Revista Iberoamericana para la Investigación y el Desarrollo Educativo
ISSN 2007 - 7467

https://doi.org/10.23913/ride.v15i29.2196

Scientific articles

UML: Una manera de representar, interpretar, analizar y desarrollar el pensamiento computacional

UML: A way to represent, interpret, analyze, and develop computational thinking

UML: Uma forma de representar, interpretar, analisar e desenvolver o pensamento computacional

María del Rocío Ramírez Jiménez
Universidad de Guadalajara, Lagos de Moreno, Jalisco, México.
mdelrocio.ramirez@academicos.udg.mx
https://orcid.org/0000-0003-1709-5539

Karime Pulido Hernández
Universidad de Guadalajara, Ocotlán, Jalisco, México.
karime.pulido@academicos.udg.mx
https://orcid.org/0009-0005-5672-7510

Carmen Elizabeth Rivera Orozco
Universidad de Guadalajara, Lagos de Moreno, Jalisco, México.
carmen.rivera@academicos.udg.mx
https://orcid.org/0000-0001-6588-6779

Norma Alicia Gómez Torres
Universidad de Guadalajara, Ocotlán, Jalisco, México.
norma.gtorres@academicos.udg.mx
https://orcid.org/0009-0004-0808-4111

Liliana Serrano Zúñiga
Universidad de Guadalajara, Ocotlán, Jalisco, México.
liliana.serrano@academicos.udg.mx
https://orcid.org/0009-0009-4370-3826

Luis Mexitli Orozco Torres
Universidad de Guadalajara, Ocotlán, Jalisco, México.
mexitli.orozco@academicos.udg.mx
https://orcid.org/0000-0001-6576-9767

Revista Iberoamericana para la Investigación y el Desarrollo Educativo
ISSN 2007 - 7467

Resumen

El lenguaje UML (Lenguaje Unificado de Modelado) es una poderosa herramienta que aporta importantes beneficios a los estudiantes de educación superior con la finalidad de que desarrollen un pensamiento computacional, adquirir una mayor habilidad para plantear y representar la solución de un proyecto o un problema de cualquier índole, ya sea informático, empresarial, industrial, educativo, entre otros, mediante diferentes tipos de diagramas que puedan ser interpretados por cualquier persona aun cuando no esté familiarizado con los aspectos técnicos de la computación.

UML es un estándar de modelado visual e internacional utilizado por los desarrolladores de sistemas de software para mantener una comunicación constante y efectiva con los actores involucrados (analistas, desarrolladores o usuarios finales). Con tales diagramas se plantean modelos de datos que les permiten alcanzar los objetivos planteados, cubrir los requisitos especificados y obtener una ventaja competitiva, en especial si se inicia con un proyecto enfocado en programas de software robustos y diseñados para soportar una enorme infraestructura, especialmente.

El presente trabajo plantea que los mismos principios informáticos pueden ser aplicados a una diversidad de problemas y escenarios, por lo que es conveniente que la habilidad del pensamiento computacional pueda desarrollarse en cualquier persona, con la finalidad de lograr una mayor facilidad para expresar la solución a una situación determinada.

Palabras clave: Lenguaje Unificado de Modelado (UML), pensamiento computacional, comunicación, estudiantes de educación superior.

Abstract

UML language (Unified Modeling Language) is a powerful tool that provides important benefits to higher education students to develop computational thinking and acquire a greater ability to propose and represent the solution to a project or a problem of any kind, whether computer science, business, industrial, or educational, among others, through different types of diagrams that can be interpreted by anyone even if they are not familiar with the technical aspects of computing.

UML is an international visual modeling standard used by software system developers to maintain constant and effective communication with the actors involved (analysts, developers, or end users). With such diagrams, data models are proposed that allow them to achieve the stated objectives, cover the specified requirements, and obtain a competitive advantage, especially if it begins with a project focused on robust software programs designed to support a huge infrastructure.

Revista Iberoamericana para la Investigación y el Desarrollo Educativo
ISSN 2007 - 7467

The present work proposes that the same computer principles can be applied to various problems and scenarios so that computational thinking can be developed in any person to achieve greater ease in expressing the solution to a situation determined.

Keywords: Unified Modeling Language (UML), computational thinking, communication, higher education students.

