (the maximum and minimum stop of the motor speed control)	2
Input	Optical encoder (the speed and position of the stepper motor, used in air flow control)	1
Output	General power on (the general power button of the psychrometric bench).	1
Output	Fan power on (the fan power switch that sucks the air at the initial point).	1
Output	Stepper Motor (how much voltage the fan in zone A will have).	1
Output	Compressor ignition switch (the on/off switch that cools the refrigerant system).	1
Output	Resistors switch 1, 2, 3 and 4 (the on/off switches of the electrical current in the resistors).	4
Output	Resistors switch A, B and C (the on/off switches of the immersion resistances inside the heater.	3

2) Electronic module

A solution based on Arduino and Raspberry Pi has been implemented in the electronic module, which allow communication with the AWS IoT Core and the psychrometric equipment, as shown in Fig. 3.

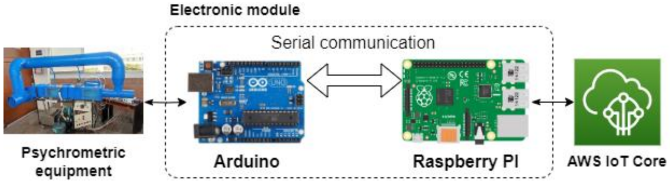


Fig. 3 Electronic module.

Between these two embedded systems, a serial communication (through a USB port) is used for the transmission and reception of data. The reason why these two embedded systems are used is for their individual strengths, while the Raspberry PI allows to maintain a stable TLS/SSL encrypted communication with the MQTT Broker, the Arduino if a low-cost module which allows to interact with a diverse number of libraries of measurement, which are used to obtain the input and output values. The integration between the Arduino and the measurement modules is given through a printed circuit board (PCB) as shown in Fig. 4; and the detail of each module and variable is shown in Table III.

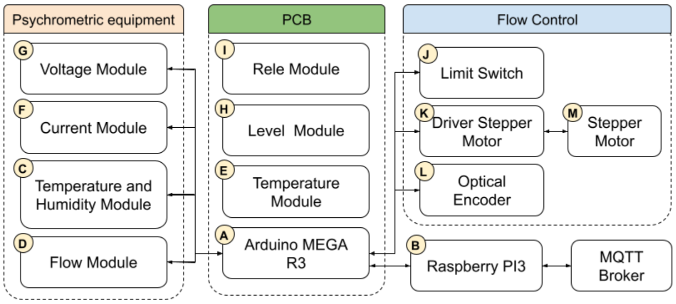


Fig. 4 Integration between the Arduino and the measurement modules.

TABLE III

DETAIL OF THE MEASUREMENT MODULES USED

Tag	Module	Variables	Comment
C	Temperature and Humidity module DHT22	Ambient humidity	Single Bus
D	Flow module BMP280	Air Flow	I2C
E	Temperature module MAX6675	Coolant temperature	SPI
F	Current module ACS712T	Electric current	Analog signal
G	Voltage module ZMPT101B	Electric tension	Analog signal
H	Level module	Water level	Digital signal
I	Relay module	General ignition, fan on, compressor ignition, resistor switch 1, 2, 3, 4 resistor switch A, B, C	Digital signal
J	Flow control module	2 end-of-stroke sensors	-
K		Stepper motor controller	-
L		Optical encoder	-
M		Stepper motor	-

The Level Module allows to turn off the immersion heaters inside the water tank of the steam generator, to prevent it from being damaged (since it breaks down if the steam generator runs out of water and the heaters continue to work).

Within the Flow Control (Fig.5) there are end-of-travel sensors, which establish the initial and final position of the stepper motor, likewise the speed of rotation and position is given by the optical encoder, which It is attached to the motor shaft.

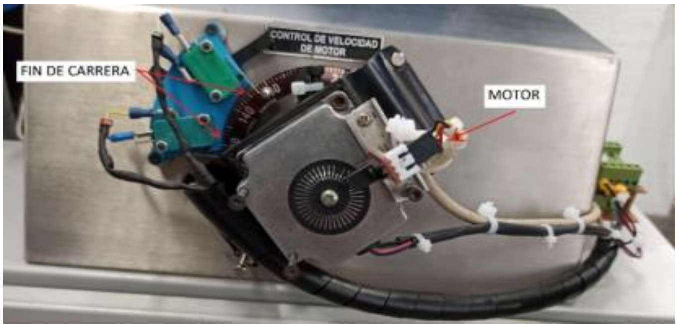


Fig. 5 Flow Control: Stepper Motor, Limit Switch and Optical Encoder.

3) Camera

From the values measured from the psychrometric equipment through the electronic module, there are 7 measurements that do not have sensors and can't be integrated into the Arduino, for example, the refrigeration system pressure (because it is a hermetic and old system). For this reason, cameras were used to see the current values of these 7 parameters, which are shown in the Table IV. It should be remembered that "input type" is the input to the electronic module, that is, the output of the psychometry equipment.

The cameras used in the present work, were the rear-facing of mid-range cell phones such as the Samsung Galaxy J5 Prime, with a resolution of 13.0 MP. To integrate it with the system, links were created in google meet with an educational account @pucp.edu.pe, which allowed unlimited video calls duration. These links were opened on cell phones and registered on the platform, as shown in Fig. 6.

