ISSN: 2594 - 2905

FORMAL THINKING, A DETERMINING FACTOR IN TEACHING CLASS DESIGN UNDER UML IN PROGRAMMING

FORMAL THINKING, A DETERMINING FACTOR IN THE TEACHING OF CLASS DESIGN UNDER UML IN PROGRAMMING

Ann Margareth Meza Rodríguez [1]

https://doi.org/10.61117/ipsumtec.v7i2.340

[1]Doctoral student in Multimodal Educational Environments and Systems. Universidad Rosario Castellanos-Unidad de Posgrado- Doctorado en Ambientes y Sistemas Educativos Multimodales. ann.meza@gmail.com, 5525095801. Av. 506 O, Colonia San Juan de Aragón II Sección, Alcaldía Gustavo A. Madero, C.P. 07696, Mexico City. https://orcid.org/0000-0003-0754-7140

Abstract-- This article aims to show the documentary research of some of the research works that have been carried out regarding the process of teaching design and modeling in programming, based on formal thinking, and the analysis of the descriptive literature review is socialized, as well as the quantitative results obtained from the application of a statistical method. In this way, it is interesting in this document to show the importance of developing or strengthening formal thinking in the process of teaching the design and modeling of classes in programs under the object-oriented paradigm under the UML (Unified Model Language) standard. For this purpose, the reader will find how and what were the criteria considered to select the works reviewed in the method section, as well as the methodology for obtaining quantitative data obtained through three questionnaires, which helped to determine whether the subjects of study have developed formal thinking, or only have developed concrete thinking. Consequently, in the results and discussion section, the selected and reviewed works are presented, presenting through a synthesis the most outstanding details. At the end, the perspective and relevance of this topic for the main research from which this article derives is presented. And, finally, the conclusions answer to the main question of this paper is given: Why is it important that higher education students have strengthened formal thinking in the process of teaching the design and modeling of classes in programs under the object-oriented technique with UML?

Keywords: Teaching strategies, formal thinking, concrete thinking, educational resources, object-oriented program design, abstract classes.

Abstract -- This article aims to show the documentary investigation of some of the research work that has been carried out regarding the process of teaching design and modeling in programming, based on formal thinking, and the analysis of the descriptive bibliographic review is socialized, as well as the quantitative results obtained from the application of a statistical method. In this way, this document is interested in showing the importance of developing or strengthening formal thinking in the process of teaching the design and modeling of classes in programs under the object-oriented paradigm under the UML (Unified Model Language) standard. To do this, the reader will be able to find how and what were the criteria that were considered to select the reviewed works in the method section, as well as the methodology for obtaining quantitative data that was obtained through three questionnaires, which helped determine whether the study subjects have developed formal thinking, or only have developed concrete thinking. Consequently, in the results and discussion section, the selected and reviewed works are presented, presenting the most outstanding details through a synthesis. At the end, the perspective and relevance that this topic takes for the main research from which this article emerges is presented. And finally, the conclusions answer the central question of this document: Why is it important for higher education students to have formal thinking strengthened in the process of teaching the design and modeling of classes in programs under the object-oriented technique with UML?

Key words - Teaching strategies, formal thinking, concrete thinking, educational resources, object-oriented program design, abstract classes.

INTRODUCTION

This article is generated as a derivative product of the research proposal entitled "Analysis of the efficiency of a didactic methodology based on the development of formal thinking in Computer Systems Engineering students in Milpa Alta, Mexico City". Its main objective is to analyze the efficiency of the implementation of a didactic methodology that develops formal thinking to achieve the design and modeling of abstract classes under the UML standard, this research is aimed at students taking the subject of Object Oriented Programming of the Computer Systems Engineering course at the Technological Institute of Milpa Alta.

That is why it is of interest to establish in this article why it is important to consider whether students have developed formal thinking, since it is considered as one of the factors that influence the achievement of learning focused on modeling and design of class diagrams. Because students must learn the semiotics of the UML (Unified Model Language) standard. And, through its three main branches: semantics, syntax and pragmatics, they must learn not only to interpret, also to solve problems by means of diagrams or schemes.

This is due to the fact that the teaching of the object-oriented software design technique has recently become a necessity for university careers focused on computer science or computer systems, since this technique promotes the analysis and design of solutions to problems related to software [1]. Thus, in the professional field, having this competence is of utmost importance, since currently, most web pages, computer applications and mobile devices, as well as software in general, are developed under this paradigm [2].