Resumo

A linguagem UML (Unified Modeling Language) é uma ferramenta poderosa que proporciona importantes benefícios aos estudantes do ensino superior com o objetivo de desenvolver o pensamento computacional, adquirindo maior capacidade de propor e representar a solução para um projeto ou um problema de qualquer natureza, seja computacional. ciência, empresarial, industrial, educacional, entre outros, por meio de diversos tipos de diagramas que podem ser interpretados por qualquer pessoa, mesmo que não esteja familiarizada com os aspectos técnicos da computação.

UML é um padrão internacional de modelagem visual utilizado por desenvolvedores de sistemas de software para manter uma comunicação constante e eficaz com os atores envolvidos (analistas, desenvolvedores ou usuários finais). Com tais diagramas, são propostos modelos de dados que permitem atingir os objetivos declarados, cobrir os requisitos especificados e obter uma vantagem competitiva, especialmente se começar com um projeto focado em programas de software robustos projetados para suportar uma enorme infraestrutura, especialmente.

O presente trabalho propõe que os mesmos princípios computacionais podem ser aplicados a uma variedade de problemas e cenários, por isso é conveniente que a habilidade de pensamento computacional possa ser desenvolvida em qualquer pessoa, a fim de obter maior facilidade em expressar a solução para um determinado situação.

Palavras-chave: Linguagem de Modelagem Unificada (UML), pensamento computacional, comunicação, estudantes de ensino superior.

Reception Date: June 2024 **Acceptance Date:** November 2024

Introduction

The use of UML as a tool for representation, interpretation, analysis, and development of computational thinking is essential in today's technological world in the face of the demand for solutions to global challenges due to its ability to visualize and communicate complex concepts, as well as to analyze and solve problems, making it an invaluable resource for future professionals.

This article describes some of the most relevant features of the UML language, as well as diagrams used in computing subjects that can serve as an example in practical situations for their use and understanding, highlighting their importance and the value they have in the context of computational thinking and finally a set of digital tools that have a great diversity of elements that support diagramming and/or collaborative work.

Nowadays, a large part of our environment is related to computer systems, and people want to communicate their ideas to others or propose a solution to a certain problem. To achieve this, one of the most used tools is graphic language since it allows one to state the result of the analysis of a process, avoid technicalities for the interlocutors, simplify complex concepts, and stimulate and strengthen the abstraction of ideas, among others.

As Zapata-Ros mentions, “It is a way of thinking that encourages the analysis and relationship of ideas for the organization and logical representation of procedures” (2015, p.1). Thus, this type of representation: graphic designs, diagrams, or models are very useful for bachelor's level students, providing them with skills that are crucial at various stages of life to achieve personal and professional success.

Computer scientists solve problems based on basic principles put forward by Seymour Papert in the 80s, who is considered the pioneer in the use of programming languages and the creator of the LOGO programming language. These principles are based on a technique to subdivide a problem, fragment the difficulties, or decompose a process into subprocesses, in other words: think of it as a whole, but understand it separately (Urcop, 2018). The objective of this work, as previously described, is to raise the usefulness of the UML language to represent, interpret, analyze, and develop computational thinking based on such principles.

This concept was proposed in 2006 by Jeannette Wing in her article Computational thinking, where he describes it as a fundamental competence for everyone and not only for computer scientists, to function effectively in the knowledge society. Developing this skill involves problem-solving, system design, and understanding behaviors based on the fundamental concepts of computer science. Wing believed that

regardless of whether it was for reading, writing, or arithmetic, this skill should be added to each child's analytical capacity (Wing, 2006, pp.33-35).

Thus, this form of discernment approaches projects or problems with a critical perspective, taking into account notions of computing and digital technologies in a complementary way to other reasoning that makes it possible to visualize a problem in smaller parts that are easy to handle and understand, recognize patterns, make abstractions and develop algorithms for its resolution. The practice of these actions contributes to greater confidence by fragmenting complexity, increasing persistence in work and the search for solutions, as well as tolerating and resolving ambiguous situations (ISTE and CSTA, as cited in How I Learn Code, 2018).

