VOL. 16, NO. 13, JULY 2021 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

MODEL-VIEW-CONTROLLER ARCHITECTURE ON A PROGRAMMABLE LOGIC CONTROLLER: EXPERIMENT WITH MICROALGAE

Claudia Lorena Garzón-Castro, Luis Beltrán, César Forero, María Loeber and Santiago Santacruz
Faculty of Engineering, CAPSAB Research Group, Universidad de La Sabana, Campus Universitario del Puente del Común, Autopista Norte de Bogotá, Chía, Cundinamarca, Colombia
E-Mail: luismb@unisabana.edu.co

ABSTRACT

In a set of experiments related to obtaining lipids from microalgae, a system was required that would automatically control some vital process variables (pH, temperature, wavelength, and light/dark cycles) in the growth of such microorganisms. This article shows the design of such system, based on the emulation of a layered software architecture in the programming of a Programmable Logic Controller (PLC). Because this type of controller is exclusively programmed using logic blocks, the challenge of programming the PLC following a structured software architecture as the MVC (Model View Controller) pattern, was achieved. This allowed us for the improvement of the maintenance processes and simplified the addition of future modules. The system was successfully tested in a set of experiments, enabling several automatic tasks as: ease of control, continuous bioprocess monitoring, real-time data gathering, and evaluation of system status and execute corresponding system adjustments.

Keywords: control software, software design, programmable logic controller -PLC, MVC pattern, microalgae cultures.

1. INTRODUCTION

Some of the photosynthetic microorganisms used in biotechnology are microalgae. Microalgae are photosynthetic organisms that, like terrestrial plants, assimilate Carbon Dioxide (CO_2), Nitrogen and Phosphorus for the formation of usable biomass in commercial products [1]. Microalgae can produce different metabolites such as saturated and polyunsaturated fatty acids, terpenes, proteins, and carbohydrates in large quantities for short periods of time, which in addition to being transformed into biofuels can generate other products with potential industrial application, such as vitamins, proteins, cosmetics and food [2, 3, 4]. So, they have become a natural resource with great potential, of high interest for both the industry [5, 2] and for the scientific community [6].

Some of the essential conditions for microalgae growth are: 1) continuous CO_2 supply to maintain optimum pH, 2) light supply (solar or artificial source), 3) temperature at the recommended value, 4) proper stirring to ensure a thorough homogenization of the medium and all mineral nutrients and finally, 5) these nutrients, required for growth, must be introduced at the beginning of the experiment in excess [7], [8], [9]. To guarantee these growing conditions, these types of bioprocesses are typically performed under predetermined conditions in a closed system called photobioreactor (PBR) [10].

These PBRs could be manually controlled, that is, manually maintain the ranges for each of the variables implicit in the process. However, on the one hand, a person would be required for this operation for 24 h and throughout the entire period that the experiment is going on. On the other hand, due to the human factor, errors of precision could arise. That is why automated closed systems are required for this type of operations, systems capable of monitoring and controlling all the variables involved in the process. Usually these automated closed systems allow: 1) to control the process variables required for the growth of microalgae, 2) to both guarantee reliability and independence of scientific staff [11] and to increase of routine production processes, achieving greater efficiency for causal-analytical physiological research as well as the verification of biological models [12], [13]. These automated closed systems are an integration between electronic hardware components and software. Some commercial PBR as PhotoBio (Applikon Biotechnology, The Netherlands) and BioStat (Sartorius A.G., San Diego, US) are available for growth of microalgae. However, this kind of commercial systems are characterized by not being flexible or extensible, which are very desirable characteristics for research-oriented applications. As a result, many researchers and engineers must develop custom software to meet their specific requirements [12], [14], [16].

A well-designed architecture is a critical factor for the success of any data acquisition software [17]. In modern software engineering, the architecture of a software program usually follows one or several design patterns [16]. Layered architecture is a model developed in software engineering to separate parts of an application based on their functionality. Due to its simplicity, MVC architecture is the most well-known model [18], [19]. Although the elements of the three layers act within each other, all of the elements are independent; thus, changes in one layer do not affect either of the other two. The first layer is called Model, because real-world objects are modelled as abstract entities in this layer, each containing attributes and behaviours. The second layer is called Control, because entities that carried out necessary operations are programmed in this layer, and data flow between the View layer and information containers is controlled. Finally, the View layer allowed the user to interact with the application in the form of a graphical [20], [21], [22], [23].