Currently the Object Oriented technique is used to develop complex systems, through which interact different disciplines such as: software development, hardware development, information sciences (internet) and data mining (information management through databases), among others [3]. All of them working together have allowed advances in technology in sectors such as: telecommunications, medicine, banking, agriculture, commerce, education, culture and many areas of the productive sector and services that aim to automate processes in almost any industry [4].

Accordingly, Mexico is no exception. Today, according to the Labor Observatory of the Government of Mexico [5], professionals working in the area of engineering with a specialty in computer science are those who have the greatest employment opportunities. And according to the INEGI (National Institute of Statistics and Geography) press release NUM. 310/19, published in June 2019 [6], Mexico City is the entity that has the largest population employed in jobs that have to do with ICT (Information and Communication Technologies), and also 58% of employees have as their level of schooling to ES (Higher Education), that is, a large number of professionals graduated from IES (Higher Education Institutions) come to be hired within the CDMX.

This study is relevant because the Instituto Tecnológico de Milpa Alta is a public university belonging to the Tecnológico Nacional de México, located in the south of the CDMX, in the municipality of Milpa Alta. This institution is characterized by receiving new students in two periods per year, January-June and August-December. It has four degree programs: Engineering in Business Management (IGE), Engineering in Biochemistry (IBQ), Engineering in Food Industries (IIA) and Engineering in Computer Systems (ISC), where the ISC program is the second of the four plans that has a high demand for admission compared to the other plans offered by the institution.

Below are the statistics obtained from the School Control System, between the period January-June/2018 and August-December/2022, 583 students took the first semester course, Fundamentals of Databases, of which 145 did not pass; 509 students took the second subject, Object Oriented Programming, of which 153 did not pass; 377 students took the third semester subject, Data Structure, of which 75 did not pass; and 284 students took the fourth semester subject, Advanced Topics in Programming, of which 36 did not pass [7].

As can be observed, in the first instance, the number of students who took programming subjects from the first to the fourth period decreases as the semesters progress, which indicates that there was an average dropout rate of almost 50% of the enrollment in the first half of the course. Secondly, the programming course with the highest failure rate was the second semester, Object-Oriented Programming, which showed a considerable decrease in enrollment at the end of semester two. And finally, the statistics of these first four semesters are considered, since, according to the study plan, these are the mandatory subjects of the course that apply the object-oriented technique in software design, also serve as the basis for the subjects of the entire course, so that the competencies developed in them are fundamental for the students' training process.

In this case, the target population to benefit is those students of higher education who are pursuing a degree in Computer Systems Engineering, whose curricular model is focused on developing and/or strengthening formal thinking, in order to learn to design programs under the object-oriented programming technique, identifying in the first instance the importance of problem solving abstract class modeling.

DEVELOPMENT

Methodology

In the first instance, a systematic literature review was carried out, for which the proposal of [8] was taken as a reference, so the following activities were carried out in a general manner:

1) Question: in this article the objective will be to establish the relevance of the development or strengthening of formal thinking in the teaching of class design in object-oriented programs.

2) Protocol: The criteria for selecting or excluding literature were established, the search engines in which the searches were carried out were determined, such as: Redalyc, Scielo, Google Scholar and Dialnet. As well as the search period, which is between 2010 and 2024.

3) Search: background information was collected, taking as reference projects focused on taking the development of formal thinking as a factor to be considered in the teaching-learning process of program design and modeling, as well as teaching strategies in programming. Key words used were: teaching-learning, object-oriented programming, abstract thinking, formal thinking, class diagrams, teaching UML class diagrams.

4) Selection: Studies that met the eligibility criteria were selected, which were those that attempt to explain what is considered as abstract or formal thinking in the discipline of programming and problem solving, and how promote a better experience of the teaching process in the classroom, in this case, through the use of a scheme that allows designing and modeling a program under the object-oriented technique.

5) Extraction: From each selected literature, the main data were extracted, such as: authors, research topic, sources of information, general objective of the study, methodology, results obtained, recommendations for future studies and key words.

6) Evaluation: In this step, both the methodological quality, i.e., whether both bias and validation of the studies found were evaluated. Likewise, the references used in each study found.

7) Synthesis: A table was prepared with the elements obtained from the extraction, summarizing each of elements of the selected literature. And, from there The contents were identified and classified to obtain the categories shown in the discussion and analysis of results section.

8) Writing: an analysis and discussion of the results obtained was made, as well as the conclusion made by reviewing the literature on the importance and relevance of the development of formal thinking in the teaching-learning process of modeling object-oriented solutions through schemes called abstract class models.

