ESPOCH Congresses: The Ecuadorian Journal of S.T.E.A.M.
Volume 3, Issue no. 4, DOI 10.18502/espoch.v3i4.17169
Production and Hosting by Knowledge E

Conference Paper

Machine Learning Application and Cloud Computing-based Monitoring for Production Management in the IPC 200 Industrial Process Simulator

Aplicación de Aprendizaje automático y monitoreo basado en computación en la nube para la gestión de producción en el simulador de procesos industriales IPC 200

I Torres[1]*, B Vaca[2], G Torres[3], M Duque[2]

[1]Centro de admisión y nivelación. Escuela Superior Politécnica de Chimborazo. 060106.
[2]Facultad de Informática y Electrónica. Escuela Superior Politécnica de Chimborazo. 060106.
[3]Facultad de Administración de Empresas. Escuela Superior Politécnica de Chimborazo. 060106.

X CONGRESO INTERNACIONAL DE CIENCIA TECNOLOGÍA EMPRENDIMIENTO E INNOVACIÓN SECTEI 2023

Corresponding Author: I Torres; email: isaac.torres@espoch.edu.ec

Published: 25 September 2024

Production and Hosting by Knowledge E

Abstract

Industry 4.0 has revolutionized the way industrial processes are managed, introducing concepts such as advanced automation, the Internet of Things (IoT), and machine learning into production and process management. This article presents the successful incorporation of two Industry 4.0 enablers into the IPC 200 Industrial Process Simulator, a key tool in the training of students in the Electronics and Automation program at the Escuela Superior Politécnica de Chimborazo (ESPOCH).
To achieve this integration, the SCRUM methodology, known for its effectiveness in managing technological projects, was employed to oversee and facilitate the implementation of the enablers, which were machine learning and cloud computing.
Machine learning implementation was carried out using the Q-Learning algorithm for process optimization and data-driven decision-making, while for cloud computing, the IoT platform Ubidots was used. This allowed for greater efficiency and flexibility in managing the simulated processes in the IPC 200.
The integration of these enablers opened the doors of innovation to the individuals using the simulator, allowing them to acquire new competencies in industrial process management, and preparing them to face the challenges of Industry 4.0. Furthermore, this improvement in the simulator provided a more advanced and realistic learning platform, resulting in more robust and applicable training for modern industrial environments.

Keywords: *Industry 4.0, Industrial Process Simulator, Node-RED, Ubidots, Q-Learning, SCRUM.*

Resumen

La Industria 4.0 ha revolucionado la manera en que se gestionan los procesos industriales, introduciendo conceptos como la automatización avanzada, el Internet de las Cosas (IoT) y el aprendizaje automático en la producción y la gestión de procesos. En este artículo, presentamos la exitosa incorporación de dos habilitadores de la Industria 4.0 en el Simulador de Procesos Industriales IPC 200, una herramienta clave en la formación de estudiantes de la carrera de Electrónica y Automatización en la Escuela Superior Politécnica de Chimborazo (ESPOCH).

How to cite this article: I Torres, B Vaca, G Torres, M Duque (2024). Machine Learning Application and Cloud Computing-based Monitoring for Production Management in the IPC 200 Industrial Process Simulator. *ESPOCH Congresses: The Ecuadorian Journal of S.T.E.A.M.*, 3(4), 130–145. DOI 10.18502/espoch.v3i4.17169

Para lograr esta integración, se empleó la metodología SCRUM, conocida por su eficacia en la gestión de proyectos tecnológicos, para supervisar y facilitar la instalación de los habilitadores, los cuales fueron aprendizaje automático y computación en la nube.
La implementación de aprendizaje automático se incorporó con el algoritmo de Q-Learning para la optimización de procesos y la toma de decisiones basada en datos y para la computación en la nube se utilizó la plataforma IoT Ubidots, esto permitió una mayor eficiencia y flexibilidad en la gestión de los procesos simulados en el IPC 200.
La integración de estos habilitadores abrió las puertas de la innovación a las personas quienes usan el simulador, permitiendo adquirir nuevas competencias en la gestión de procesos industriales, preparándolos para enfrentar los desafíos de la Industria 4.0. Además, esta mejora en el simulador brindó una plataforma de aprendizaje más avanzada y realista, lo que resulta en una formación más sólida y aplicable a entornos industriales modernos.
Palabras Clave: *Industria 4.0, Simulador de Procesos Industriales, Node Red, Ubidots, Q-Learning, SCRUM.*

1. Introduction

Industry 4.0, characterized by the convergence of advanced technologies such as intelligent automation, the Internet of Things (IoT), and data analytics, has significantly transformed the way industries operate and manage their processes. In this context of rapid technological evolution, the training of future professionals becomes essential to ensure that they are prepared to face the challenges of modern industry.