However, until now, reports on the application of MVC architectures to PLC to control the conditions of

VOL. 16, NO. 13, JULY 2021 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

microalgae cultivation have not been previously published. Therefore, the present study and results provided herein are novel. The objective of this article is to show the advantages of the developed software, which is called "*Sistema Automatizado de Control de Variables de Proceso*" (SACVAP, by its name in Spanish). For the development of SACVAP, a MVC and a layered programming architecture were used to program a PLC is presented. The application of this type of architecture was to obtain: 1) a program that was better organized according to the rules of software engineering, 2) a modular design that could increase the number of process variables that were controlled automatically and 3) implement a program to integrate the hardware devices of the system to acquire different signals. SACVAP allows controls the temperature, wavelength, and the light/dark cycles (photoperiod) and monitor the variable pH, all in on-line form, to experiment with microalgae. However, due to this type of device is normally programmed in Ladder, not in the structured logic found in more formal languages such as C, C++, Java, C#, Visual Basic, and Python, among others, this presented a significant challenge that must be taken into account.

The organization of the work is as follows. In section 2, overview of the system is presented. Section 3 describes the software design. Section 4 presents the results and discussion. Finally, the conclusions are included in Section 5.

2. OVERVIEW OF THE SYSTEM

To control and monitor a closed culture of microalgae and minimize the culture time, a system based on a PLC S7-1200 Siemens (Republic of Austria) was designed and implemented. It allows from one side, the control of temperature, wavelength, and light/dark cycles, and in the other side, the pH monitoring. A significant advantage of the proposed system was its flexibility: since other types of bioprocesses that require the management of the above-mentioned physical variables could also be evaluated.

2.1 Hardware Characteristics

Connections between the components of the system are PLC, boards, sensors and actuators, Human-Machine Interface (HMI) and production system.

PLC: The PLC is responsible for process control, which involves operations such as sensor data acquisition, control algorithm execution and driving the actuators according to the control signal generated. Furthermore, the PLC constantly communicates with the HMI through industrial Ethernet protocols and an internal web server.

Boards: The prototype contains the following boards: 1) analogue input, 2) digital input, 3) thermocouple management, 4) digital output and 5) analogue output. These boards detect signals from the sensors and determine which actions each actuator must execute.

Sensors and actuators: The sensors were J-type thermocouples and pH electrodes (Sensorex). The actuators comprise heaters, cooling fans, electromagnetic plates for agitation (Velp Scientific) and Light-Emitting Diode Surface Mount Device extensions (LED SMD).

HMI: This application accesses the data blocks in the PLC memory for storage and display in a monochrome touch screen. It enables the user to interact with the control process. The user configures the experimental conditions, such as: 1) the temperature set point, 2) duration of the light/dark cycle, 3) wave longitude for culture illumination and whether it is required and 4) pH monitoring. Additionally, the HMI can follow the process, checking that the work stands correspond with the configurations set at the beginning of the experiment.

Production system: The production system comprises four PBR. Each PBR is characterized by 1) a 3-L beaker; 2) a stainless-steel cap with ports to allow gas exchange, sample acquisition, supply CO_2 and air, introduce temperature and pH sensors, and central illumination; 3) external and central illumination, which was provided by the LED SMD extension; and 4) electromagnetic agitator.

2.2 Software Characteristics

The software was developed using Totally Integrated Automation (TIA) software Siemens. The programming was based on modular programming architecture of Ladder language but adjusted to the layered MVC pattern, which is typical of object-oriented programming (OOP). This type of design permitted the rapid addition of required subsystems. For this development, software engineering features established by [24] were considered: modifiability, interoperability, availability, security, performance, reliability and low cost were considered. Based on the functional and non-functional requirements of the system, as well as the Ladder-based modular design, the MVC model used for the design and development of SACVAP was constructed, as show in Figure-1.

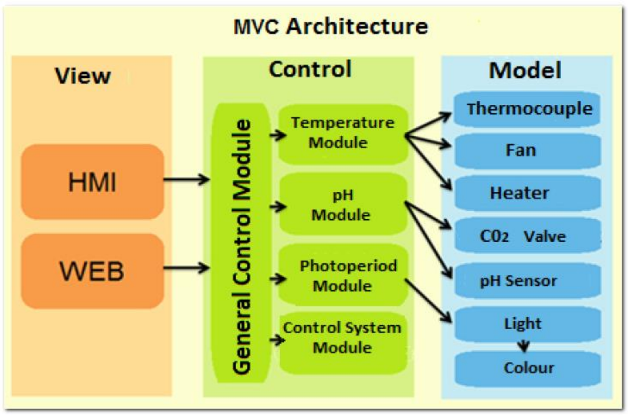

Figure-1. MVC model with modular Ladder programming.

VOL. 16, NO. 13, JULY 2021 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

3. SOFTWARE DESIGN

The design of the program was based on the engineering software concept of use cases. The objective was to control of temperature, wavelength, and light/dark cycles, and of monitoring the pH.

