Intelligent Tutor System for Learning Object Oriented Programming

Ivelisse Teresa Machín Torres[1], Vivian Estrada Sentí[2]

[1]Ivelisse Teresa Machín Torres, Professor, Department of Computer Science, Sancti Spíritus, Cuba
[2]Vivian Estrada Sentí, Professor of the Doctoral Program, University of Computer Science, Havana, Cuba

---***---

Abstract - *The author presents in the article the proposal of an intelligent tutor system, with the use of artificial intelligence techniques that extract and analyze the information of the interactions of the student with the platform in its teaching and learning process. In order to achieve this objective, the types of intelligence preponderant in each one are considered, since it will allow to reach more individualized solutions. The results show a positive impact on problem solving, which improves the learning process.*

Key Words: Learning, multiple intelligence, artificial intelligence, intelligent tutor system, pedagogical method.

1.INTRODUCTION

The use of information technologies and its better application for them is currently a key issue in the development of any nation. The economic, social and cultural profits obtained from the technological factor will belong to those who manipulate them properly [1].

The design topics and Object Oriented Programming (OOP) are aimed at preparing students so that, knowing the general fundamentals and evolution of programming languages, students should model problems using the object oriented paradigm, make programs using some languages that support it and others hybrids that combine them with aspect orientation and Logic Programming. Thus, students should know the fundamentals, evolution and characteristics of OOP languages, their possibilities and limitations, essential features, functional and logical principles, as well as data structures, cohesion techniques, coupling, encapsulation, heredity and polymorphism, using the defined data structures.

A previous research was analysed showing a need to achieve a sufficient students' motivation towards the subject OOP [2]. The OOP should have a diachronic approach, based on procedural programming. This approach begins with the exposition of each language construction as an evolution of another previously known, overcoming some of its limitations. As a guiding thread of the exposition, recurrent concepts are used that lay on any particular paradigm mechanism [3].

Some researchers [4], agree that learning a given language is not always easy to embrace, syntax and semantics are specific to each language, as well as the programming logic of each paradigm. Kinsumba, Becerra and Lau [5] identified a set of important factors for passing Programming subject.

According to Beltrán, Sánchez and Rico [6] different weaknesses are constantly identified in the OOP application. On the other hand, Olier [7] identifies: lack of interest, a poor OOP understanding concepts, poor stated problems understanding, and poor structures identification to be used in an efficient statement solution, poor memorization or assimilation in language syntax programming.

Based on previous statements, the main objective of this article research is to elaborate a design and evaluation of the Intelligent System Content tool to strengthen the concepts of OOP supported by the theory of multiple intelligences [8]. The validation of the used tool was done by means of a pre-test and a post-test applied to two groups: controlled and experimental. The results demonstrate that the experimental group registers higher levels of association related to OOP concepts.

2. INTELLIGENT TUTOR SYSTEMS REVIEW

The intelligent tutor systems (ITS) emerge as a better alternative to technologies used until that moment for teaching through computer aids [9]. The development in this field is aimed to provide tools to teach the student as if there were an experienced human tutor [10]. To achieve this goal, artificial intelligence techniques are used in the generation of learning ways, selection of activities, support during activities, assessment, presentation of contents, among others.

Studies describing the successful application of ITS in different areas of knowledge are reported. Their application brings significant gains in learning according to other types of systems used to support the teaching-learning process [11]. Authors like Aleven [1], Eagle and Barnes [12], make emphasis that the main cause of these positive results is the feedback the student receive during the activities [13]. It is emphasized that there is a direct relationship between the amount of feedback and these gains [14]. Within the ITS are systems such as Lisp [15], ELM-ART [16], SQLT-Web [17], and HESEI [18]. These systems show that their application favors the achievement of learning gains. In contrast to this effectiveness, only SQLT-Web has been used exhaustively in real-world contexts [19] and the rest only in laboratory tests with students in conditions that can be considered as controlled [20].

In the consulted researches, the following difficulties are noticed:

1. The specific intelligences of each student are not identified and it is not given an appropriate treatment in the work sessions.
2. It is not significant to find architectures that favor reusable components among systems, even within the same area of knowledge [21]. This causes that ITS development has to be carried out from the initial point prolonging the delivery times and increasing production costs [22].
3. Inclusion of new activities in time of exploitation is not considered [23]. These are added during the development, as part of the tasks to be done at that stage, conceiving the systems in a form of black boxes where users employ the knowledge added during the development.
4. In relation to the above, the capacity of teachers is limited to extend and modify knowledge according to their considerations and to the specificities of their students. For its extension is necessary to produce new versions, which usually occurs at a slower speed than the needed by end users.
5. The encoded knowledge, with no evolution, causes that the obsolescence process of the system to be accelerated.