For this reason, two Industry 4.0 enablers, also called I4.0, are implemented in a simulator that represents a real production plant, thereby expanding the scope of training and learning. At the same time, allowing electronic engineering and automation students of Escuela Superior Politécnica de Chimborazo (ESPOCH) to develop skills and competencies in I4.0.

It is increasingly common to talk about agriculture, education, health and industry 4.0. [1] says that: talking about Industry 4.0 is talking about the great changes that have emerged in the technological field, making it the main protagonist. Furthermore, these changes are observed in daily life, ranging from the cultural to the organizational. From another point of view, [2] mentions that industry 4.0 is a new pattern of organization and monitoring of the value line throughout the life period of the product or service and in the trajectory of manufacturing groups based on information technologies. For this digital innovation to emerge in different areas, there are specific characteristics that describe the presence of Industry 4.0. Also known as enablers, in [3], the author mentions that the enablers of Industry 4.0 are: the Internet of Things (IoT), cyber-physical systems, cloud computing, big data, artificial intelligence, collaborative robots, virtual reality, augmented reality, 3D or smart manufacturing, and data security.

Of the latter, two have been applied in this research: Machine learning algorithms, which is a sub-branch of Artificial Intelligence, and Cloud Computing. [4] establishes that machine learning is a part of artificial intelligence focused on generating new models automatically from a group of heterogeneous data. This information analysis paradigm is causing a notable impact since, by analyzing large amounts of data from industrial processes, it makes it possible to recreate their behavior and, at the same time, gives way to the projection of future development based on already known data.

On the other hand, [5] describes the process carried out by a robot that combines artificial vision with machine learning algorithms in order to classify approximately 4000 recyclable objects in a time of 60 minutes. The operation of the robot is based on collecting information through sensors, as well as images captured by cameras. These data are the input of an intelligent algorithm, which in turn is responsible for identifying each object by determining food or drink wrappers. It is even capable of identifying logos, product brands and 3D shapes. The results obtained in this industrial process have shown the reduction of possible errors generated by humans when classifying, in addition to optimizing time in the classification process of recyclable objects.

Machine Learning algorithms have also been put into practice in the Wine Quality Analysis process through artificial intelligence techniques described in [6]. Where the quality of the wine is “determined based on 10 variables (volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulfates and alcohol).” The experiment consisted of collecting 300 records to analyze them with the J48 algorithm by training a cross-type examination, from which the most influential variables at the moment in the quality of the wine were determined: alcohol, pH, sulfates, citric acid, and the alcohol-sulfate ratio. As a result, it was determined that to improve the quality of the wine, the control of these last 4 variables is sufficient.

This example was partially replicated in this research with a view to controlling the quality of the final product. Considering that the IPC 200 simulator recreates liquid production, for example, if soft drinks were being produced, levels of sugar, gas, water, flavorings, and other components could be controlled to obtain a better-quality product. Another example is seen in [7], where the training carried out on neural networks to learn from a given context is described using the Neurosystems simulation environment from the manufacturer Siemens. Here, an intelligent system is trained based on the data perceived in the real environment, and based on these, it makes decisions and sends the signal to the programmable logic controller to give an order to an actuator.

[8] describes the development of autonomous reinforcement learning in order to manage and control dynamic systems. This work has a general purpose; it is not specifically oriented to a process; rather, it provides the possibility of reusing it in a process

in which reinforcement learning can be applied based on the Q-Learning algorithm, coded in the Python language. On the other hand, [9] exposes the application of neural networks in autonomous reinforcement learning for the control of wind turbines. Here, they are based on controlling the angle of the wind rotor blades depending on the wind conditions. To do this, they train the agent to adjust the angle value; this value is transmitted to the blades through actuators controlled by a PLC.

Likewise [10] describes that reinforcement learning is a set of steps that, when applied through dynamic programming, create a feedback controller (PLC) since it learns based on what it initially develops.

Industry 4.0 enablers can work independently but are more efficient if they work together, for example, to store, analyze, and visualize data obtained from processes with artificial intelligence algorithms. It is feasible to use the cloud computing service, since [11] mentions that cloud computing is basically a paradigm that consists of using the resources offered through the Internet, such as: email, applications, and data storage, among others. All these applications can be accessed through a web client, better known as a browser. These applications are executed on a server external to the user; that is, the client is physically unaware of the equipment since they access the services through the Internet network.

Cloud computing has several features that allow the creation of or substantially improve tasks. The most common ones are Self-service a la carte means that the user is free to contract the services that seem most convenient to them. Broad network access means that as long as the client has access to the internet, they can access all cloud services, whether light or heavy. Speed and elasticity refer to the power and performance of the servers where the applications are hosted. This is achieved given that cloud computers have robust hardware compared to conventional computers. Supervised Service explains that resources such as RAM, storage, and bandwidth, among others, can be controlled based on the client's usage requirements. In addition, cloud computing has virtualization, 24/7 availability of information, high-level security, and scalability, according to [11].