TABLE IV
VARIABLES DEFINED IN THE SYSTEM

Type	Variable	Num. Variables
Input	Coolant Pressure - State 1, 2, 3 and 4 (the pressures of the cooling system).	4
Input	Refrigerant Flow - Rotameter (the mass flow rate of the refrigerant)	1
Input	Condensed temperature (the condensed liquid temperature that comes out of the process).	1
Input	Condensed volume (the condensed liquid volume that comes out of the process).	1

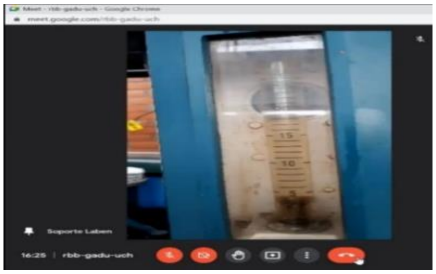


Fig. 6 Video of the variable "Refrigerant flow - Rotameter".

B. Cloud AWS

The Cloud - AWS block consist of the Amazon Web Services (AWS) to be used and the software involved in the proposed IoT system. AWS was chosen as cloud provider, since it is considered a leader in Gartner's Magic Quadrant for Cloud Infrastructure and Platform Services 2022 [17], and has been so for 12 consecutive years.

1) MQTT Broker - AWS IoT Core:

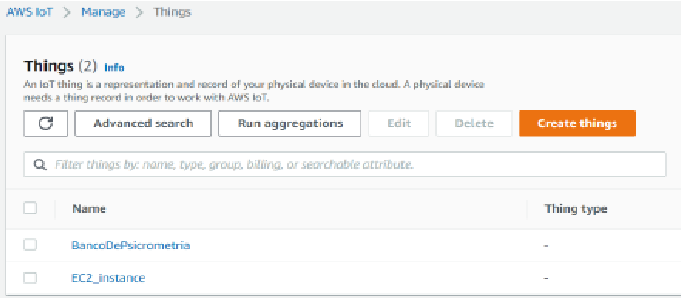
As part of the architecture, an MQTT broker was used, which allows the publisher/subscriber network protocol for exchanging messages between devices and is compatible with the raspberry pi through a connection via ethernet or Wi-Fi [18].

For using the MQTT broker, it is possible to use many options, here we are taken 3 in considerations:

- A free public one in the cloud, a good alternative in educational or test environments; but not for production.

- A second option would be to install an MQTT broker in a cloud instance, which can be done with Mosquitto or RabbitMQ, and it is even possible to configure the environment to use MQTTs (MQTT with TLS) so that the data sent is encrypted. From the security point of view, this solved the problem of communication in an insecure medium; however, according to the CIS Critical Security Controls (previously known as SANS Top 20 controls and recommendations for cyber defense), the control and inventory of assets is the first item against attacks and vulnerabilities.

For this reason, because this solution might scale in the future to hold more equipment and therefore more IoT devices, a system to control all devices would also have been needed (so, it would have to be develop). Consequently, this second alternative was discarded, and a third option was chosen, which is to employ a SaaS Solution from a cloud provider. In this case, being in a AWS environment, the AWS IoT Core service was selected. It includes a device management section "things" (as shown in Fig. 7) and another section of authentication and authorization. Authentication enables the use of X.509 certificates on IoT devices for the use of TLS, and authorization defines the actions that an authenticated device can perform.



An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

Filter things by: name, type, group, billing, or searchable attribute.

☐	Name	Thing type
☐	BancoDePsicrometria	-
☐	EC2_instance	-

Fig. 7 "Things" created in the AWS IoT Console.

Another important factor is the standardization of MQTT topics, since there are many variables in the psychometrics equipment (39 variables), a document was created with the assignment of each measured variable to an MQTT topic.

Some examples are shown in Table V, as such the assignment of the input variable "Refrigerant temperature - State 1" (defined in Table II) to an MQTT topic. With this standardization, both the web server developers and the electronic module developers, were aware of the topics they should use.

TABLE V
EXAMPLES OF MQTT TOPIC STANDARDIZATION

Type	Variable	MQTT Topic
Input	Coolant Pressure - State 1	LABEN2021/ TR_1
Input	Coolant Pressure - State 2	LABEN2021/ TR_2
Input	Coolant Pressure - State 3	LABEN2021/ TR_3
Input	Coolant Pressure - State 4	LABEN2021/ TR_4
Input	Ambient temperature - Zone A	LABEN2021/ TBS_A
Input	Ambient temperature - Zone B	LABEN2021/ TBS_B
Input	Ambient temperature - Zone C	LABEN2021/ TBS_C
Input	Ambient temperature - Zone D	LABEN2021/ TBS_D
Input	Ambient temperature - Zone E	LABEN2021/ TBS_E
Input	Electrical voltage - General	LABEN2021/ V_1

2) Web Server – EC2:

When choosing the web server, it was necessary to support WebSocket with the largest number of web browsers compatibility. The reason for using WebSocket is because it allows push data to be sent to connected users in real time, keeping the communication channel open. Analyzing the different options available in the market and taking as a reference [5], it was decided to use NodeJS v16.17.0 with the Express v1.17.2 framework as backend server and the socket.io v4.2.0 [19] library for bidirectional communication between the server and the client.

As mentioned in the socket.io documentation, if WebSocket are not natively supported in the web browser, the library automatically fallback to ajax long-polling, thus allowing more flexibility. To monitor and deploy the web server, the free version of the pm2 [20] library was used.

In the web server, a module was created for user management that allowed the following functionalities to be achieved in real time:

- See whether a student is connected or not to the system.
- Assignment of psychometric variables to a student and update of the student's UI with only those variables.
- View changes made by one student on everyone else's screen. For example, turning on a switch.
- In case of problems with the psychometric equipment, it was possible to unassign all variables to all students and turn the equipment off immediately by pressing the emergency button.