A previous study carried out in computer-related careers groups at José Martí University in the province of Sancti-Spíritus, Cuba [24] revealed some essential deficiencies and limitations that are presented in the OOP topics, Including the following:

1. It is not always possible to identify the classes correctly.
2. Inheritance and Polymorphism are frequently misused.
3. Sometimes are used some methods that do not clearly express what it is needed.
4. Very often there is a lack of objectivity in the management of Encapsulation.
5. In the handling of Language Semantics, repeated difficulties are presented.
6. It can be affirmed that the inability to identify the essence of a problem is often observed.

The practice shows that often the required supports do not exist and it is necessary to use techniques that, in some cases, are ineffective; due to they do not take as a base the relations student-professor-contents, which is indispensable for the correct appropriation of knowledge. This has as a result the lack of motivation and a certain rejection to the content for considering them difficult to learn.

3. METHODOLOGICAL DESIGN

A number of scientific research methods, including analysis and synthesis, interviews and surveys, were applied to prepare the proposal.

A documentary analysis related to ITS and Object Oriented Programming was carried out, as well as the techniques used. From this analysis a system was elaborated, starting from an integration of the techniques of Artificial Intelligence, and its development at the same time with the changes in teaching methods and even in the form of conceiving learning and the education proposed by the theory of multiple intelligences. A layered architectural style was applied.

When the system was finally designed, the validation was carried out by specialist's criteria, as well as validation by users' criteria, applying surveys to the effect.

4. RESULTS

4.1 Description of ITS

The intelligent tutor system was named ISCO (Intelligent System Content) and was developed as an instantiation of model to improve the assimilation of contents in programming (OOPMAC) and it is an alternative to the insufficiencies in the teaching of Object Oriented Programming. Figure 1 shows the OOPMAC model.

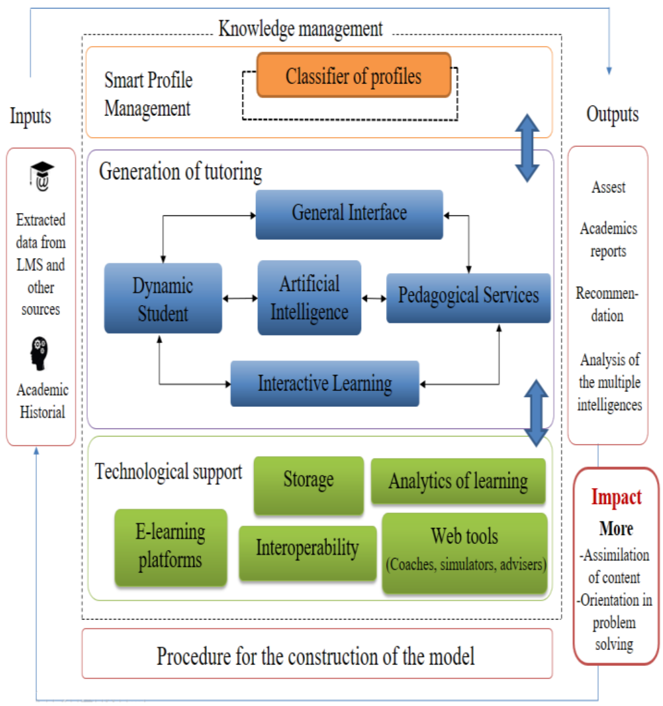

Fig -1: Model to improve the assimilation of contents in programming.

It provides functionalities to support the teaching-learning process, some of them using artificial intelligence techniques. The ISCO tool is an application developed in JAVA language with a three layer architecture (Data Layer, Business and Interface) which provides some advantages such as robustness due to encapsulation, easy of maintenance, support and flexibility, as well as high scalability.

The applied layered architectural style guarantees a logical separation between functions of a different nature. This makes it easier to protect the integrity of the components against external changes, with three well-defined layers: presentation, business and access to data. In the ISCO tool Artificial Intelligence techniques are integrated in parallel with the changes in teaching methods and even in the way of conceiving the learning and teaching proposed by the theory of multiple intelligences.