It is worth mentioning that cloud computing services are limited free of charge; that is, the providers of these services provide a certain amount of resources and tasks for free. If the client's needs increase, there are economic surcharges, considering they are relatively low in relation to the services provided. It is evident that the benefits provided by using cloud computing are significant, breaking the barrier of hardware limitations. By applying these benefits to the control of industrial processes, resources are optimized, in addition to providing more possibilities when it comes to controlling the process in real time and/or obtaining statistical data.

[12] has implemented the control of a manufacturing process composed of sensors, actuators, and controllers that perform a specific task. The result obtained after implementing industrial process control from the cloud is a bidirectional connection. That is, signals can be received from the PLC and also sent from the cloud control interface. In addition to having control from anywhere in the world, it allows you to acquire, enter, modify, and delete information as necessary.

Another example of using cloud services is shown in [13], where a PLC is connected to an elastic EC2 instance, a service provided by Amazon Web Service. Exactly a temperature sensor registers this variable periodically; this data is delivered by cable to the PLC controller, which in turn is connected to the Internet, sending the data to the EC2 instance. In the working environment of the cloud instance, you can observe the data recording in real time. The visualization menu offers various ways to observe the information received.

In [14], before defining artificial intelligence, he seeks to define intelligence itself. ”It is the faculty that living beings have to learn, reason, make decisions, and form an idea of a specific reality.” Once defined, the concept of artificial intelligence is defined as follows: ”Artificial intelligence is the ability of a machine to present the same capabilities of the living being mentioned above; that is, a machine must be able to learn, interpret, make decisions, and even propose possible solutions.”

[15] establishes that the most used branch of artificial intelligence today is machine learning. This is because what manufacturers of new technologies most seek is for their prototypes to be able to increasingly imitate human beings.

Since the objective is for machines to become capable of learning, thinking and deciding, the most used branch of artificial intelligence is Machine Learning. [16] establish that ML is the task of designing models capable of performing a specific task without explicit instructions. That is, what ML seeks is to create a sequence of steps (model/algorithm) that performs a task or process. According to [17], to produce a certain product in the industrial sector, some type of machinery is necessarily involved; it can be motors, mechanical arms, hydraulics, etc. Those that are prone to failures in their operation, and to prevent this, intelligent algorithmic techniques are being incorporated to detect failures in time, minimize repair costs, and reduce downtime.

On the other hand, [18] describes the use of artificial intelligence through artificial vision for quality control in industrial processes. He also explains that obtaining an efficient artificial vision system will depend on the elements immersed in it; the more cameras there are, the better the appreciation of the objective will be. For example, to control the filling of bottles, artificial vision can be used, which will determine if a bottle is full to the corresponding limit or has a good seal. [19] talks about the incorporation of automatic but deep learning techniques oriented to industrial processes, applying neural networks, feedforward, convolutional, autoencoders, recurrent, etc. With this, the classification processes, failure prediction, obtaining work data, extraction of relevant characteristics, visualization of process data, etc. have become intelligent.

In [20], the authors mention that artificial intelligence is the ability of machines to perform a task that is normally performed by a human being, for example, classifying objects. On the other hand, they say that the fourth industrial revolution is increasingly on the rise, and the industrial sector is no exception; it provides ample space to receive the enablers of the 4.0 ecosystem, and even more so for artificial intelligence.

[21] mentions that the main basis of artificial intelligence is to provide high-quality living conditions for humans. The objective is to replace it. What has happened is that, due to the accuracy and efficiency of complex calculations, machines are more efficient in terms of labor. For example, in the industrial bodywork sector, it is more efficient to use cobots for welding, since due to the accuracy of their movement they produce fewer errors than a human being can. But there are areas in which robots, even with the best artificial intelligence algorithm, are not going to match humans, for example in the ability to feel feelings for another person, for an object, or for unexpected situations.

According to [22], cloud computing is a distributed computing system focused on the final client. It is a collection of virtualized and connected computers displayed as unified computing services. This service is provided according to the requirements contracted by the client. The architecture of cloud computing services is composed of strategically separated hardware, platforms, and applications. Cloud services are provided in three different ways: as software as a service, as a platform, and as infrastructure.

2. Materials and methods

The management and control of the tasks of this research have been done using the agile SCRUM methodology. This methodology consists of four phases: start, planning, development, and closing. In the initiation phase, the project requirements are established. There are 17 in total, which are focused on incorporating the two enablers of I4.0.