And together with this systematic documentary review, a statistical method was applied on a descriptive research; whose objective is to establish a diagnosis of the presence or not of formal thinking in the students, to show it as a factor, of the many that exist, to be considered in the teaching-learning process, and to establish that the absence of this strengthened formal thinking can mean delays in the teaching process. Thus, the hypothesis proposed is: The presence of formal thinking in the students allows them to be placed in the abstract field so that they can solve problems the semiotics of class diagrams with UML in the subject of object-oriented programming in the second semester of the Computer Systems Engineering course at the Instituto Tecnológico de Milpa Alta. Therefore, the following activities were carried out:

1) Data collection: The primary source was visited at the beginning of the , and to avoid bias, the application of each and every instrument was carried out on the same day and at the same time, which gave internal validity to the study. The instruments used for data collection were a socio-academic questionnaire, a test to measure the level of existing concrete thinking, a test to measure the presence or absence of formal thinking, and a test to measure the level of strengthening of formal thinking. These were applied systematically using an online tool. Simple observation was also used, which was recorded in a digital logbook, in which all the events generated data collection were backed up. The population of students who were taking the course in the period January-June/2024 was 56. Considering a confidence level of 95 percent and an error of 5 percent, the sample size considered is 49 students, who were chosen randomly.

Data:
Population, $N = 56$ Margin
of error, $e= 0.05$
Confidence level, 95% therefore, $k= 1.96$ Probability of
success, $p = .5$

Probability of failure, q= 1-p= .5

$$n= \frac{k^{2}qpN}{e^{2}(N\text{-}1)\ 2+kpq} = 48.98 \approx 49 alumnos$$ Eq. (1).

2) Counting and Analysis: Once the data were applied to the instruments, a procedure was carried out to analyze the data that statistically reflect the information, in order to identify those elements that will serve to interpret the data obtained.

3) Presentation and interpretation: Once the data had been processed and represented in tables, graphs and/or figures, a process was carried out to interpret the information obtained to determine whether the hypothesis was fulfilled or not.

DISCUSSION AND ANALYSIS OF RESULTS

Importance of formal thinking in problem solving.

The graduate profile of a TecNM Computer Systems Engineering student is made up of 15 competencies, of which the ability to apply new technologies to solve problems in their work environment is taken as a reference for this research [9]. To solve these problems, one of the many skills that every engineer must strengthen is abstract, logical and deductive thinking; which as a whole we will call formal thinking in this research.

If higher education students have this formal thinking, it could allow them to understand, analyze and model the problems of information and knowledge in today's society, and thus be able to come up with efficient and effective solutions [10].

It is worthwhile at this point to clarify what is meant by formal thinking for this research, and its difference with concrete thinking, which, although in their conceptualization are dichotomous, in the teaching process should complement each other, in order to propose a series of teaching strategies for the design and modeling of programs under the object-oriented technique.

Despite this consideration, it has been identified that, in most of the studies selected for this article, teacher-researchers prefer to remain only in the teaching of concrete thinking, which consists of observing a real-life phenomenon, which is classified and explained only logically in terms of how it is composed and its characteristics. According to Piaget, this concrete thinking develops from early childhood to adolescence [11].

Unlike concrete thinking, formal thinking, according to Piaget, develops between the ages of 11 and 15,
where students are able to solve more complex problems by making use of concepts and theories to understand the behavior of
These phenomena will now be conceptualized as objects, and it will be able to identify and determine how it is linked to other objects, either with the same or different characteristics [12].

Vygotski, on the other hand, refers to this stage as concepts in action, since the higher level student takes mental characteristics from previous stages, and, if new mental processes exist, they can construct solutions to problems of a hypothetical-deductive type. Marta Uribe, describes that, in this university stage, the capacity to use symbols to represent objects, categories, and operate between them is also developed. However, if in the teaching process the teacher only presents didactic sequences focused on using only concrete thinking, students will not be able to develop or strengthen this formal thinking, since there will be no stimulus to develop new skills [14].

[14], [15], [16] and [17] present studies in which they point out that, regardless of age, or the environment and stimuli to which university students have been subjected, approximately 30% have not developed formal thinking, and almost 50% have managed to develop such formal thinking. This implies that teachers should establish didactic strategies or methodologies that promote the development of formal thinking, , , the negative impact on the students' trajectory is very marked, becoming one of the many factors that influence school dropout.

[18] and [19], agree that there is a link between formal thinking and the development of abstractive capacity, since this ability is key in the training of any engineering, and for this case in that of computational sciences that involves problem solving from the observation of phenomena [19].

Then, in order to propose possible solutions, students should be taught to carry out a reasoning process that allows reducing the complexity of a problem, dividing it into modules and in turn each module could be conceptualized or generalized through the use of the patterns that are extracted. And once you have the design of each and every one of the modules, the process is to link them all together. The intention should be that all these solutions can be approached from any discipline of computational sciences, such as: telecommunications, telecommunications, computer science, computer science and information technology.