The student himself takes control of the learning process, the materials and resources, adapting them to their requirements and possibilities. The ISCO tool can be used in a wide range of domains and it contains a set of inter-related components. In the presentation page, the mask administrator Thymeleaf and the jQuery and Bootstrap libraries were used. Thymeleaf, jQuery and Bootstrap assists the division of responsibilities in the presentation through a Model-View-Controller (MVC) pattern, jQuery provides plugins to handle the status of UI components, handling events in the browser and the server, validations, definition of surfing patterns and support for locating and world spread. Bootstrap provides a number of styles and components that allow students to reduce layout time and improve design quality. In the business layer, Spring Framework [26] was used, which provides a programming and configuration model for applications in the Java EE platform, regardless of the way it is deployed. It works as a control inversion box, dependency injection is support and feature oriented programming (AOP). This framework is designed to reduce development times by allowing developers to focus on logic by delegating the necessary mechanism to integrate application components.

The application provides a REST API to handle all the information that is used from the user interface through AJAX requests. This approach allows a modern user experience and, which makes it easier to add other clients (mobile applications, web, desktop, etc.). The data access layer works as an abstraction level between the domain model and the data sources used by the application. The Hibernate ORM [27] tool was used for object-relational mapping rather than declarative configurations and the use of the API provides the persistence of data that survives during the implementation of the program. As a data source, a relational database was used with the PostgreSQL database manager system version 9.1. As web server and Servlets box, Apache Tomcat was used in version 8.5.2. Four modules were defined: Pedagogical Services, Dynamic Student, Interactive Learning and Artificial Intelligence.

The Dynamic Student module is composed of two differentiated blocks, one that models the issues that have little variation over time (IM and learning preferences) and another that reflects the dynamic characteristics (student's knowledge situation), which are updated Constantly because it depends on students´ interaction during the course or Interactive Learning Module.

In order to achieve the identification of the intelligences the multiple intelligence test is used, for the classification by learning styles the Felder test is used, in both cases the previously defined rules are applied, and the system indicates the inference result, showing the classification.

The Interactive Learning module is composed by the Learning Path that is defined by the Pedagogical Services Module. This learning path holds the LOM (Learning Object Metadata) and the sequencing to be applied in the didactic unit. So, it must be conceived in a series of tasks (activities, workshops, resources, practices, etc.) that are designed and conditioned by the course which is offered to the student through a user interface, adapted to their characteristics, thanks to the data provided by the Dynamic Student Module. The students interactions with the tasks in the interface produces an amount of information, where this interaction is stored (log files) as well as the results of the task accomplishment, when comparing the answers or actions of the student and those from the Module Pedagogical Services defined in the course.

The analysis of this information (log files and results) using data mining techniques provides the progress of the student, which can be take it apart into results or achievements that will move the students' motivation and the current knowledge level or current learning acquired.

The Pedagogical Services Module codifies teaching methods that are appropriate for the objective field and student. It is the implementation engine from the adaptive system. Initially the pedagogical method is most appropriate (which responds to the question how to teach?) due to the data provided by the Dynamic Student Module (IM and learning style). One of the most relevant functions in the Pedagogical Services Module is to select the most appropriate educational intervention. This is done by comparing the Dynamic Student Module with the expected results of the Interactive Learning Module (Course), the different points of view observed are pointed to the Pedagogical Services Module that takes a corrective action according to a set of possible alternatives to guide the student in the problems solution.

Figure 2 shows the process flow for the determination of the pedagogical method and the tracing of the learning path.

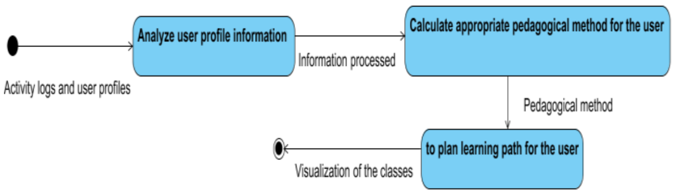

Fig -2: The process flow for the determination of the pedagogical method. Source: own elaboration.

1-The input to determine the pedagogical method are the activity logs and the user profiles.

2-The data for the calculation of the method are analyzed and classified.