In the planning phase, the 17 requirements were assigned in different sprints, giving a total of 8. The estimated time to develop this research was 312 hours, which gave 4 months of work with 8 hours per day. The technique used to estimate work hours was T-Shit.

With a view to sectioning the work, an architecture composed of 3 layers was proposed.

In layer 1, there is the IPC 200 simulator with its three modules (manufacturing, bottling, and palletizing), each with sensors, actuators, and a controller. The monitoring of the process is done locally through the software provided by the manufacturer; no additional component was added to this layer.

In layer 2, a gateway device was added that serves as a bridge between the controllers and the IoT platform. This equipment receives the data from the PLC and, through the HTTP/TCP/IP protocol, sends it to the Ubidots IoT gateway.

Finally, layer 3 is responsible for receiving the data sent from the Gateway. This data describes the status of the liquid manufacturing process, which in turn is displayed on a digital control panel.

On the other hand, they are stored to be processed by the machine learning module. The intelligent algorithm, after processing the data, sends back the orders that pass through the control panel to the gateway. This order reaches the controllers of each of the modules and finally they are responsible for issuing the execution order to the actuators. This layer is where the two enablers incorporated in this research are clearly found. On the other hand, they are stored to be processed by the machine learning module. The intelligent algorithm, after processing the data, sends back the orders that pass through the control panel to the gateway. This order reaches the controllers of each of the modules, and finally, they are responsible for issuing the execution order to the actuators. This layer is where the two enablers incorporated in this research are clearly found.

Following the agile SCRUM methodology, the development phase continued, during which the two industry 4.0 enablers were implemented in the process simulator.

2.1. Control, monitoring, and management of the production module of the IPC 200 simulator.

Starting from the architecture described in Figure **1**, the connection is established between the microcontroller of the production module of the IPC 200 simulator and the Ubidots IoT platform. For this, the following scheme was used:

From the sensors, the analog signals (temperature, level, pressure, and flow of liquids) are sent through connectors to the analog inputs of the S7-1200 PLC. It receives them and processes them, showing the values on display screens or in the SCADA application that the simulator has. For this investigation, these same variables are passed to the Gateway device, which in this case is a computer with basic characteristics (a Core i5 8-core processor at 2.4GHz, 16GB of RAM, and 500GB of storage). The NODE RED development module is installed on this computer. It is worth mentioning that it is