, artificial intelligence, and of course, programming in any of its paradigms [18].

It is then pertinent to take the definition of Garrido, who defines formal thinking as a process that has to do with mental structures, and that this process takes from the object of study details that are relevant to simplify a problem. And, in order to conceptualize possible solutions to a problem, it is carried out through the development of models at different levels of abstraction [20].

UML Semiotics for the design and modeling of object-oriented program classes

In the previous section, it was identified that the development of formal thinking implies that students can operate objects through theories, and for this they must have the ability to synthesize large amounts of information. Therefore, it is convenient to use schemes that, on the one hand, allow the students to combine in modules or blocks that synthetically represent the conceptualization of the information, and, on the other hand, relate these blocks of information in a more complex way.

That is why the competence to carry out representations by means of figurative systems should not remain at the stage of concrete thinking. Brandi, in his graduate work, shows the importance of graphic literacy, which should consist of teaching the syntactic characteristics and taxonomy of a figurative system [21]. This implies learning to generate and interpret complex conceptual developments, which are synthesized through semiotics.

Therefore, if we speak from the perspective of semiotics, UML (Unified Model Language) diagrams allow to graphically represent logical sets of elements of one or more interrelated models [22]. These UML diagrams have been used in the computer science industry for their ease of use and representation, because, being a universal language, and under a defined syntax, the same figurative element can be interpreted from different perspectives, where each one of them is a part of the area of specialty with which the solution will be addressed, for example: database, telecommunications, programming, software engineering, servers, cloud computing, computer security, etc.

Let us now consider that in programming there are two ways to carry out the modeling process. On the one hand, there is the specification abstraction, and on the other hand the implementation abstraction [20]. Where the first type of abstraction, serves to identify the features and

For this type of abstraction, structures such as , abstraction level diagrams, use case diagrams or class diagrams, among others, are regularly used under the UML standard. While the second type of abstraction is used to determine how the modules will interact and thus give functionality to the program; for this type of abstraction is also performed under the UML standard, and structures such as: flowcharts, state diagrams, sequence diagrams, activity diagrams, among others, are regularly used [23].

Then, UML diagrams can be elaborated to represent static structures, use cases, sequences, user-machine-user collaborations, and state diagrams [24]. Within the classification of diagrams to represent static structures, there are class diagrams, which are involved in software engineering processes, and which are developed as part of the design of a proposed solution.

These class models can be seen as the representation of reality, and each sign used represents a part of that reality, and they are defined as object-oriented models. Therefore, UML class diagrams have an advantage over imperative, declarative or functional modeling languages [23]. Because the development of the class diagram starts by using concrete thinking, where the object, its structure and characteristics are identified; and with formal thinking it is determined, for each of the identified objects, how they are created, how they work or how they interact with each other already in the programming.

Analysis of statistical results

For this article, a questionnaire, three tests, and an observation logbook were developed. The questionnaire was implemented to identify the socio-academic profile and academic habits of the study subjects. The tests were implemented to determine the level of thinking that the study subjects had developed up to that moment, and thus determine the percentage of students who presented a developed or strengthened formal thinking. And, the observation log was implemented to note possible improvements in the implementation of the main research, ideas, data not reported in the tests, and general observations of the application of tests. Therefore, the analysis and results obtained from the data collection instruments implemented are described below.

The socio-academic and study habits questionnaire allowed us to identify that:

- 80% are male
- Forty percent of the study subjects live in the municipality of Xochimilco, more than 30 minutes away.
- Ninety-six percent of the sample are single and have no children,
- Seventy-four percent of the study subjects do not work.
- 55% come from CETIS, a high school.
- Seventy-six percent of the study subjects have kinesthetic and auditory as their predominant learning styles.
- 80% of the study subjects have suitable or adapted spaces for their study sessions.
- Forty percent of the study subjects do not have a fixed schedule for their study sessions.
- 30% make diagrams to reinforce their knowledge.

As for the tests, the first test allows us to determine whether the subjects have developed concrete thinking by identifying concepts and definitions, as well the elements that make up a class diagram.

Then, when applying the first test to the study subjects, they were asked to identify the elements of a class, the syntax, relate concepts with their respective definition, and finally interpret a complete diagram. The following results were obtained:

Table 1. Concepts that allow us to identify whether the study subjects have concrete thinking present, through the identification of concepts and classification of information.