3-With the information processed as input, we proceed to calculate the pedagogical method for the user (Here the neurodifusa network is used).

4-The neural network calculates the values for each type of pedagogical method.

5-The learning path is planned for the user, making an adequate visualization of the classes according to the learning method.

Table1 shows the pseudocode of the algorithm for determining the pedagogical method to use with the student.

Table -1: Pseudocode of the algorithm for determining the pedagogical method to use with the student.

Algorithm 1. Determine pedagogical method
Entries: student data
Outputs: pedagogical method

START
List of input neurons: = They are obtained from Mamdani's network
List of entries: = 0
For each neuron in the list of input neurons do
From i: = 1 to the size of the list of input neurons with step 1 do
If list of input neurons (i) == "sequential" then
List of entries (i): = Student data (learning style: "sequential")
But if list of input neurons (i) == "global" then
List of entries (i): = Student data (learning style: = "global")
But if list of input neurons (i) == "visual" then
List of entries (i): = Student data (learning style: = "visual")
But if list of input neurons (i) == "verbal" then
List of entries (i): = Student data (learning style: = "verbal")
But if list of input neurons (i) == "active" then
List of entries (i): = Student data (learning style: = "active")
But if list of input neurons (i) == "reflective" then
List of entries (i): = Student data (learning style: = "reflective")
But if list of input neurons (i) == "sensory" then
List of entries (i): = Student data (learning style: = "sensory")
If not
List of entries (i): = Student data (learning style: = "intuitive")
The list of entries is sent to the Mamdani network (Entry List)
The Mamdani network makes the classification
List of output neurons: = They are obtained from Mamdani's network
Pedagogical method: = 0
For each neuron in the list of output neurons do
From i: = 1 to the size of the list of output neurons with step 1 do
If the list of output neurons (i) == "sequential" then
Pedagogical method (i): = exit neurons (i)
But if list of output neurons (i) == "global" then
Pedagogical method (i): = exit neurons (i)
But if list of output neurons (i) == "visual" then
Pedagogical method (i): = exit neurons (i)
But if list of output neurons (i) == "verbal" then
Pedagogical method (i): = exit neurons (i)
But if list of output neurons (i) == "active" then
Pedagogical method (i): = exit neurons (i)
But if list of output neurons (i) == "reflective" then
Pedagogical method (i): = exit neurons (i)
But if list of output neurons (i) == "sensory" then
Pedagogical method (i): = exit neurons (i)
If not
Pedagogical method (i): = exit neurons (i)
Return Pedagogical Method
END

Figure 3 shows the course management interface in the intelligent tutor system.

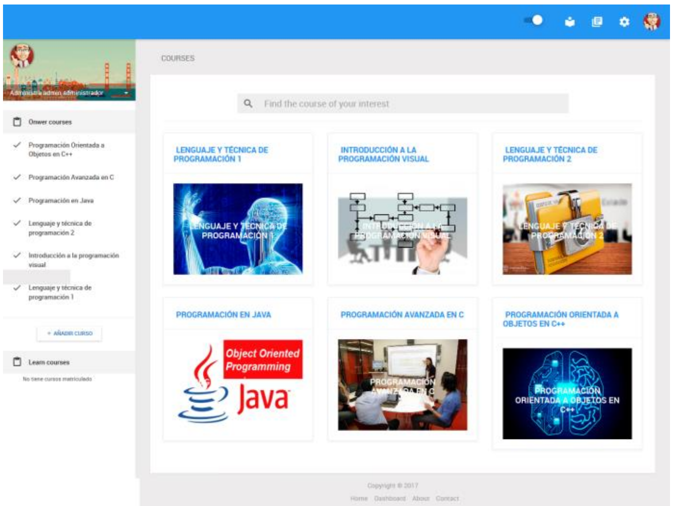

Fig -3: Course management interface. Source: own elaboration.

Figure 4 shows the student's intelligence map based on multiple intelligences and the pedagogical method.

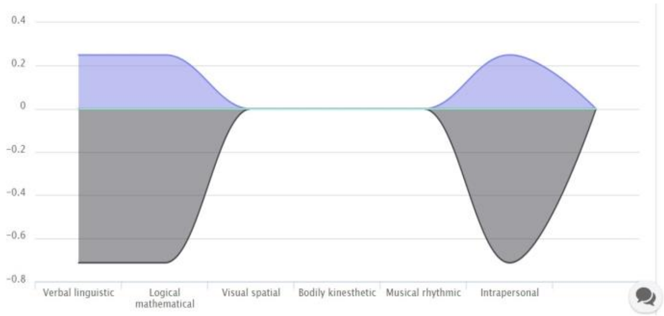

Fig -4: Student's intelligence map. Source: own elaboration.