The aws-iot-device-sdk-v2 (v1.8.8) library was used for the web server to be a client of the MQTT broker. Thus, and in combination with Socket.IO, it is possible to have the following functionalities in real time:

- Visualization of the current value of the variables sent by the psychometric equipment.
- Remote control of equipment by students and teachers.
- Turn off the equipment by pressing the emergency button.
- Real-time graph of the values received by the equipment.

The complete flow from the electronic part to the end user and vice versa is shown in the Fig. 8 and Fig.9 respectively.

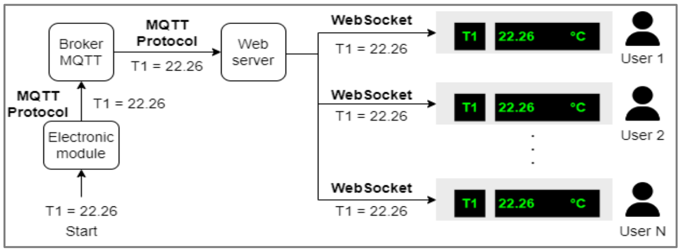


Fig. 8 From the electronic module to the user.

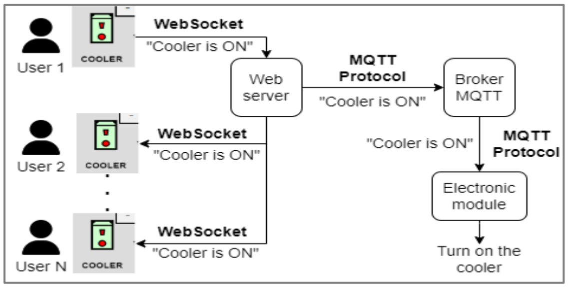


Fig. 9 From the user to the electronic module.

In the frontend, the user accesses the platform through the URL https://www.labenergiaremoto.com, with his credentials, but mush has been created before by the system administrator.

Upon entering, the main interface (Fig. 10) will be displayed, which will vary according to the role of the user (teacher or student). Together with WebSocket, the p5.js library was used, which allows the creation of sketches that render the psychrometric equipment which the student is interacting with.

The options seen on Fig.10 (A to K) vary depending on the user role, as shown in Table VI.

3) MySQL database - RDS:

During the use of the application, there will be at most 25 writes per second (of all the input sensors) since there is only 1 psychometric equipment. The performance advantage that NoSQL achieves against SQL RDBMS in writing data [21] does not generate a great impact; and even though it is more common to have IoT solutions with NoSQL databases [22] and [23]; a MySQL database has been chosen for the persistence layer [24], because it meets the needs of the project.

In future versions of this solution, where more equipment and sensors might be integrated, it is proposed to have a NoSQL database for the storage of IoT data and an RDBMS for user management, since it has ACID properties that allow the correct assignment of variables without errors. The database schema needed to support the platform is shown in Fig. 12.

Finally, it was decided to select the database as a separate service, using AWS RDS, which is responsible for managing backups, operating system patches, automatic failure detection, and recovery. In this way, the project team focuses on the solution and does not invest hours in the management of a database as if it was deployed in a EC2 instance.

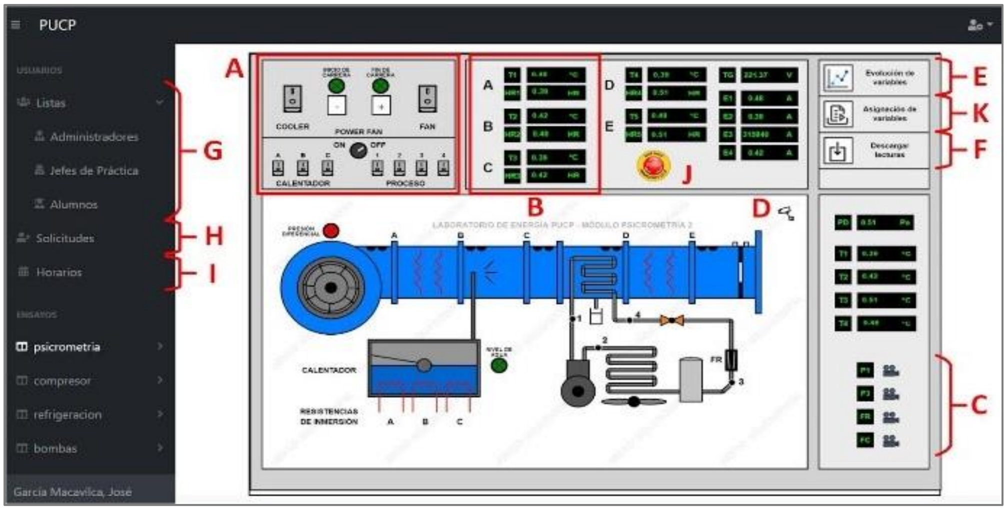
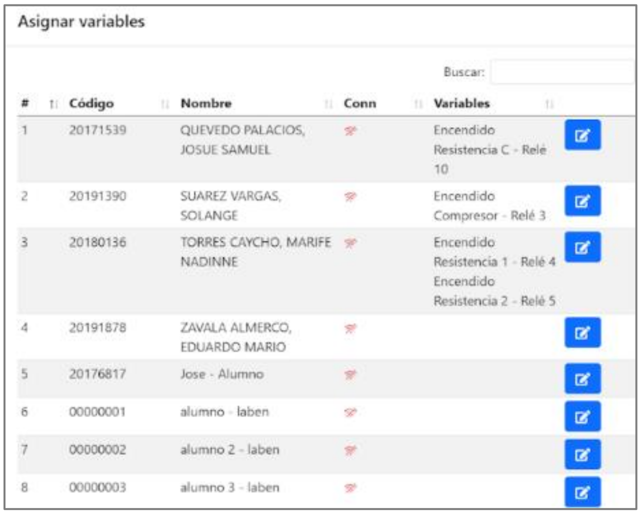


Fig. 10 Main interface.