Specific concepts	Percentage of students who responded correctly
Identifies elements that make up a class and their order	76%
Syntax of a class	78%
Identifies the definition of a class	93%
Identifies the definition of object	57%
Identifies the definition of an attribute	98%
Identifies the definition of a method	78%
Identifies the concept of abstraction	72%
Identifies the elements of a diagram of classes: class, access level, data type, parameters, associations.	92%

This implies that, from the concrete point of view, the subjects of study identify this type of diagrams, their elements, definitions, concepts and how they interact with each other, i.e., concrete thinking is present.

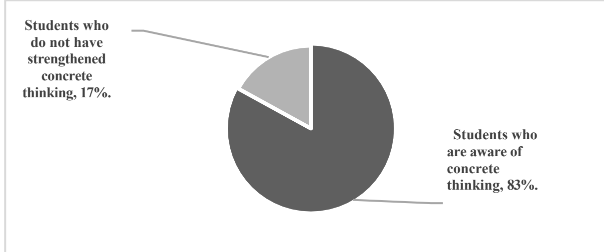


Figure 1. Comparative graph of the presence of concrete thinking in the study subjects.

The second test was applied to identify whether these same study subjects have developed the formal thinking to solve a problem that results in the development of a class diagram. For this purpose, a reading was given describing the symbology of the UML standard, in which cases to use such symbology, and at the end of the reading, three application examples were shown.

Then, once the reading was completed, two problems were presented, with different approaches, both in data and in writing, 48% were able to model only the attributes of the classes; and only 32% were able to determine attributes and methods; establish the interaction, links and use of syntax between the different classes correctly, 20% answered correctly; the remaining of each criterion did not answer.

Table 2. Criteria that allow to identify if the subjects of study have formal thinking present, through the resolution of design exercises and class modeling with UML.

Measurement criteria	Percentage of students who were able to complete the exercise of solving the following questions problems
Modeling a class using the semiotics of the UML standard	48%
Model attributes and methods using the semiotics of the UML standard.	32%
Model the interaction, links and use of syntax between the different classes	20%

This implies that, although they know the concepts, when articulating them, there is no real understanding of how to do it, and this implies a barrier for the next stage, which is the implementation through the interpretation of these diagrams using programming.

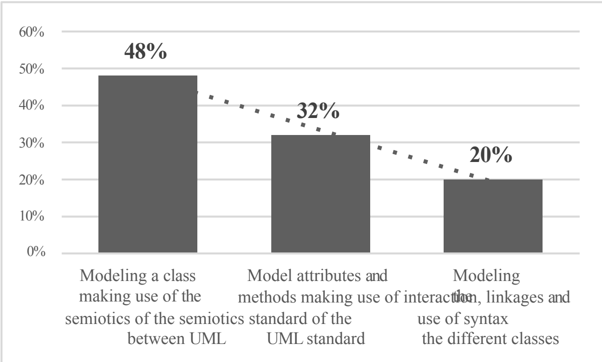


Figure 2. Trend graph, showing that the higher the complexity of the exercises, the lower the number of students answering correctly.

Finally, the third test identifies whether the study subjects are able to move from the class diagram to programming by interpreting a class model provided to them, in order to determine whether formal thinking has been strengthened.

When applying this third test to the same sample, which consists of identifying whether the study subjects can implement the interpretation of the diagram in the programming, it was observed that only 30% were able to create the classes and link them correctly according to the established diagram, and 46% remained only in the construction of the classes, without being able to link them through programming; the remaining 24% did not respond.

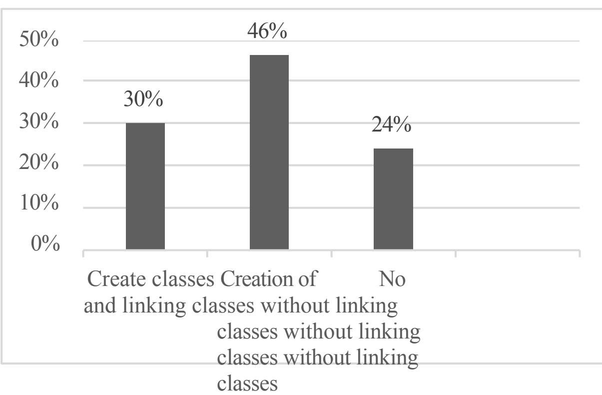


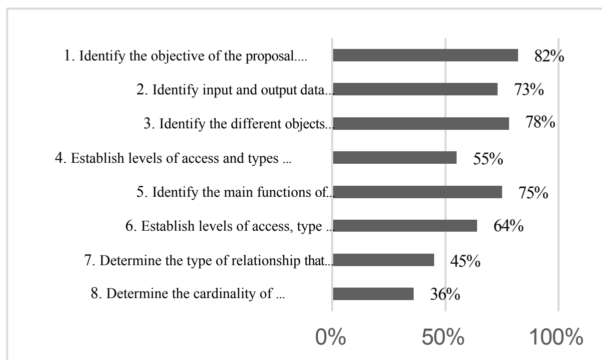
Figure 3. Graph of the third test, where the aim is to identify whether the study subjects have strengthened formal thinking.