DOI 10.18502/espoch.v3i4.17169

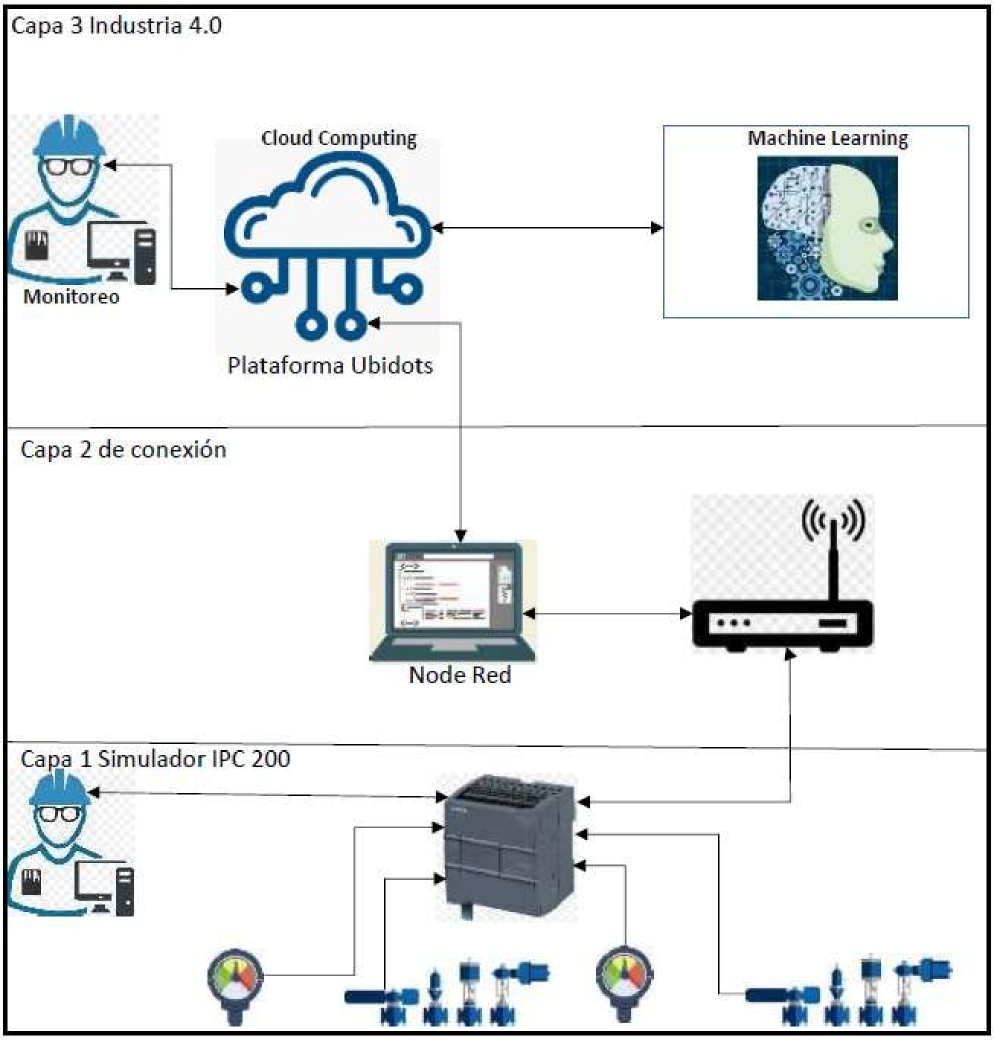

Figura 1

Application architecture. Produced by : Author.

Figura 2

Connection scheme from sensors to IoT platform. Made by: Author.

necessary to install additional libraries to configure the connection between hardware and the IoT platform, in this case, Ubidots. The libraries are node-red-contrib-s7, which allowed the extraction and insertion of data into the Siemens s7-1200 PLC, and ubidots-node-red, which also allowed the entry and exit of data from the platform.

In general, what was built into NODE RED was the extraction of the measurements recorded in the S7 1200 PLC and storing them in local variables. Then, through a function, the payload or load was converted into JSON format to send it to the IoT platform.

In the **Transport** field, the Ethernet option (ISO-on-TCP) was chosen, which is the communication protocol that was used. In **Address**, the IP address of the S7 1200 PLC was placed with the communication port, in this case, 102. The following values were left by default, and finally, the name s7_plc_1200 was established in the **Name** field. The connection was made in a general way, and the variables were created.

In the next step, variables were created with their names and with the value of the address that comes from the PLC, respecting the syntax established by the library. It is advisable to review the documentation on the official site of the **contrib-s7** library. The s7-in nodes were then created to capture the variables from the S7 1200 PLC.

In the **PLC** field, a name was selected for the created connection, the **Mode** field was left by default. In the **Variable** field, the name of the available list to capture in the node was selected and finally a name was added in the **Name** field.

Then a function type node was created in which the payload or the load stored in the variable is converted into JSON format. Finally, the ubidots-out node was added to send the payload in Json format to the Ubidots platform.

The default value was established in the **Account Type** field. In **Name**, a name was placed to identify it within the development module. In **Token**, the value assigned by the Ubidots platform was placed. In **Device label**, the name of the device created on the Ubidots platform was placed. Finally, the **Enable TLS connection** box was activated. This configuration was carried out for each of the variables extracted from the PLC. In this case, the four variables (temperature, level, pressure, and flow) are provided by the production module of the IPC 200 simulator.

On the other hand, on the Ubidots platform, a blank device was created, and the name with which it was identified in the entry nodes to the platform was added. The token was automatically created, and this value was added to the properties of the nodes shown.

Once the variables were available, a control panel was created for visualization and interpretation of the data. **Widgets** were added according to the data that was desired to be displayed. In this case, a tank type was created to observe the liquid level in the production tank. To monitor the temperature, a thermometer-type object was created, and to observe the pressure and flow, meter-type objects were created.

There are two tabs to configure the widgets. In **settings**, you can choose the name of the variable you want to display. While in **appearance**, the name with which it was displayed on the board was edited, as were measurement ranges, font, background, and other properties. These actions were carried out for each variable to be shown.

Up to this point, data monitoring was developed from the S7 1200 PLC until it was displayed on a dashboard. Now the “opposite” operation was carried out, that is, sending data from the board to the PLC to control the on and off of the different actuators (solenoid valves and motors) that the production module of the IPC 200 simulator has. In general, the data flow of this procedure began with the control buttons on the board being linked to the variables in the Ubidots environment. Then they were sent to the NODE RED module, passing through the gateway, finally arriving at the PLC, which in turn sends signals to turn an actuator on or off.

These actions are described in more detail below:

An **inject** type node was created, and in JSON format, the name and the default value that were going to be injected into the Ubidots variable were configured. After that, a ubidots-out type node was created in which the device label and the token value were entered, the same as previously used in the nodes. Additionally, the name with which it was recognized in Ubidots was placed.

This action was carried out to control each actuator, in this case, two solenoid valves and two motors. The default values were injected from NODE RED so that the variables are created in the Ubidots list.

The fields were filled with the information obtained; the new field to configure is the last one where the variable name was placed. It was exactly the same as what was called in Ubidots, from which we wanted to import the signal emitted by the button.