TABLE VI
OPTIONS ACCORDING TO THE USER ROLE

Role	Opt.	Functionalities
Student/ Teacher	A	Control of the output variables assigned to be controlled.
Student/ Teacher	B	Display of the current value of the input variables.
Student/ Teacher	C	Access to the camera of the variables that do not have sensors.
Student/ Teacher	D	Access to the panning camera that shows the environment where the psychrometric equipment is.
Student/ Teacher	E	Visualization of the evolution of the variables in real time.
Student/ Teacher	F	Download of the data captured in the laboratory carried out.
Teacher	G	List administrators, teachers, and students.
Teacher	H	Observe account creation requests, where the request may be approved or denied.
Teacher	I	List the student groups.
Teacher	J	Emergency button to be pressed to turn off the psychrometric equipment in case of emergency.
Teacher	K	Assign the variables to be controlled by the students, as shown in Fig. 11.


Asignar variables

Buscar:

#	Código	Nombre	Conn	Variables
1	20171539	QUEVEDO PALACIOS, JOSUE SAMUEL		Encendido Resistencia C - Relé 10
2	20191390	SUAREZ VARGAS, SOLANGE		Encendido Compresor - Relé 3
3	20180136	TORRES CAYCHO, MARIFE NADINNE		Encendido Resistencia 1 - Relé 4 Encendido Resistencia 2 - Relé 5
4	20191878	ZAVALA ALMERCO, EDUARDO MARIO		
5	20176817	Jose - Alumno		
6	00000001	alumno - laben		
7	00000002	alumno 2 - laben		
8	00000003	alumno 3 - laben		

Fig. 11 System variables assignation

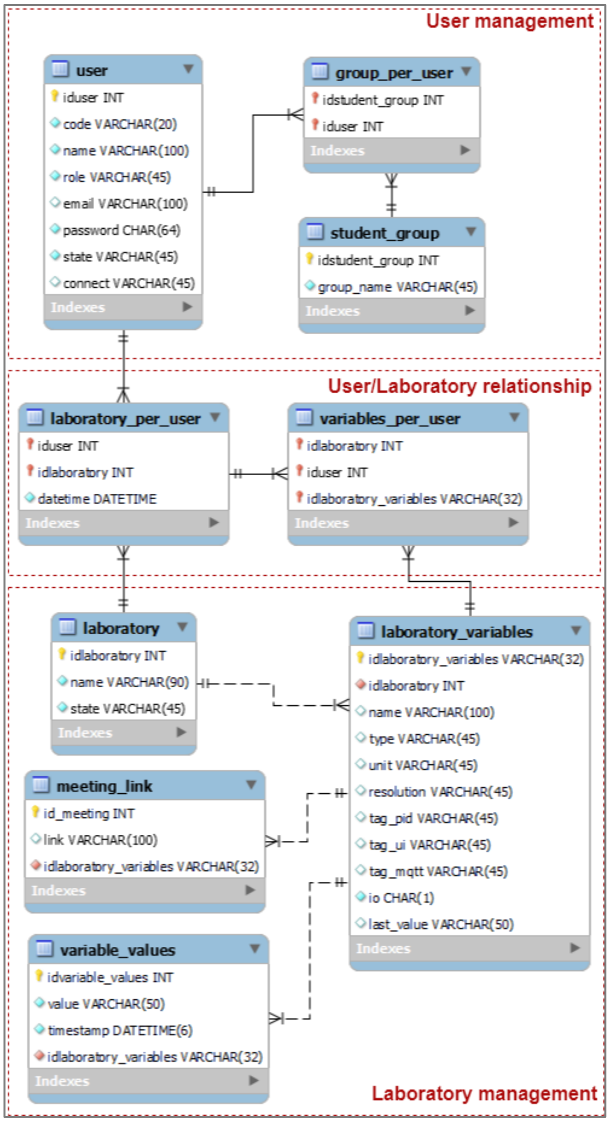


Fig. 12 Database Schema.

4) Nginx reverse proxy:

NGINX is an open source, high-performance HTTP server and reverse proxy. The purpose of using Nginx in the architecture is to use it as a load balancer and to set up the SSL/TLS certificates in a component other than NodeJS, thus removing its complexity.

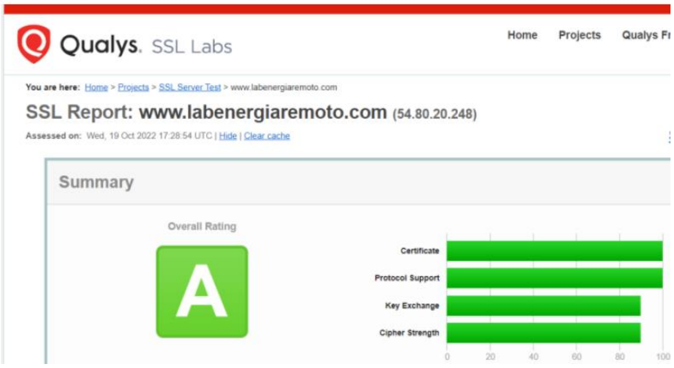
To save on costs, a certificate from Let's Encrypt was used in conjunction with a Certbot, which allows the 3-month certificate to be renewed automatically and for free. Nginx will be in charge of receiving the incoming HTTPs requests and forward them to the NodeJS. Once the NGINX server is configured with its SSL/TLS certificate, its correct operation can be verified through a SSL Server Test, as shown in Fig. 13.