3.3 Preliminary validation of the model

In order to evaluate the proposed model, an experiment was carried out on 54 students of the Computer Engineering course. The objective of the study is to verify how the variable number of attempts necessary to solve a problem when receiving the instruction through the proposed model is affected.

Students were divided into two randomly formed groups. During 10 days the students of both groups faced one exercise per day, the same for each group. For the experimental group (Eg) to register and start working on the courses, the support described in this work was counted. For the control group (Cg) we continued working from the static analysis. The conformation of the groups was 27/27 (Eg/ Cg), with students concluding the first semester, in the subject "Programming".

Table-2: Results of the first experiment.

Number of Repetitions	Experimental Group	Control Group	P value
1	0,58	0,23	0,0162
2	0,41	0,10	0,0190
3	0,01	0,37	0,0034
4	0,00	0,21	0,0118
5	0,00	0,09	0,0110

A total of 2059 solutions were sent, 994 correct and 1065 incorrect, for 48.27% effectiveness. Of the 994 problems solved, 442 did so on the first attempt.

Chart -1 describes the behavior of the variable number of attempts, where it is represented the times that a problem was solved after N incorrect attempts according to the results evidenced in Table-2.

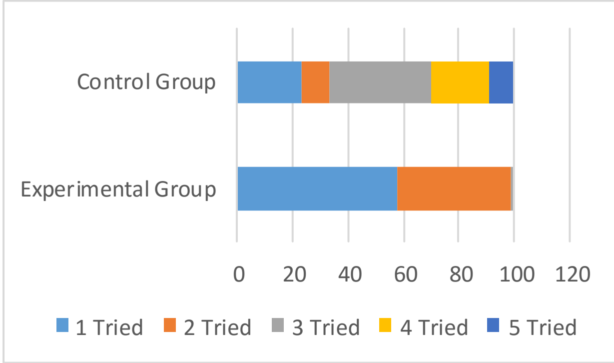

Chart -1: Results obtained of the first experiment.

On the data obtained from these measurements statistical tests were performed to determine if there were significant differences between the attempts required for the groups. For both cases, when performing a test of hypothesis relative to the difference of two proportions of samples from binomial distributions, significant differences were obtained. The P-value for the test resulted in the majority of cases less than 0.05, so that the null hypothesis could be rejected with high confidence levels.

The above allows to affirm that the students who were exposed to the model described in this work needed, on average, a smaller number of incorrect attempts to solve the problems.

4. CONCLUSIONS

The ISCO system built by combining ITS and multiple intelligences promotes a structured and organized knowledge, which accelerates the learning process concerning the students.

In the proposed ITS, each concept to be learned is enriched, according to different approaches with coaches, video tutorials, simulators, social interaction and other means that make knowledge acquisition easier.

The ISCO system is very easy to maintain and update, the modifications in the types of intelligences and the resources, are very simple, as well as the addition of new ones.

REFERENCES

[1] V. Aleven, «Rule-Based Cognitive Modeling for Intelligent Tutoring Systems,» de Advances in Intelligent Tutoring Systems, Springer-Verlag, 2010, pp. 33-62.

[2] S. J. Navarro, X. Canaleta, D. Vernet, X. Solé, V. Jiménez and N. Costa, "Motivation, demotivation, motivation and collateral damage" of XX. Conference on Teaching University of Informatics, Oviedo, 2014.

[3] I. H. Losada, Educational software design for the teaching of object-oriented programming based on the taxonomy of Bloom, Rey Juan Carlos University, 2012.

[4] L. Ríos, M. Lezcano and A. Díaz, "Conceptual Maps and Intelligent Agents in Prolog Teaching", CD Memoirs of the Second Cuban Congress on Local Development, Granma, 2009.

[5] P. A. Kinsumba, M. J. Becerra Alonso and R. Lau Fernández, "Prediction of Academic Success in the Computer Science Career of Agostinho Neto University," Pedagogical Reference, vol. 4, no. 143-146, 2016.