Since the information comes from Ubidots in Json format, it was necessary to transform it into a variable that saves the payload that comes from the control button. For this, a function type node was introduced and the lines of code that transform the payload into JSON format were entered.

The s7-out type node was configured with the same connection as previously made. In addition, new variables were created that will receive data from the Ubidots control panel.

In the same way, the variables were created according to the address syntax to be read in the PLC. In the property configurations of the same node, one of the previously created variables was chosen.

This action was carried out for each of the actuators to be controlled. Within the same dashboard that was created previously, new widgets were added to control the actuators. In this case, two switch-type buttons were placed to control two solenoid valves and two slider-type buttons with a maximum value of 1000 to control the revolutions per minute in motors. Additionally, several statistical widgets were added to better observe the behavior of temperature, level, pressure, and flow in the liquid throughout the production process. Finally, the control panel looks like this:

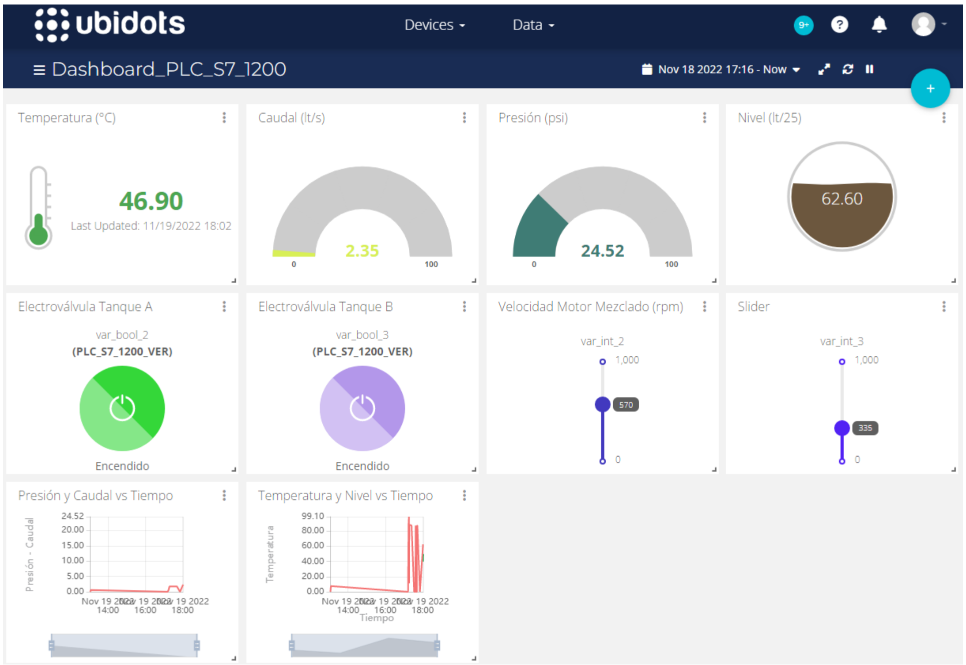

Figura 3

Ubidots dashboard with data received and sent from PLC. **Taken from:** Ubidots Plataform. **Made by:** Author.

It was observed that the data sent from the board was received in the PLC. Pressing the on and off buttons changed the state in the observation table and also the values of the sliders. Likewise, when sending data from the observation table, these are reflected on the control board, thus managing to monitor and control the operation of the production module in the IPC 200 simulator.

2.2. Implementation of an artificial intelligence module.

The main idea was that the simulator is able to activate the actuators based on what it has learned from the work environment, receiving a positive reward if it does so more efficiently and a negative reward if it does so less efficiently.

After developing the monitoring and control from a digital board (dashboard) on the Ubidots platform, the intelligent module was developed. This module was created in the Python programming language. The learning method used was model-based reinforcement learning with neural networks for greater efficiency, using the Q-Learning algorithm.

Taking into account the type of learning to be used, the following components were created:

Agent: It was the main program developed in Python. This had the ability to, through library functions oriented to the artificial intelligence that Python has, describe the environment, state, actions, and reward.

Environment: Here they described the conditions in which the production module of the IPC 200 simulator normally worked. Basically, it refers to the sensors and actuators with which it is provided; in this case, it has sensors to measure the temperature, level, pressure, and flow of liquids. On the other hand, it has actuators that are the solenoid valves for filling tanks A and B, as well as motors that move a rotor that mixes the liquids. Translated into programming language, this means the same input variables that had to be defined at the beginning of the algorithm.

Status: After the variables that explained the work environment were defined, it was necessary to know the status, that is, the values that each of the sensors were sending. Additionally, know the status of the mixing motors that can be turned on or off, all taking into account the time elapsed in producing a certain amount of product. This data reaches the agent or program in a file sheet in Excel or JSON format.