5) Amazon Route 53:

It is a highly available and scalable DNS web service. Two functionalities of the service were used:

- Domain name registration: for the registration of the domain “labenergiaremoto.com”.

- DNS routing: it is defined how to send internet traffic to the implemented web application. To achieve this, Route 53 has public hosted zones. A public hosted zone is a container of rules that define how traffic coming from the Internet is sent to a specific domain (or its subdomains). Thus, a public hosted zone is created, and traffic addressed to www.labenergiaremoto.com is configured to be sent to the public IP of the EC2 instance, as shown in Fig. 14.

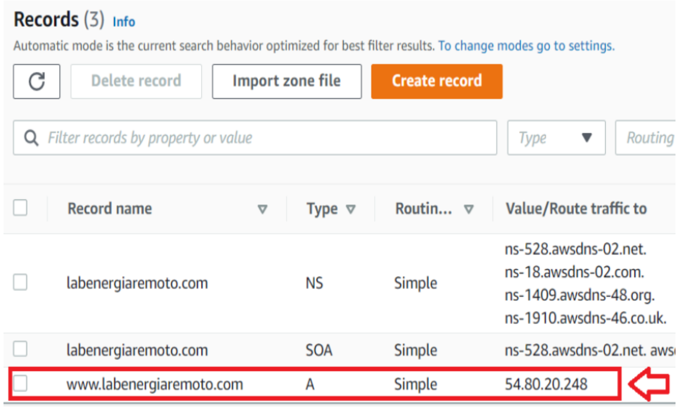


You are here: Home > Projects > SSL Server Test > www.labenergiaremoto.com

SSL Report: www.labenergiaremoto.com (54.80.20.248)

Assessed on: Wed, 19 Oct 2022 17:28:54 UTC | Hide | Clear cache

Fig. 13 SSL/TLS Report.



Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

☐	Record name	Type	Routin...	Value/Route traffic to
☐	labenergiaremoto.com	NS	Simple	ns-528.awsdns-02.net. ns-18.awsdns-02.com. ns-1409.awsdns-48.org. ns-1910.awsdns-46.co.uk.
☐	labenergiaremoto.com	SOA	Simple	ns-528.awsdns-02.net. aws
☐	www.labenergiaremoto.com	A	Simple	54.80.20.248

Fig. 14 DNS Routing in Route 53.

IV. DEPLOYMENT, TESTING AND ASSOCIATED COSTS

For the deployment on the cloud, two of the AWS services used and their configurations are shown in table VII.

The AWS IoT Core and Route 53 services work under the SaaS model, so they do not provide configuration of their underlying infrastructure.

TABLE VII
AWS SERVICES CONFIGURATIONS

Service	Instance Type	Details
EC2	t2.micro	vCPU: 1 RAM: 1 GiB Storage: 16 GB Operating System: Ubuntu 20.04.5 LTS
RDS	db.t3.micro	vCPU: 2 RAM: 1 GiB Storage: 20 GB

1) Pre-production tests:

For the pre-production tests, the Apache JMeter software was used, in which a test plan with "Thread Group" was created, with the "Number of Threads" configured as 10 (to simulate 10 users); Ramp-up as 0 (to simulate all concurrent users) and Loop Count as 5 (to simulate 5 iterations).

Using “pm2 monit”, CPU and memory consumption were monitored as shown in Fig. 15. A system functionality that consumes resources by extracting a lot of information from the tables is "download csv file" which export the data generated in the platform.

The CPU and memory consumption were monitored as shown in Fig. 16.

Fig. 15 JMeter CPU and RAM consumption on EC2 using pm2.

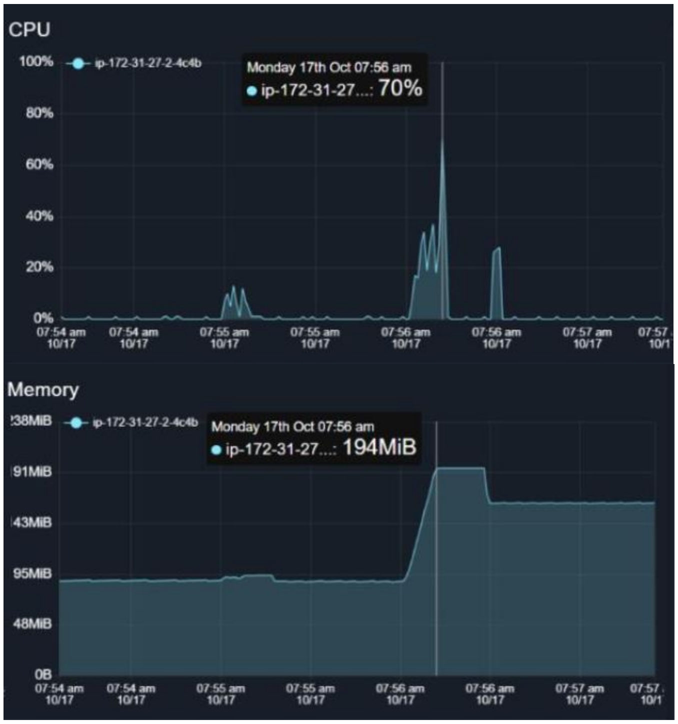


Fig. 16 "Download CSV functionality" CPU and RAM consumption on EC2.

In both tests, CPU consumption never exceeded 70% and memory consumption reached 194MB at most (out of 1GB available).

To measure MySQL RDS consumption, the MySQL Workbench dashboard was used as shown in Fig. 17.