The model reacts, synthetically, to three operations or states:

Initial configuration: In this case, no experiments are conducted, and process variables valuated in subsequent experiments are established and initialized. The following information is introduced into the system: 1) experimental characteristics (lead researcher, code assigned to the experiment, type of microalgae, and culture volume), 2) values and set points of the process variables (temperature, photoperiod, and wavelength of light), and 3) the configuration of the experiment (active PBRs and sample intervals of the process variables).

Monitoring: Once the experiment is running, the values of the previously configured process variables are accessed for each PBR, and following information is shown using the HMI of the PLC: 1) graphs that change in real time, 2) tables with the states of the process variables, and 3) general information on each one of the records. The system automatically keeps the variables between reference values.

Parameter reconfiguration: If, for whatever reason, the initial settings needed to be reconfigured, the experiment could be halted, and the system could be returned to its initial configuration Additionally, the web page designed to remotely connect to the PLC, allows the parameters to be reconfigured and enabled both the reading and the writing of information to and from the PLC, respectively.

3.1 Software Architecture

The model MVC was designed for OOP, when it is applied to Ladder programming, some changes were required. The most significant change was related with the communication between the View and Control layers (both the web page and touch screen), since such communication is not performed using method calling (as the objects DO) but is achieved through the modification of variables that were shared between layers. Because these variables are connected asynchronously, the modification of a variable in one layer immediately affected the linked variable in the other layer. Figure-2 shows the adaptation of an MVC architecture for Ladder.

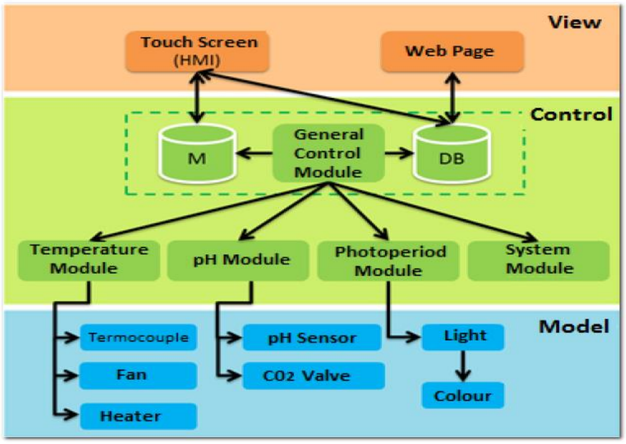

Figure-2. MVC architecture in Ladder.

In the PLC, control is achieved by continually executing the configured or programmed modules. Because programming in Ladder is performed, using independent blocks that executed specific tasks, families of blocks were created that naturally corresponded to each of the model layers. The blocks established for the design of the proposed system are shown in Figure-3, where:

Block FB (function block): The function block categorized diverse functions and stores the values of specific variables into memory.

Block FC (function): In this block, the variables are not stored into memory.

Block DB (data block): Data blocks stored the values of specific variables, like a database.

Block OB (organization block): Organization blocks are the primary execution blocks that operated the FB and are responsible for managing errors.

Variable type M: Variable that occupies a memory space in the programmable controller

Variable type DB: Variable that is contained in a DB block

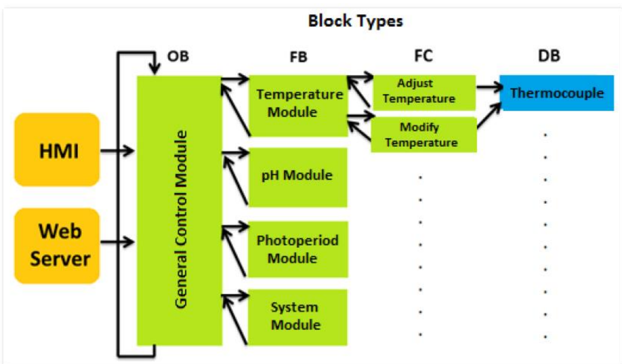

Figure-3. Relationship between blocks and modules.

After designing the architecture, the next step was to make out the OOP components into Ladder language. To this, the programming blocks provided by the PLC were considered. The list of blocks is shown in Table-1.

Table-1. Comparison between OOP and Ladder.