[6] J. Beltrán, H. Sánchez and M. Rico, "Quantitative and qualitative analysis of the learning of Programming I at the Central University of Ecuador," Technological Journal ESPOL, vol. 28, no. 194-210, 2015.

[7] A. J. Olier Quiceno, A. A. Gómez Salgado and M. F. Caro Piñeres, "Design and Implementation of a Teaching Tool for Introduction to Object Oriented Programming," IEEE LATIN AMERICA TRANSACTIONS, vol. 15, no. 97-102, 2017.

[8] H. Gardner, Structures of the mind: the theory of multiple intelligences, Mexico: Economic Culture Fund, 2016.

[9] K. Brawner, H. Holden y e. al., «Recommendations for Modern Tools to Author Tutoring Systems.,» de Proceedings of the Interservice/Industry Training Simulation & Education Conference (I/ITSEC), 2012.

[10] K. VanLehn, «"The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems",» Educational Psychologist, pp. 197-221, 2011.

[11] G. Paviotti y P. G. Rossi, «Intelligent Tutoring Systems (ITS): a European point of view,» de Intelligent Tutoring Systems: an overview, 2012, pp. 9-15.

[12] M. J. Eagle y T. Barnes, «Evaluation of Automatically Generated Hint Feedback,» de Proceedings of the International Conference on Educational Data Mining. S. K., Memphis, Tennessee, USA, 2013.

[13] M. Chi, K. VanLehn y e. al., «Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics,» de Proceedings of the International Conference on Intelligent Tutoring Systems, 2010.

[14] J. Dabolins y J. Grundspenkis, « "The Role of Feedback in Intelligent Tutoring System",» Applied Computer Systems 14, pp. 88-93, 2013.

[15] B. J. Reiser y J. R. Anderson, «"The LISP tutor: it approaches the effectiveness of a human tutor",» de Byte, 1985, pp. 159-175.

[16] G. Weber y P. Brusilovsky, «"ELM-ART: An adaptive versatile system for Web-based instruction",» International Journal of Artificial Intelligence in Education 12, pp. 351-384, 2001.

[17] A. Mitrovic, «"An intelligent SQL tutor on the web",» International Journal of Artificial Intelligence in Education (IJAIED) , pp. 173-197, 2003.

[18] N. Martínez, « Model for designing Intelligent Teaching-Learning Systems using Case-Based Reasoning,» Doctoral Thesis, Central University of Las Villas., 2009.

[19] T. Dadic, «Intelligent Tutoring System for Learning Programming,» de Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation, 2011, pp. 166-186.

[20] E. J. Altuna y L. Guibert, «Intelligent Tutoring Systems and their impact on teaching programming,» Memories of the XV International Congress of Informatics in Education, INFOREDU 2013, La Habana., 2013.

[21] M. Gujarathi y M. Sonawane, «Intelligent Tutoring System: A Case Study of Mobile Mentoring for Diabetes,» International Journal of Applied Information Systems, vol. 3, nº 8, pp. 41-43, 2012.

[22] H. Meshref y I. Mohamed, «Intelligent Tutoring Systems: A New Proposed Structure,» de Proceedings of the International Conference on Advances in Computing, Communications and Informatics, TNadu,India, 2012.

[23] B. du Boulay y T. del Soldato , «Implementation of Motivational Tactics in Tutoring Systems: 20 years on,» International Journal of Artificial Intelligence in Education , pp. 170-182, 2015.

[24] I. T. Machín Torres, «Main difficulties presented by students in the teaching-learning process of Object Oriented Programming,» Research Report, José Martí University of Sancti-Spíritus, 2013.

[25] I. T. Machín Torres, "Model to improve the assimilation of contents in programming," IJARCCE, vol. 6, no. 47-53, 2017.

[26] C. Walls, Spring in Action, Shelter Island: Manning Publications, 2015.

[27] P. Fisher and B. D. Murphy, Spring Persistence with Hibernate, Apress, 2010.

[28] I. T. Machín Torres, "Intelligent tutoring system for the application of programming through intelligence analysis," Scientific, vol. 29, pp. 1 -16, 2017.

BIOGRAPHIES

Ivelisse Teresa Machín Torres. Engineer in Computer Science Professor. Researcher in the area of educational technology.

Vivian Estrada Sentí.
Ph. D Computing, Professor
Adviser postgraduate, Havana, Cuba