Efficiency of 99% is shown in "Table Open Cable" which ensures that the data requested several times from the same table will be provided as quickly as possible since it remains "open". Likewise, it is observed that the InnoDB buffer does not exceed 80%.

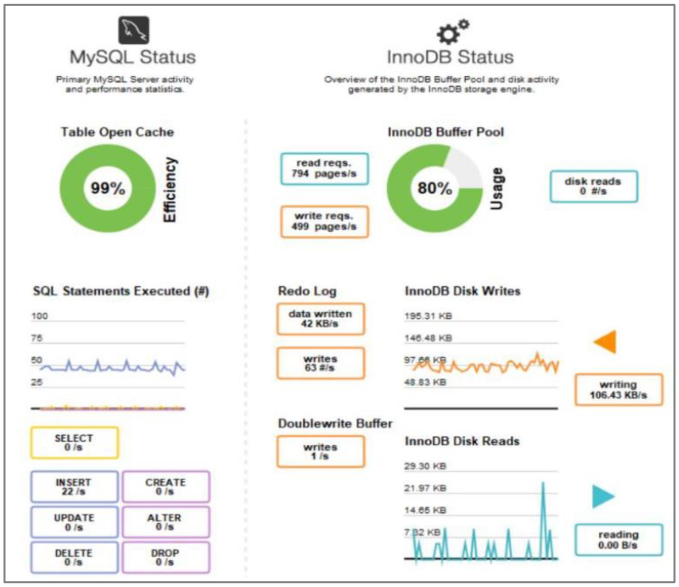


Fig. 17 MySQL RDS resources consumption.

2) Production tests:

During the 2022-1 semester, real laboratories were carried out with students from various courses using the zoom platform, as shown in Fig. 18 and Fig. 19.

Likewise, as shown in Fig. 20, in cases where students were allowed to attend the laboratory, they still used the platform to track the history of all the measurements done during the laboratory session.

Regarding the storage consumed after a laboratory session, using the AWS console for the RDS service, 18,381 GB were available before the laboratory. After the laboratory session, 18,345 GB were left, giving a total of 35.98 MB in 3.3 hours that lasted the laboratory experience.

If extrapolated to 4 hours, it would have an approximate of 43.62 MB for a single laboratory session in 1 course. According to the head of the laboratory's own information, in a regular semester (4 months) there are a total of 51 laboratory sessions, which would result in approximately 2.22 GB per cycle, 10% of the total provisioned capacity.

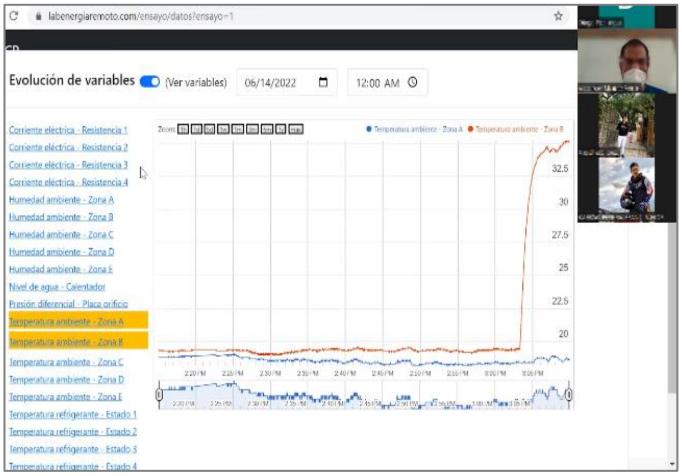


Fig. 18 Dashboard with 2 variables being rendered.

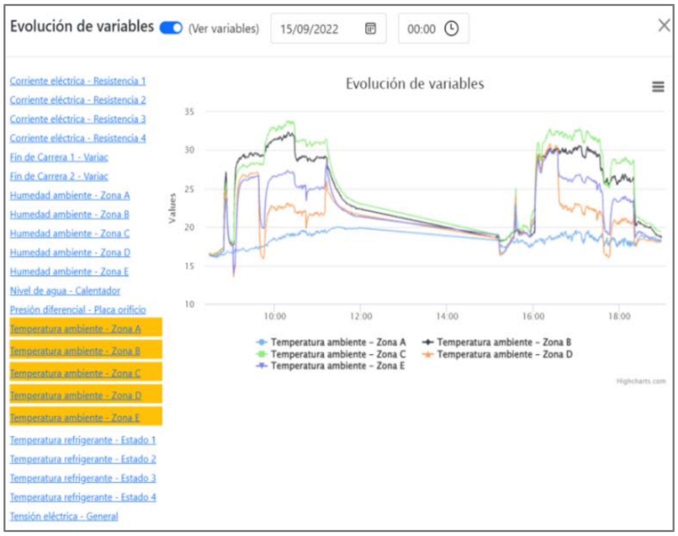


Fig. 19 Dashboard of 5 variables record being rendered.

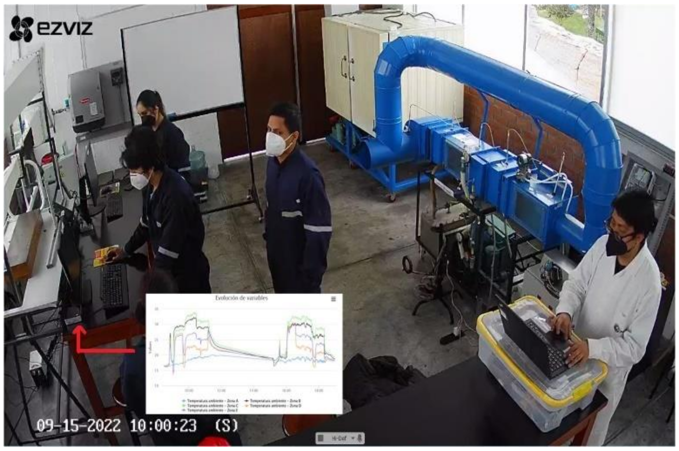


Fig. 20 Student using the platform during a laboratory session.