LADDER	POO
FB (Function Block): Function block with allocated memory.	Class
DB (Data Block): Data structure linked toa FB.	Object
OB (Organization Block): Cyclic block which runs automatically. FC (Function Call): Block of functions programming.	Method
Input variables.	Method parameters
Output variables. Not necessarily must be only one.	Method returns
Static variables.	Class variables

The other significant difference among languages that needed to be resolved was the way in which the programs were executed in Ladder. Unlike OOPs, which use method calling, the sequential and continuous execution of code rungs is performed in Ladder. When required, execution in OOP occurs via calling methods, in which only the invoked method is executed. In Ladder language, the execution method does not allow specific segments to be called individually [25], [26]. Thus, if a function block (FB) were located within the rung of another block, the block would be executed completely, calling all the implemented functions, see Figure-3. The option to avoid performing unwanted functions when executing the surrounding block includes the use of “Enable” input variables, which, justly, enable each segment to be executed. These fixes resulted in the object-oriented emulation of layered architecture in Ladder language.

3.1.1 Layer descriptions

A. Model layer

The model layer consisted of a total of five classes or DBs, representing abstractions of real objects and this contained the variables necessary for their control, including fan class, thermocouple class, pH sensor class, heater class, and LED SMD class.

The object that represents the actuators such as the fan and the heater contain the variable "Status", which determines whether the element is active or not. The object representing the thermocouple and pH sensor consisted of four variables, which allow determined the parameters required to monitor the object and store the current measurement. Among these variables are: the allowable variable range of the experiment (maximum-minimum), current value of the variable, and the settled set point to be maintained. The object representing the LED SMD contained four variables that stored the photoperiod time and the wavelength used in the experiment.

B. Control layer

The control layer was subdivided into two groups of blocks, including a general control block and a few modules control blocks.

General control module: The general control module contains the basic structure of the performance of the application, serving as the location of calls to other control modules. In addition, the general control module is responsible for receiving variables that arrive from the View layer (identified with the prefix “HMI”) and assigning them to different input variables in each of the control modules, encapsulating the data. The general control module is the only block that could access global variables and it allows contained variables values are visible to the View layer. In addition to the variables from the View layer, the block can exchange information with any peripheral element of the machine, such as the thermocouples, pH sensors, SMD LED extensions, fans and heaters, to ensure that all of the components functioned appropriately. The general control module is the only path of input and output for variables to and from the PLC.

Photoperiod module: The photoperiod module contains the processes necessary to configure and modify the photoperiod of the experiment, and these same parameter values are applied to each of the four PBRs. When the experiment is underway, the module is responsible for establishing the amount of time that the lights must be turn on and off and for activating the proper wavelength used to illuminate the culture.

Temperature module: The temperature module is responsible for configuring and processing information related to this variable in each of the PBRs. Once the configuration is saved and the experiment is in progress, the module collect data sent by the thermocouples from each PBR and processed the data through the general control system, using a Proportional Integral Derivative (PID) controller.

pH module: The pH module is responsible for monitoring the pH. The following modules were not included in the principal targets of control but do affect the targets and are required to develop the system.

DataLog module: The DataLog module is responsible for regulating the creation of record logs during the experiment. The researcher can determine how often the program should create a record of the current state of each process variable, and the module is then responsible for creating the archive that stored the record once the initial configuration was completed. The module also assigns the name of the archive using the following information: experiment code, researcher code, and date of

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

the experiment. During the experiment, records are written at the intervals indicated during the initial configuration.

Web module: The web module allows the web page to send and receive variables that set up the configuration of the parameters or receive information on the state of the project. The basic function of the web module is to allow access to the webpage using the Uniform Resource Locator (URL) of Siemens TIA V11 software.

C. Object-block relationships

The class diagram shown in Figure-4 demonstrates the relationship between the objects, or data, in the model layer and the blocks of the control layer. The record control block is dependent on the other controllers because they provided the record control block with the necessary information to create a new record in the log.