This indicates that formal thinking has not yet been strengthened in most students at this stage of their education.

Another important factor that was observed and recorded in the logbook was that the study subjects do not have a method for solving this type of problem. In the second test, the study subjects were asked: *Order the following steps that are carried out to elaborate a class diagram.* Table 4 shows the steps and the number of people who placed the sequence correctly. This implies that, by not having defined the way in which a class diagram is constructed has an impact on the final result of the diagram, as well on the correct use of semiotics.

Number of students who correctly answered the order of the steps to be carried out in the development of a class diagram.

Step	Students that responded in correct order
1. Identify the objective of proposed solution.	45
2. Identify the input and output data of the proposed solution.	40
3. Identify the different objects involved	43
4. Establish access levels and data types of attributes	30
5. Identify the main functions of each of the objects.	41
6. Establish access levels, type of output data and necessary parameters for each of the methods.	35
7. Determine the type of relationship that may or may not exist between the objects.	25
8. Determine the cardinality of relationships between objects.	19



Percentage of students who correctly located the order in which each of the steps in the process of developing a class diagram should be carried out.

CONCLUSIONS

The hypothesis of this article is: "The presence of formal thinking in students allows them to be placed in the field of the abstract so that they can solve problems using the semiotics of class diagrams with UML in the subject of object-oriented programming in the second semester of Computer Systems Engineering at the Technological Institute of Milpa Alta"; the data collection instruments were developed with the intention of showing why it is important for higher education students to have formal thinking strengthened at this stage of their education, and why it is important for them to be able to solve problems by using the semiotics of class diagrams with UML in the subject of object oriented programming in the second semester of the Computer Systems Engineering degree program at the Technological Institute of Milpa Alta.

training as computer systems engineers.

The development or strengthening of formal thinking is not a recently created concept, however, in recent times it has been taken up with greater emphasis in the teaching processes, due to the complexity of the problems to be solved in recent times.

The emphasis on the way in which the solutions to problems that, in this case, will be solved through computational sciences, will be represented is due to the fact that this solution or solutions are built through the abstraction of real world objects, which allow the design and modeling of classes that will be programmed later on.

As could be identified in the literature selected for this article, as well as in the tests applied, and in the observation log, the students understand the syntax and taxonomy well; however, when asked to develop a model from a given problem, few are those who manage to develop a diagram with the semiotics that marks UML, as well the implementation in programming. [25] highlights the importance of first understanding that the resolution a problem to be solved through programming can be approached using mental strategies such as modeling, abstraction, decomposition and heuristic reasoning [25].

Therefore, the importance of the development of this abstractive capacity derives from the relevance of modeling interpretations for problem solving; then, this ability allows to decompose a problem, which helps in the analysis and understanding of it, and thus create models to propose possible solutions. Thus, formal thinking becomes an important factor for science and engineering in any of its branches, especially in the development of science and technology.

[18] refers that the ability to solve a problem implies that first real-world objects of study are identified, and from them characteristics and functionalities are obtained, which will be linked to each other, and each object is called a module. Therefore, when having more complex problems, several objects of study are identified that make up a whole, and therefore, the same process must be carried out for each one. Hence, for a more complex problem, more components will be part of the solution. Therefore, it is necessary for programmers to have one or more tools that allow them to identify each and every one of the modules, as well as the relationships between them.

As Alvaro Carbajal rightly mentions, "models are instruments or conceptual schemes by which human beings try to systematically articulate the knowledge they have of experience through the research process. Both contribute to having an image of reality..." [26].

Given these characteristics, and considering that the resources to be used must meet the needs of context and current events, it is now time to carry out more in-depth research, the first one proposed is one that analyzes whether the implementation of a didactic methodology that strengthens formal thinking, making use of new learning environments, can serve as support for those teachers who wish to improve their educational practices.

Another research that can be derived from this article is to create or propose work spaces that have an explicit meaning, in which communities of are integrated, where interaction among peers and with teachers becomes relevant. Where those activities, resources, evaluation instruments and didactic sequences allow to provide a guide that supports the optimization of this teaching process.