3) Associated costs:

There are two types of costs associated with this research project. All the costs are in US Dollars. Human resources are not being considered since the project was financed by the same university.

- CAPEX (Capital Expenditures), for the local environment (electronic module), shown in table VIII.
- OPEX (Operational Expenses), for all the cloud services, shown in table IX.

EC2, RDS, and Route 53 costs are fixed based on the instance type and disk size; however, the cost of IoT Core – Messaging are variable, but for this project it has been calculated after 8 months of using the platform. An approximate of 8,081,205 messages are generated by both devices (raspberry and EC2) per month which provides the approximate cost shown in the table IX.

TABLE VIII
CAPITAL EXPENDITURES (CAPEX) – IN US DOLLARS

Device	Quantity	Unit Cost	Total Cost
Arduino Mega	1	33.75	33.75
MAX6675	4	6.25	25
DHT22	5	8.5	42.5
BMP280 pressure sensor	1	3.75	3.75
ACS712T-20A Current Sensor	4	3.75	3.75
AC Voltage Transformer - ZMPT101B	1	5	15
Water Level Sensor Switch	1	8	8
End of career sensor	2	1.5	3
ENC28J60 module	1	7.5	7.5
Relay Module x8	1	10	10
PaP Stepper Motor Nema 17 1.7A 5kg.cm	1	18.75	18.75
Driver PaP TB6600 4.0A	1	12.5	12.5
Infrared Encoder Sensor	1	1.5	1.5
Card Manufacturing	1	62.5	62.5
Additional Components (Terminals, Swords, Resistors)	1	13.75	13.75
		Total	$262.5

TABLE IX
OPERATIONAL EXPENSES (OPEX) – IN US DOLLARS

AWS Service	Monthly cost
EC2	10.07
IoT Core – Messaging	8.08
IoT Core – Connection Time	0
RDS	14.71
Route 53 – Hosted Zone	0.5
Total	$33.36

V. RESULTS AND DISCUSSION

The platform was deployed at the beginning of 2022 and has been running all year, which is reflected in a total of 2 academic semesters and 102 laboratory sessions using the platform in total.

Each semester an approximately of 2.2GB was consumed, meaning that, with provisioned storage, up to 4 years of measurement history can be saved.

From the point of view of cloud architecture and the services used, the platform has shown no problems; however, at the software implementation level, a drawback was detected which was highlighted by the students at the end of the first laboratory experience (March 2022). As shown in Fig. 18 and Fig.19, the system allows to display throughout a laboratory session: the data obtained in previous sessions and the data which is obtained in real time. If it is considered that a sensor sends its status every second, in one hour there are 3600 values from 1 single sensor. In 4 hours, there are 14,400 values for a single sensor (keeping in mind that there are 25 “input” sensors). In the first tests with students, the chart.js library was used to display the data; however, when trying to display the 4 hours laboratory data, the rendering of the data points on the web page took between 5 to 8 seconds for 1 variable. This time was increased by 2 or 3 seconds for each additional variable that was selected, which was not acceptable from the usability point of view.

This number of points to be displayed were going to increase as new laboratories began, for this reason, it was decided to switch to the Highcharts.js library, which took only 1 second to display the 14,400 data points of each variable, increasing to 3 seconds at most, to display up to 7 variables.

Likewise, it is important to calculate and to be aware of the availability time that the general solution will have, considering that there is no redundancy in any of the services used local or in the cloud. For this reason, the calculation would be:

- Avail_{EC2} x Avai_{RDS} x $\mathrm{Avai}_{IoT\text{-}Core}$ x $\mathrm{Avai}_{Route53}$
- 99.99% x 99.95% x 99.9% x 100%
- Cloud Availability = 99.84%

VI. CONCLUSION

In this article, an IoT system based on the AWS cloud was designed and deployed, which allowed the monitoring and control, through a web interface, of a psychometric equipment in an energy laboratory of a Peruvian university. The deployed solution provides the ability to manage users, roles, and access based on dates and times.

The proposed solution architecture uses 4 cloud services to operate the IoT and Web backend, which provides a combined availability of 99.84%, which can be improved if redundancy is added by performing horizontal scaling.

The use of the MQTT and WebSocket protocols made it possible to provide an end-to-end real-time solution, with minimal latency and supported by TLS, securing the data that travels from the raspberry pi to the end user.

The use of cameras to visualize sensors that cannot be connected to the platform gave a good result, considering the limitations of the mobile equipment used.

Finally, the deployed solution has been used throughout the year 2022 without presenting any inconvenience, which shows that it can be used for the development of laboratories both in a face-to-face context and in a non-face-to-face context.

ACKNOWLEDGMENT

This work has been carried out within the Internet of Things research group (IoT-PUCP) in collaboration with the Energy Laboratory (LABEN), both belonging to the Pontifical Catholic University of Peru.

REFERENCES

[1] S. C. Sugarman, “Chapter 16 Fluid Flow, Psychrometric, and Refrigeration Terminology,” in Testing and Balancing HVAC Air and Water Systems, 2021, pp. 277–282.

[2] J. Qin, “Research and Performance Analysis on Instant Messaging Based on WebSocket,” Mobile Communication, vol. 41, no. 12, pp. 44–48, 2017.