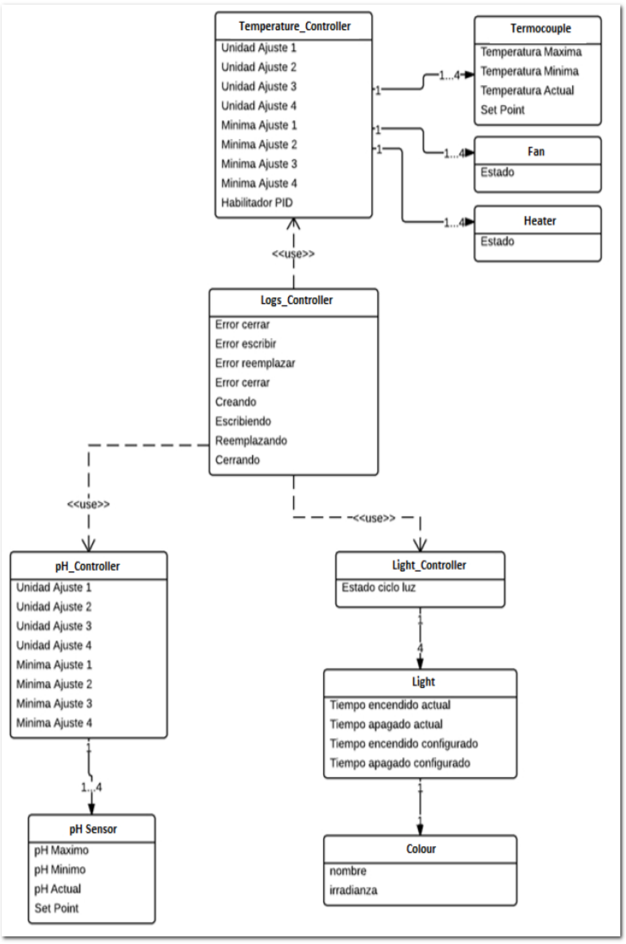

Figure-4. Application class diagram.

3.1.2 View layer

The view layer was divided into two parts, corresponding to the method of information acquisition.

A. Touch screen

Interaction through the touch screen is based on navigation between different images that were loaded onto the device. The desired interaction is achieve using distinct elements of the graphical interface, such as buttons or text fields, where the user can input information and change screens as necessary. The touch screen also allows the user to temporarily save the values of various variables created directly on the screen for use in another view. All the input or variables displayed on the screen are connect to the control layer implemented in the PLC through variables specifically designated as "HMI-PLC connection variables", which are connected to the first control layer of the application.

B. Web access

Access through the web server is only granted under two conditions: the user must be in an open valid session and an experiment must be underway. The design or template of the web interface is predetermined and constant.

4. RESULTS AND DISCUSSIONS

Extensive literature supports the benefits of MVC architectures due to its layer-based distribution and component dependence [22]. MVC architectures are so efficient that almost every system involving human-machine interactions are assembled upon it. These architectures have incredible potential in web development applications [20], [23], as well as mobile services [27]. MVC architectures also show potential for the development of video games [28], as well as applications in the medical field [21].

Based on this type of architecture, the achievements reached for the developed system from the point of view of the requirements were: a) a system that allows to experiment by varying and automatically controlling different wavelengths, temperature and photoperiod, in the same way, it allows to monitor the pH of the medium; b) optimize the processes related to this type of experiments as it introduces a high reliability factor and reduces operating costs; c) a relative ease in the assembly of the experiments; d) a constant, remote and real-time control; e) greater precision and f) access to a data history.

The achievements reached for the developed system from the point of view of development were: a) good extensibility; b) easy detection of errors and data security because it handles encapsulation; c) Quick and efficient location of execution errors in the application, in addition to modifying a block and correcting its functionality without affecting the other blocks involved in the program; d) It can be used to control and monitor another type of bioprocess that requires the handling of some, or all the physical variables used in the cultivation of microalgae; d) implementation of independent modules, with encapsulated values, that can be imported or exported to other projects, and reuse the code throughout the application, facilitating the programming of the same. However, this reuse of code must be done with great caution, as there are cases in which calling the same object in two different processes leads to the overwriting of

ARPN Journal of Engineering and Applied Sciences

information, so the logic of the program must be very clear. Avoid these drawbacks.

Of course, some weaknesses of the system can be mentioned: a) implement this type of architecture in a PLC has a high cost at the resource level, because when creating different modules throughout the application and maintain different variables to organize access to information between modules, they should be used to the maximum the resources of the machine, this can limit the size and scope of the application; b) among the resources, memory is the main deficiency. It requires a lot of memory when using this type of architecture, but, although it is a weakness, it can be solved with an external memory; c) the architecture facilitates code modification. Although the modification of input and output variables is required, an automatic update is not possible in the implementation of the modules, and the current block must be replaced manually by the updated one.

On the other hand, considering the efficiency, the prototype was tested by determining the effect of the nitrogen (N) concentration on the growth and production of lipids using the microalgae species *Neochloris oleoabundans* UTEX #1185. The cultivation conditions programmed in SACVAP were temperature of 25°C, photoperiod of 16/8 h and illumination with white light and irradiance of 1225 $\mu E \cdot m^{-2} \cdot s^{-1}$. Additionally, the culture was agitated at 125 rpm and had supplied with CO_2 in a range between 5% and 6%, and air in a range between 95 and 94%. Four PBRs were used, each with a different N concentration: PBR1 had a concentration of 0% N, PBR2 contained 25% N, PBR3 contained 50% N, and PBR4 had a N concentration of 100%.