Finally, it is concluded that, although the factors that influence the success or not of the teaching process class diagrams are multifactorial, formal thinking is a factor that does help in the understanding and construction of the scaffolding of higher education students, through the strengthening of mental structures, skills and abilities, which allow a better acquisition and apprehension of the semiotics of class diagrams developed with UML.

ACKNOWLEDGMENTS
I am grateful to the Instituto Tecnológico de Milpa Alta for allowing me to carry out this research project, which seeks to contribute a proposal for a didactic methodology that will have an impact on the areas of opportunity in the teaching-learning processes of the community of that institution.
The present work was developed within the framework of the studies of the Doctorate in Multimodal Educational Environments and Systems of the Universidad Rosario Castellanos for obtaining the degree.
To DTT. María Eugenia Rodríguez Paz. And the TC. Dr. Ana Beatriz Vega Cruz and Dr. José Jacobo Gómez Quiroz.

BIBLIOGRAPHY .
[1] Gayo, D. (2021). IX Jornadas de Enseñanza Universitaria de la Informática. Reflexiones y experiencias sobre la enseñanza de POO como único paradigma, July 2003. Madrid: Thomson-Paraninfo.

[2] Sánchez-García, J., Urías-Ruíz, M., Gutiérrez- Herrera, B. (2015). Análsisi de los problemas de aprendizaje de la programación orientada a objetos, 11-4, pp.289-304.
[3] Cervantes, J., Gómez, M., González, P., & García, A. (2016). Introducción a la Programación Orientada a Objetos. CDMX: UAM - Unidad Cuajimalpa.
[4] IBM (2018). The evolution of process automation. Retrieved from: https://www.ibm.com/downloads/cas/RJGDMZ2D
[5] National Employment Service (2023). Engineering. Retrieved from: https://www.observatoriolaboral.gob.mx/static/estudios-publications/Engineering.html
[6] INEGI. (2019). Statistics apropos occupations related to information and communication technologies national data 1. retrieved. from: https://www.inegi.org.mx/contenidos/saladeprensa/apro posito/2019/OcupaTIC2019_Nal.pdf
[7] Technological Institute of Milpa Alta 2023). Failure rate - Computer systems engineering by period January-June/2018 to August-December/2022. Retrieved from: Department of School Services.
[8] University of Navarra (2024). Steps for conducting a systematic review. Retrieved from: https://biblioguias.unav.edu/revisionessistematicas/pasos -perform-systemic-reviews
[9] TecNM - Milpa Alta Campus (2024). Graduation profile of the Computer Systems Engineering career. Retrieved from: https://itmilpaalta.edu.mx/pag/isc.php
[10] Serna, M., & Polo, J. (2014). Logic and abstraction in engineering education: a necessary relationship. Engineering Research and Technology, 15-2, pp. 299-310.
[11] Ibarra, L. (2022). Pedagogical thinking and concrete-abstract thinking. 9-1. Retrieved de: https://www.scielo.org.mx/scielo.php?script=sci_arttext &pid=S2007-78902021000800015
[12] UNIR (2022). Pre-operational stage: what it consists of and its importance at the children. Retrieved from: https://mexico.unir.net/noticias/educacion/etapa- preoperational/
[13] Vygotski, L. S (2007). Pensamiento y lenguaje en Vygotski, L. S. Obras escogidas II, Madrid: Aprendizaje. Visor. cited in Cano, A., Cognición en el adolescente según Piaget y Vygotski. ¿Dos caras de la misma moneda?, Boletim Academia Paulisata de Psicología 27, pp. 148 - 166
[14] Uribe, Marta (1993). El desarrollo del pensamiento formal y la docencia universitaria. Retrieved from: https://www.iisue.unam.mx/perfiles/articulo/1993-60-el-desarrollo-del-pensamiento-formal-y-la-docencia-universitaria.pdf
[15] Hernández, C. A., Ramírez, P., & Rincón, G. A. (2013) Mathematical thinking in university students. Eco Matemático, 4-1, pp. 4-10.
[16] Díaz Barriga, F. (1987). El pensamiento adolescente y el diseño curricular en educación media superior. Perfiles Educativos, 37, pp. 16-26. Retrieved from: https://www.iisue.unam.mx/perfiles/articulo/1987-37-el-pensamiento-del-adolescente-y-el-deseno-curricular-en-educacion-media-superior.pdf
[17] Vázquez, L., & Segarra, P. (2017). Development of formal thinking through analysis and evaluation of scientific information. Latin-American Journal of Physics Education, 11-2. Retrieved from: https://dialnet.unirioja.es/servlet/articulo?codigo=63534 24
[18] Serna, E. (2015). Logical interpretative and abstractive capacity. Libro electrónico. Medellín: Fondo Editorial ITM, Instituto Antiqueño de Investigación.
[19] Zapotecatl, J. L. (coordinator) (2018). Introduction to computational thinking: basic concepts for everyone. First edition. Mexico: Academia Mexicana de Computación, A.C.
[20] Garrido, A. (2018). Abstraction in programming. Granada, Spain: Department of Computer Science and A.I. ETS Computer Engineering. Retrieved from: https://ccia.ugr.es/~jfv/abstraccion.pdf
[21] Brandi, J. P. (2016). Graphics in teaching and learning. learning. Retrieved from: https://core.ac.uk/download/pdf/296390387.pdf
[22] Genova, G., Valiente M. C., & Nubiola, J. (2006). Models in UML: a semiotic approach. Novática, 180, March-April. Retrieved from: http://www.ie.inf.uc3m.es/ggenova/pub-novatica2006a.pdf
[23] Sommerville, I. (2011). Chapter 5. System modeling. In I. Sommerville, Software Engineering, pp. 118-147. Mexico, Mexico: Pearson Educación.
[24] Ferré, X., & Sánchez, I. (2011). Object Oriented Development with UML. Universidad Veracruzana. Retrieved de: https://www.uv.mx/personal/maymendez/files/2011/05/ umltotal.pdf
[25] B@UNAM (2022). Computational thinking and creativity. 172-9. Retrieved from: https://boletin1020.bunam.unam.mx/pdf/172.pdf
[26] Carvajal, Á. (2002). Theories and models: ways of representing reality. Comunicación, 01 (12), pp. 1-14. Retrieved from: https://www.redalyc.org/pdf/166/16612103.pdf
[27] Arenas, L. (2008). Asignatura de Programación Orientada a Objetos. Retrieved from: http://fcasua.contad.unam.mx/apuntes/interiores/docs/98 /opt/java.pdf
[28] Weisfeld, M. (2013). The Object-Oriented Thougth Process. 3rd ed. USA: Pearson Education.
[29] Bustos, G. (1999). Object Oriented Modeling: A Critical Evaluation. Retrieved from: https://www.sites.upiicsa.ipn.mx/archivos/profesores/hm ontes/cursos/orientada-objetos/23-2/tareas/tarea_2b.pdf