Action: Depending on the previous state, the agent learns how it should behave in the production of a certain amount of product. For example, to make fifty 50-ml bottles, you need to have 20 liters in tank A and 15 liters in tank B. Turn on the engines of tanks A and B for 10 minutes, and then turn on the engine of the main tank for 5 minutes. The liquids must be between 36 and 40 °C, the pressure in the filling hoses must be between 15 and 20 psi, and the filling flow rate must be 0.5 lt/s. The agent needs to know that to produce that quantity, he needs that data. To produce 100 bottles of the same quantity, he needs twice as much raw material, to maintain the temperature, increase the filling flow, and for the pressure in the hose to also be the same.

Reward: Since reinforcement learning was being used, it was necessary to establish the reward; these were values assigned at the beginning of the program in global variables to be used when the production process has been completed. It works like this: when the production of a certain number of bottles is completed, a reward is assigned based on the time spent. The idea is to reward a higher value for the least amount of time spent. To achieve this, the *Q-learning* algorithm was implemented.

Before describing the operation of this algorithm, it is necessary to know its parts, which are named below.

Policies: Basically, it is a table where the steps are represented together with their rewards in each iteration. In this case, Table I recorded the data on the temperature, level, pressure, and flow rate of the liquid. Taking into account that the average values of these variables in each production process have been taken, associating them with a quality rating of the final product. The average values of the variables describe the agent's environment, and the quality rating assigned corresponds to the reward. For the quality rating, a scale from 1 to 10 has been established, considering that 1 is the lowest quality and 10 is poor or high quality. The number of bottles has been taken from the values that are most repeated in the production processes.

Tabla 1

Learning policies.

No. of bottles	Temperature (°C)	Level of liquids (lt)	Flow (lt/mi)	Pressure (KPa)	Quality
22	33.25	6.75	12.4	152.3	**3**
25	28.45	7	13.6	142.5	**6**
21	30.72	6.8	17.5	155.62	**9**
18	35.60	6.1	15.1	160.30	**10**
15	35.62	5.7	15.0	160.33	**10**
17	34.68	3.8	14.85	158.60	**8**
13	32.17	3.5	13.65	157.20	**5**
Made by: Author					

Actions: Actions are the path described by the trace of variable values until reaching the reward column.

Rewards: It is the value acquired after having taken a specific action or path. In this case, they are the values of the quality column. The agent receives a reward value; the higher the value, the better the way it is executed. On the contrary, if the reward is less than 10 points, it learns that it should not take that path.

Greedy behavior: It is a behavior where only high rewards are pursued without taking into account the low ones. In this case, it did not adopt this behavior because it had to go through all the values of the variables involved in the process.

The Q-learning algorithm uses as a basis the Bellman mathematical equation described below:

$$Q(s, a) = Q(s, a) + \alpha[R + (\lambda \max q(s',) - Q(s,)]$$

Equation 1: Bellman mathematical equation

In summary, this equation describes the update that is experienced in each iteration, based on the current value and a possible reward that will be received. Within the process, there are two variables that influence the reward: learning and the discount rate. Learning regulates the speed at which the agent learns, and the discount rate takes into account the long-term reward, whether positive or negative.

3. Results and discussion

As mentioned, the main problem that this research addresses is implementing the two enablers of Industry 4.0 (machine learning and cloud computing) in the IPC 200 industrial process simulator. It is located in the ESPOCH electronics and automation laboratory with a view to optimizing the production process and converting the IPC 200 simulator into an intelligent station for training students pursuing this major.

Thanks to implementing simulator monitoring and control through a dashboard in the Ubidots IoT platform and the incorporation of the Q-Learning reinforcement learning algorithm, the problem has been solved. Converting the conventional simulator belonging to industry 3.0 into training equipment for the industrial 4.0 ecosystem.

The main contributions resulting from this research are: Monitor the work environment of the IPC 200 industrial process simulator. Control the actuators of the industrial simulator from the virtual board on the Ubidots IoT platform. Implement a reinforcement learning (RL) artificial intelligence algorithm to improve the production process.

4. Conclusion

Based on the results obtained, it is shown that in the development of this research, it was possible to connect the controller of the production module of the IPC 200 simulator via the Internet to an IoT platform, thus using cloud computing. It is considered achieved since Section 2.1 describes the process to connect the production module of the IPC 200 simulator through a gateway with the Ubidots IoT platform through the set of nodes developed in Node Red.

. On the other hand, using the sensors, the environment was recreated in which the agent would learn to act in the best way through rewards. In this environment, the machine learning algorithm was developed and applied in the IPC 200 industrial process simulator. This is described in Section 2.2, which shows how the agent learns within the production environment of the simulator.