Figure-5 shows the growth curves obtained for each of the PBRs. In PBR1, due to the lack of nitrogen, a low growth rate was observed. However, PBR4, which contained the highest concentration of N in the culture medium, showed the greatest growth rate. This result matches those of previous investigations [29], [30], which showed that higher amounts of nitrogen in the medium resulted in larger biomasses. The PBR2 as well PBR3 had presented a similar growth rate. Respect to the lag fase, it was observed that the PBR3 as well PBR4 were longer than PBR1 and PBR2.

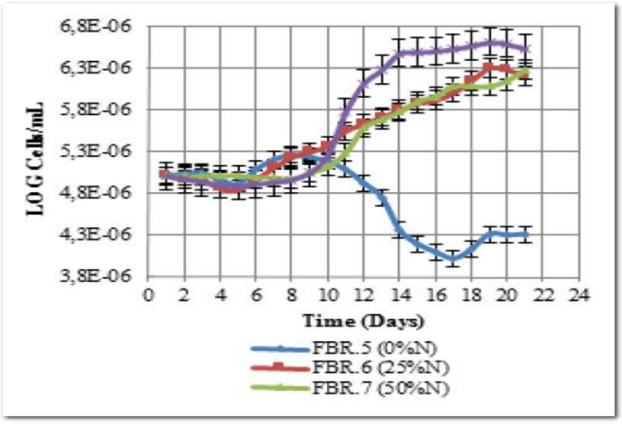

Figure-5. Growth curves for the microalgae *Neochloris oleoabundans* UTEX #1185 at different N concentrations.

The amount of lipids produced by the microalgae was analyzed quantitatively by applying Nile Red (NR) stain [31], [32]. The amount of lipids generated in each of the PBRs varied, depending on the applied treatment. As shown in Figure-6, PBR1 which was treated with 0% N, had the greatest percentage of lipids, followed by PBR2 and 3, while PBR4 produced the lowest amount of lipids. These indicated that, in the absence of N, the microalgae accumulated a greater amount of lipids in their interior. However, population growth was lower under these conditions. Conversely, microalgae cultured with 100% N showed a rapid growth in population but produced few lipids. Therefore, the lipid concentration increased when N was limited, but the growth rate or biomass production subsequently decreased. These results are in accordance with those previously reported in the literature [30].

The pictures were obtained with a Nikon ECLIPSE Ni-E epifluorescence microscope. The yellow color represents the amount of lipids in each microalga. The experimental results shown in [33] were obtained using the prototype of the present article.

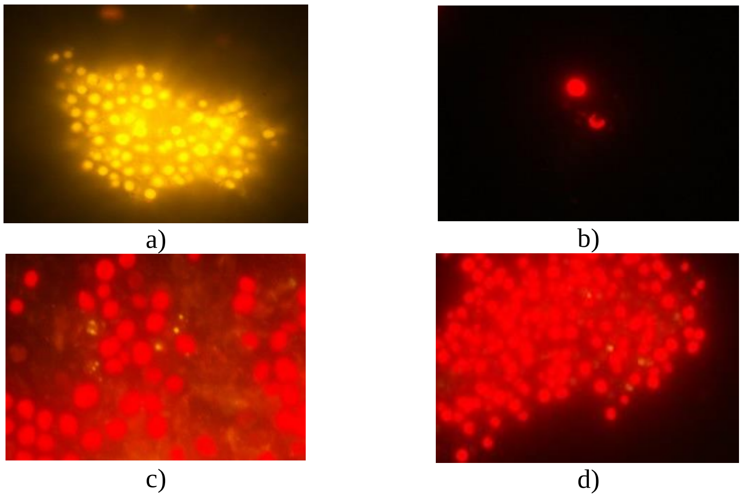

Figure-6. a) PBR1 with 0% N, b) PBR2 (25% N), c) PBR3 (50% N) and d) PBR4 (100% N).

CONCLUSIONS

It was shown that it is possible to make applications on PLC based on an architecture of Software Engineering, by developing a system that was successfully tested in several experiments on microalgae growth processes. This system called, SACVAP, has been of great assistance, executing several processes automatically, including constant bioprocess monitoring, information acquisition, real-time evaluation of the state of the system, and performing appropriate adjustments. SACVAP can give independence to scientific personnel and increase the reproducibility of routine production processes.

This prototype is currently being used to develop a method that can identify operational points to guarantee maximum biomass production by varying the culture conditions. If conditions that increase biomass and lipid production while reducing the culture time were identified, the industrial sector could invest in and provide technical support for the commercialization of bioenergy derived from microalgae.

The interdisciplinary character of this work shows the importance of itself because it is possible to realize developments that a single discipline would not be able to implement. It was possible to develop software: 1) low cost, 2) easy to use, 3) tailor made, 4) capable of interacting with hardware in charge of monitoring and control some variables and 5) guarantee the investigators the precision and reliability of its use.