UML is defined as a standard language for specifying, constructing, visualizing, and documenting the components of a software system. It was created in 1994 by the software development company *Rational to br*ing together previous design methods, and its main references are Grady Booch, James Rambaugh, and Ivar Jacobson. They were looking for a way to unify and represent the proposed model to solve a problem related to software development using a graphical notation. UML 1.0 was finally presented to the OMG (Object-Management Group) in 1997 through a consortium called UML Partners (Booch et al., 2006, pp.15-18).

Since then, there have been different revisions, but it continues to be widely applied to various fields of software development (Jiménez, 2021, p.20). Additionally, it is considered a powerful tool because it provides a variety of visual and conceptual elements that stimulate computational thinking in young university students.

Materials and methods

It is important to mention that some UML diagrams may be more useful than others at a certain time due to their characteristics; however, their essential purpose is to create a model to simplify reality, offering a view from different perspectives (Booch et al., 2006, p.50).

The types of UML diagrams proposed by Booch et al. (2006, p.50) are:

1. Class diagram.
2. Object diagram.
3. Component diagram.
4. Composite structure diagram.
5. Use case diagram.
6. State diagram.

7. Activity diagram.
8. Deployment diagram.
9. Package diagram.
10. Sequence diagram.
11. Timing diagram.
12. Communication diagram.
13. Global interaction diagram.

As can be seen, this wide variety of diagrams allows for the schematization of the most diverse situations. Below are some examples of diagrams that can be used, which can be adjusted depending on more specific cases.

In the curricula of computer science areas, subjects are included that involve the management of information through technologies that implement the use of databases or the design and development of video games, to name a few. Specialists in such areas of knowledge must ensure that a clear understanding of the nature of the data is obtained and how it is used and have a communication model that is non-technical and free of ambiguities (Connolly & Begg, 2005).

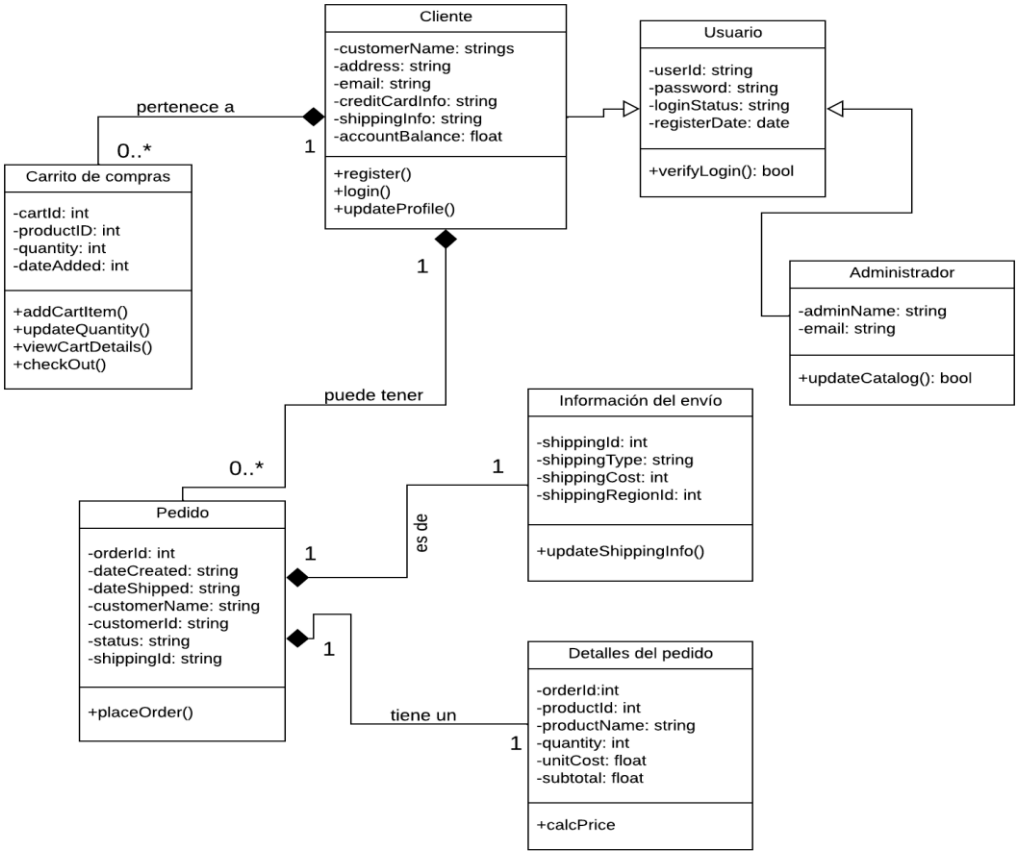
Thus, specifically, UML class diagrams are especially useful for representing the data included in the IT solution to design rich entity-relationship (ER) models, emphasizing their advantages, as mentioned by Booch et al.