[30] Muñoz, J., Álvarez, F., Garza, L. A., & Pinales, F. J. (2005). Model for collaborative learning of object-oriented analysis and design. 5-1, pp.73-82.

[31] Gonzáles, E. I., López, A., Trujillo, V., & Rojas, R. (2018). Teaching and learning didactic strategy for software programmers. RIDE. Iberoamerican journal for educational research and development. Retrieved from: https://ride.org.mx/index.php/RIDE/article/view/402/20 79.

[32] López, L. (2007). Methodology for teaching learning object-oriented programming logic. Retrieved from: https://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2007/lometo.pdf

[33] Rodríguez, C. (2014). Some elements of the didactics of programming and computational thinking. [Bachelor's thesis]. Facultad de Ciencias, UNAM, CDMX. Retrieved from: https://repositorio.unam.mx/contenidos/ficha/algunos-elementos-de-la-idactica-de-la-idactica-de-la-programacion-y-el-pensamiento-computacional-285508?c=56d5A6&d=true&q=*:*&i=5&v=1&t=search_0&as=0

[34] Vargas, Gabino (2017). Didactic educational resources in the teaching-learning process. Cuadernos, 58-1, pp. 68-74. Retrieved from: http://www.scielo.org.bo/pdf/chc/v58n1/v58n1_a11.pdf

[35] Marrero, A. L., & Díaz, Maiky. (2022). Elaboration and implementation of educational resources to stimulate the competencies competencies. Retrieved de: http://scielo.sld.cu/pdf/uh/n295/0253-9276-uh-295- e1.pdf

[36] Macedo, L., Montemayor, G., Limón, D., Hinojosa, V., & Huerta, C. (2016). Educational resources. Generalities for their development and evaluation. Retrieved from: https://suayed.facmed.unam.mx/wp-content/uploads/2022/03/recursos_educativos_mar29.pd f

[37] Rosario Castellanos University (2024). Morphology of the PLE. Retrieved from: https://edu.rcastellanos.cdmx.gob.mx/posgrado/pluginfile.php/200/mod_resource/content/16/index.html.

[38] Agency for development cooperation in education, science, technology and innovation A.C. 2017). Socioconstructivist model of integral education. Retrieved de: http://www.uno.edu.mx/downloads/DocumentosOficiales/SocioconstructivistEducationalModelforComprehensiveEducation.pdf