Therefore, it can be said that through this research, it was possible to apply machine learning algorithms for management and monitoring in the cloud for the production of the ICP 200 industrial process simulator.

It is worth mentioning that the smart algorithm is limited to learning since the sensors that the IPC 200 simulator has are not enough to recreate the environment that the agent needs to learn in a more efficient way.

Applying two Industry 4.0 enablers in the IPC 200 industrial process simulator opens doors for more enablers of said ecosystem to be applied. For example, IoT sensors allow us to add more functionalities, such as quality control, to the bottling module through artificial vision recognition. Another application of the industrial 4.0 ecosystem could be predictive maintenance by adding vibration or sound sensors to the actuators (motors). Since the data read by the sensors is already received on the IoT platform, its analysis can be applied using big data, exporting it, and displaying it on report boards.

The possibility of implementing collaborative robots is also open with a view to increasing the level of production in a shorter time. In this way, we managed to convert the industrial process simulation station into a complete industrial ecosystem 4.0 environment. It would be feasible to incorporate more sensors that allow the agent's environment to be better recreated, for example, pH sensors, sugar level sensors, etc., to determine the quality of the product. This will allow the intelligent algorithm to be improved since, with a richer environment of information, it will be able to learn in a better way.

References

[1] Ana Inés Basco GBDCPG. Industia 4.0 Fabricando Futuro Buenos Aires: Unión Industrial Argentina; 2018.

[2] Román JLdV. Industria 4.0: la transformación digital de la industria. In coddiinforme; 2017; Bilbao. p. 10.

[3] Reyes PSdl. Gestión de la producción en la Industria 4.0. Sevilla; 2020.

[4] Marcos AG, Alba Elías F. Machine Learning en la industria: El caso de la siderurgia. Universidad Internacional de la Rioja. 2019, p. 9.

[5] Cuadrado AM. Utilización Del Machine Learning En La Industria 4.0. Valladolid; 2019.

[6] Galeano Arias L, Aguirre S, Castrillon Gómez O. Análisis de calidad del vino por medio de técnicas de inteligencia artificial. SciELO. 2021; p. 12.

[7] Salazar Castillo JE. Diseño E Implementación De Un Sistema De Entrenamiento En Redes Neuronales. Quito; 2009.

[8] Diaz Latorre AS. Crai Biblioteca. [Online].; 2019. Available from: https://red.uao.edu.co/handle/10614/11694

[9] Sierra García J, Santos M. Revista Iberoamericána de automática e informática industrial. [Online].; 2021. Available from: http://polipapers.upv.es/index.php/RIAI/article/view/16111/14274

[10] Diaz H, Arnesto L, Sala A. Aprendizaje por Refuerzo para sistemas lineales discretos con dinámica desconocida: Simulación y aplicación a un sistema electromecánico. Jornadas de Automática. 2017; p. 29.

[11] Hernández N, Flores Fuente AS. Computación en la nube. Mundo FESC. 2014; p. 6.

[12] Hidalgo IAC. Desarrollo De Una Plataforma Iot Para La Supervisión Y Control De Procesos Industriales De La Fabricación Inteligente En Tiempo Real A Travéz De La Nube. Quito; 2021.

[13] Gilces Niola CA. Sistema basado en el paradigma de servicios en la nube para la gestión de datos de variables industriales. Guayaquil; 2022.

[14] García Serrano A. Inteligencia Artificial, Fundamentos, Práctica, Aplicaciones España: RC Libros; 2012.

[15] Vásquez C. Tipos de Inteligencia Artificial. Avansis. 2016; p. 7.

[16] Mirjalili V, Raschka S. Python Machine Learning 2019: Marcombo; 2020.

[17] Ramírez J, Sarmiento h, López J. Diagnóstico de fallas en procesos industriales mediante inteligencia artificial. Espacios. 2018; p. 12.

[18] Nogué A, Antiga J. Aplicación práctica de la visión artificial en el control de procesos industriales. Dossier Técnico. 2012; p. 28.

[19] Gonzáles A. Researchgate.net. [Online]. 2018. Available from: https://www.researchgate.net/publication/334328255_Aplicaciones_de_tecnicas_de_inteligencia_artificial_basadas_en_aprendizaje_profundo_deep_learning_al_analisis_y_mejora_de_la_eficiencia_de_procesos_industriales

[20] Teigens DV, Skalfist P, Mikelsten D. Inteligencia artificial: la cuarta revolución industrial Stanford: Cambridge Stanford Books; 2016.

[21] Niño A. Inteligencia artificial en la reducción de procesos industriales. Tecnología Investigación y Academia. 2022; p. 192.

[22] Árias A. Computación en la Nuber 2da Edición España: IT Campus Academy; 2015.