The software was designed in a modular and orderly way to be able to address new requirements in a quick and effective way. In the future, it is possible increase the number of variables to control.

ACKNOWLEDGEMENTS

The authors would like to thank at the Universidad de La Sabana by the financing for the development of ING-107 “Diseño e implementación de un sistema de monitoreo y control automático de variables de proceso para la producción de microalgas con potencial de uso en biodiesel” project.

REFERENCES

[1] C. Posten and G. Schaub. 2009. Microalgae and terrestrial biomass as source for fuels-A process view. J. Biotechnol. 142: 64-69. https://doi.org/10.1016/j.jbiotec.2009.03.015

[2] O. Bernard. 2011. Hurdles and challenges for modelling and control of microalgae for CO_2 mitigation and biofuel production. *J. of Process Control*, 21(10): 1378-1389, https://doi.org/10.1016/j.jprocont.2011.07.012

[3] R. Filali, A. C. Badea, S. Tebbani, D. Dumur, S. Diop, D. Pareau and F. Lopes. 2011. Optimization of the interval approach for Chlorella vulgaris biomass estimation. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 4554-4559, Orlando, FL, USA. https://doi.org/10.1109/CDC.2011.6161110

[4] B. Wang, Y. Li, N. Wu and C. Lan. 2008. CO_2 bio-mitigation using microalgae. Appl. Microbiol. Biotechnol. 79: 707-718, https://doi.org/10.1007/s00253-008- 1518-y

[5] A. Ben-Amotz. 2008. Biofuel production by marine microalgae conversion of electric power plant wastes to biodiesel and bioethanol. J. Biotechnol. 136S, S523, https://doi.org/10.1016/j.jbiotec.2008.07.1228

[6] C. L. Garzón-Castro, J. A. Cortés-Romero, J. Arcos-Legarda and E. Tello. 2017. Optimal decision curve of light intensity to maximize the biomass concentration in a batch culture. Biochem. Eng. J. 123: 57-65, https://doi.org/10.1016/j.bej.2017.04.001

[7] K. Chojnacka and A. Noworyta. 2004. Evaluation of Spirulina sp. growth in photoautotrophic, heterotrophic and mixotrophic cultures. Enzyme and Microbial Technol. 34(5): 461-465, https://doi.org/10.1016/j.enzmictec.2003.12.002

[8] H. Takache, J. Pruvost and H. Marec. 2015. Investigation of light/dark cycles effects on the photosynthetic growth of Chlamydomonas reinhardtii in conditions representative of photobioreactor cultivation. Algal Research. 8: 192-204.

[9] K. Vasumathi, M. Premalatha and P. Subramanian. 2012. Parameters influencing the design of photobioreactor for the growth of microalgae. Renewable and Sustainable Energy Reviews. 16(7): 5443-5450, https://doi.org/10.1016/j.rser.2012.06.013

[10] K. Marxen, K. H. Vanselow, S. Lippemeier, R. Hintze, A. Ruser and U. P. Hansen. 2005. A photobioreactor system for computer-controlled cultivation of microalgae. Journal of Applied Phycology, 17(6): 535-549, https://doi.org/10.1007/s10811-005-9004-8

[11] A. Gitelson, Y. Grits, D. Etzion, Z. Ning and A. Richmond. 2000. Optical properties of Nannochloropsis sp. and their application to remote estimation of cell mass. Biotechnology and Bioengineering, 69(5): 516-525, https://doi.org/10.1002/1097-0290(20000905)69:5<516::AID-BIT6>3.0.CO;2-I

[12] L. M. Beltrán, C. L. Garzón-Castro, F. Garcés and M. Moreno. 2012. Monitoring and control system used in microalgae crop. IEEE Latin America Transactions, 10(4): 1993-1998, https://doi.org/10.1109/TLA.2012.6272485

[13] L. M. Beltrán, C. L. Garzón-Castro, D. F. Valencia and V. A. Uribe. 2013. Web control and monitoring system: Experimentation with Haematococcus pluvialis. International J. of Eng. 26(3(C): 219-228, https://doi.org/10.5829/idosi.ije.2013.26.03c.01

[14] F. González-Fernández, C. L. Garzón-Castro and E. Prieto. 2017. Software for monitoring and control of enzymatic extracts: A tool for research. ARPN Journal of Engineering and Applied Sciences. 12(21): 6067-6073.