[3] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, “A Survey on Application Layer Protocols for the Internet of Things,” Transaction on IoT and Cloud computing, vol. 3, no. 1, pp. 11–17, 2015.

[4] J. Bermúdez-Ortega, E. Besada-Portas, J. A. López-Orozco, and J. M. de la Cruz, “A new Open-source and Smart-device Accessible Remote Control Laboratory,” in 2017 4th Experiment@International Conference (exp.at’17), 2017, pp. 143–144. doi: 10.1109/EXPAT.2017.7984376.

[5] G. Carro Fernandez, E. Sancristobal Ruiz, M. Castro Gil, and F. Mur Perez, “From RGB Led Laboratory to Servomotor Control with Websockets and IoT as Educational Tool,” in Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV), 2015, pp. 32–36. doi: 10.1109/REV.2015.7087259.

[6] K. M. Al-Aubidy, A. W. Al-Mutairi, and H. Z. Al-Kashashneh, “IoT Based Remote Laboratory for Solar Energy Experiments: Design and Implementation,” in 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), 2021, pp. 47–52.

[7] M. M. Kamruzzaman, M. Wang, H. Jiang, W. He, and X. Liu, “A Web-Based Remote Laboratory for the College of Optoelectronic Engineering of Online Universities,” in 2015 Optoelectronics Global Conference (OGC), 2015, pp. 1–6. doi: 10.1109/OGC.2015.7336830.

[8] D. Pirrone, C. Fornaro, and D. Assante, “Open-source Multi-Purpose Remote Laboratory for IoT Education,” in 2021 IEEE Global Engineering Education Conference (EDUCON), 2021, pp. 1462–1468.

[9] K. Tokarz et al., “Internet of Things Network Infrastructure for The Educational Purpose,” in 2020 IEEE Frontiers in Education Conference (FIE), 2020, pp. 1–9. doi: 10.1109/FIE44824.2020.9274040.

[10] S. Fu and P. C. Bhavsar, “Robotic Arm Control Based on Internet of Things,” in 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2019, pp. 1–6. doi: 10.1109/LISAT.2019.8817333.

[11] A. Fernández-Pacheco, S. Martin, and M. Castro, “Implementation of an Arduino Remote Laboratory with Raspberry Pi,” in 2019 IEEE Global Engineering Education Conference (EDUCON), 2019, pp. 1415–1418.

[12] M. Mehrtash, K. Ghalkhani, and I. Singh, “IoT-based Experiential E-Learning Platform (EELP) for Online and Blended Courses,” in 2021 International Symposium on Educational Technology (ISET), 2021, pp. 252–255. doi: 10.1109/ISET52350.2021.00060.

[13] I. Pavel, M. Brânzilă, C. Sărmășanu, and C. Donose, “LabVIEW Based Control and Monitoring of a Remote Test-Bench Experiment for Teaching Laboratories,” in 2021 International Conference on Electromechanical and Energy Systems (SIELMEN), 2021, pp. 398–402.

[14] S. Kondratyev, V. Pikalov, A. Muravyev, and A. Evseev, “Smart Educational Robotics Laboratory Ecosystem for Remote Control of Robotic Manipulators Through Telepresence Technologies,” in 2022 2nd International Conference on Technology Enhanced Learning in Higher Education (TELE), 2022, pp. 62–65.

[15] Arduino, “Portenta-h7.” https://store.arduino.cc/products/portenta-h7 (accessed Feb. 05, 2023).

[16] Revolutionpi, “Revolution PI.” https://revolutionpi.de/shop/en/ (accessed Feb. 05, 2023).

[17] R. Bala, D. Smith, K. Ji, D. Wright, and M. A. Borrega, “Magic Quadrant for Cloud Infrastructure and Platform Services,” 2022. https://www.gartner.com/doc/reprints?id=1-2AOZQAQL&ct=220728&st=sb (accessed Feb. 05, 2023).

[18] J. A. Sanchez-Viloria, L. F. Zapata-Rivera, C. Aranzazu-Suescun, A. E. Molina-Pena, and M. M. Larrondo-Petrie, “Online Laboratory Communication Using MQTT IoT Standard,” in 2021 World Engineering Education Forum/Global Engineering Deans Council (WEEF/GEDC), 2021, pp. 550–555. doi: 10.1109/WEEF/GEDC53299.2021.9657292.

[19] Y. V. Singh, H. Singh, and J. K. Chauhan, “Online Collaborative Text Editor Using Socket.IO,” in 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1251–1253.

[20] G. I. Nugraha and F. Hidayat, “Development of API Service to Store and Utilize Video Analytics Data from Edge,” in 2022 International Conference on Information Technology Systems and Innovation (ICITSI), 2022, pp. 81–85. doi: 10.1109/ICITSI56531.2022.9970928.

[21] S. Rautmare and D. M. Bhalerao, “MySQL and NoSQL Database Comparison for IoT application,” in 2016 IEEE International Conference on Advances in Computer Applications (ICACA), 2016, pp. 235–238.

[22] A. Al-Sakran, H. Qattous, and M. Hijjawi, “A Proposed Performance Evaluation of NoSQL Databases in the Field of IoT,” in 2018 8th International Conference on Computer Science and Information Technology (CSIT), 2018, pp. 32–37. doi: 10.1109/CSIT.2018.8486199.

[23] S. Amghar, S. Cherdal, and S. Mouline, “Which NoSQL Database for IoT Applications?,” in 2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT), 2018, pp. 131–137.

[24] S. Reetishwaree and V. Hurbungs, “Evaluating the Performance of SQL and NoSQL Databases in an IoT Environment,” in 2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM), 2020, pp. 229–234.