[15] W. R. Jones, M. J. Spence, A. W. Bowman, L. Evers and D. A. Molinari. 2014. A software tool for the spatiotemporal analysis and reporting of groundwater monitoring data. Environmental Modelling and Software, 55: 242-249, https://doi.org/10.1016/j.envsoft.2014.01.020

[16] W. Wang, C. Li, E. W. Tollner and G. C. Rains. 2012. Development of software for spectral imaging data acquisition using LabVIEW. Computers and Electronics in Agriculture, 84: 68-75, https://doi.org/10.1016/j.compag.2012.02.010

[17] L. Bass, P. Clements, R. Kazman. 2003. Software Architecture in Practices. Second Ed., Boston, MA, USA.: Addison-Wesley Longman Publishing Co.

[18] E. Pantoja. 2004. El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. Acta Nova. 2(4): 493-507.

[19] R. Santaolaya-Salgado, O. G. Fragoso-Díaz and S. A., Zamudio-López. 2014. Formal Model for Restructuring of Object-Oriented Frameworks to Architecture Model-View-Adapter. Ingeniería, Investigación y Tecnología, 15(2): 187-198, https://doi.org/10.1016/S1405-7743(14)72209-7

[20] S. Chen and F. Y. Lu. 2002. Web-based simulations of power systems. IEEE Computer Applications in Power. 15(1): 35-40.

[21] H. Lin, H. Wu, W. Liang and J. Wang. 2011. Development of a real-time clinical decision support system upon the web MVC-based architecture for prostate cancer treatment. Medical Informatics & Decision Making. 11(16): 1-11.

[22] H. McHeick and Y. Qi. 2011. Dependency of components in MVC distributed architecture. Canadian Conference on Electrical and Computer Engineering pp. 691-694, https://doi.org/10.1109/CCECE.2011.6030542

[23] Y. Ping, K. Kontogiannis and T. C. Lau. 2004. Transforming legacy Web applications to the MVC architecture. Proceedings - 11th Annual International Workshop on Software Technology and Engineering Practice. pp. 133-142.

[24] F. P. Brooks. 1987. No silver bullet: Essence and accidences of software engineering. Computer. 20(4): 10-19.

[25] H. D. Páez, R. Zamora and J. Bohórquez. 2015. Programación de Controladores Lógicos (PLC) mediante Ladder y Lenguaje de Control Estructurado

(SCL) en MATLAB. Facultad De Ingeniería. 24(39): 109-119.

[26] H. Wan, X. Song and M. Gu. 2010. Parameterized specification and verification of PLC systems in Coq. Proceedings - 2010 4th International Symposium on Theoretical Aspects of Software Engineering, pp. 179-182, https://doi.org/10.1109/TASE.2010.12

[27] H. La and S. Kim. 2010. Balanced MVC Architecture for Developing Service-Based Mobile Applications. IEEE 7th International Conference on E-Business Engineering, Shanghai, China, https://doi:10.1109/ICEBE.2010.70

[28] C. Vichido, M. Estrada and A. Sánchez. 2003. A constructivist educational tool: Software architecture for web-based video games. Proceedings of the Fourth Mexican International Conference on Computer Science, Tlaxcala, México.

[29] L. Gouveia, A. Marques, T. L. Da Silva and A. Reis. 2009. Neochloris oleabundans UTEX #1185: A suitable renewable lipid source for biofuel production. Journal of Industrial Microbiology and Biotechnology, 36(6): 821-826, https://doi.org/10.1007/s10295-009-0559-2

[30] E. A. Nain, A. Ruiz Marín, L. J. Pérez-Reda, Z. Reyes García, Y. Canedo López, S. Campos García and J. H. Moreno-López. 2011. Densidad celular y acumulacion de lípidos en cultivos libre Chlorellas vulgaris y Neochloris oleoabundans a diferentes concentraciones de nitrógeno y carbonato de sodio. Unacar Tecnociencia, 5(1): 56-71, https://doi.org/10.1109/STEP.2003.35

[31] L. Balduyck, C. Veryser, K. Goiris, C. Bruneel, K. Muylaert and I. Foubert. 2015. Optimization of a Nile Red method for rapid lipid determination in autotrophic, marine microalgae is species dependent. Journal of Microbiological Methods, 118: 152-158, https://doi.org/10.1016/j.mimet.2015.09.009

[32] J. Rumin, H. Bonnefond, B. Saint-Jean, C. Rouxel, A. Sciandra, O. Bernard, G. Bougaran. 2015. The use of fluorescent Nile Red and BODIPY for lipid measurement in microalgae. Biotechnology for Biofuels, 8(42): 1-16, https://doi.org/10.1186/s13068-015-0220-4

[33] D. X. Hurtado, C. L. Garzón-Castro, J. Cortés-Romero and E. Tello. 2018. Comparison of lipid extraction methods for the microalgae Acutodesmus obliquus. Int. J. Agric. Biol. Eng. (IJABE), 11(5): 211-217, https://doi.org/10.1016/j.algal.2015.02.009