> UML class diagrams are a superset of entity-relationship (ER) diagrams, a widely used modeling tool for logical database design. While classical ER diagrams focus only on data, class diagrams go a step further by allowing behavioral modeling. In the physical database, these logical operations typically become triggers or stored procedures (2006, p.140-141).

Figure 1 shows a logical design that meets the information needs of a database system applied to an online shopping cart. It shows various classes, their relationships, and cardinality.

Each class is represented by a box divided into three sections: the top section corresponds to the name of the class, the middle section defines the attributes and their scope, the minus sign (-) is used to define a private attribute, while a plus sign (+) is used to define a public attribute. In the lower section, the methods or actions that the objects of that class can perform are declared.

Figure 1. Example of a Class Diagram.



Source: Own elaboration with Lucidchart

As an example, it can be explained that the class called User is composed of four private attributes (userId, password, loginStatus, registerDate) and a public method called *verifyLogin ( ),* which returns a true or false (boolean) value after its execution. Two more classes are inherited from this class: the Client class and the Administrator class. In other words, these classes are subclasses of the User class or special types of users.

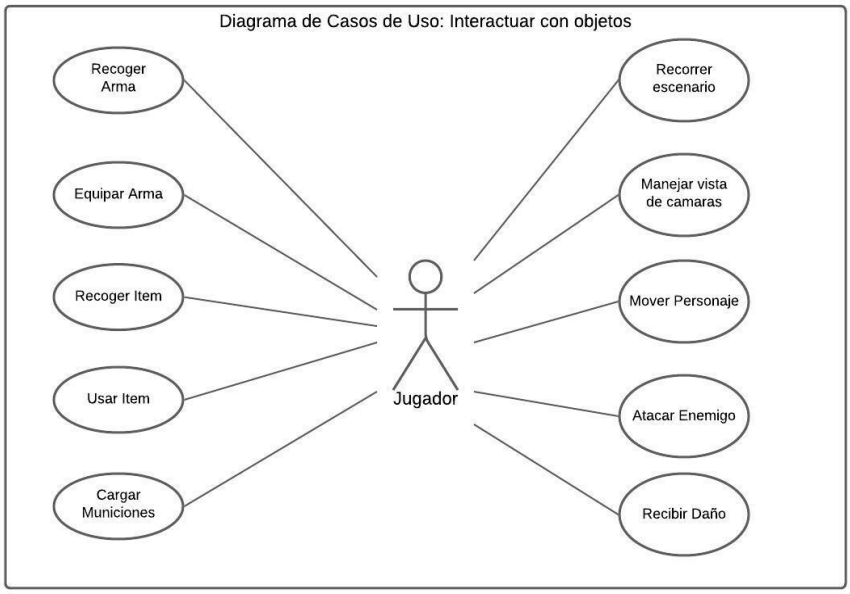
Furthermore, in the context of video game development, UML diagrams play an essential role in understanding, organizing, and visualizing the key elements of the game as a whole.

Among the most common diagrams is the class diagram, which represents the fundamental components that shape the structure of the system. Deployment diagrams can also be used, which allow the physical relationships between nodes to be visualized, as well as the distribution of components within them.

Other relevant diagrams are the use case and activity diagrams. Both are used to specify the detailed design of the game elements and how they interact with each other (see Figure 2). While the use case diagram helps to understand the different situations and actions that can occur, another type of diagram, such as the activity diagram, allows to

visualize the flows and sequences of actions that the elements carry out during their development, including the exchange of messages between them (Fernández et al., 2014).

Figure 2. Use a case diagram of a video game. Taken from the thesis "Development of a three-dimensional video game."



Source: (Macas & Mena V., 2008)

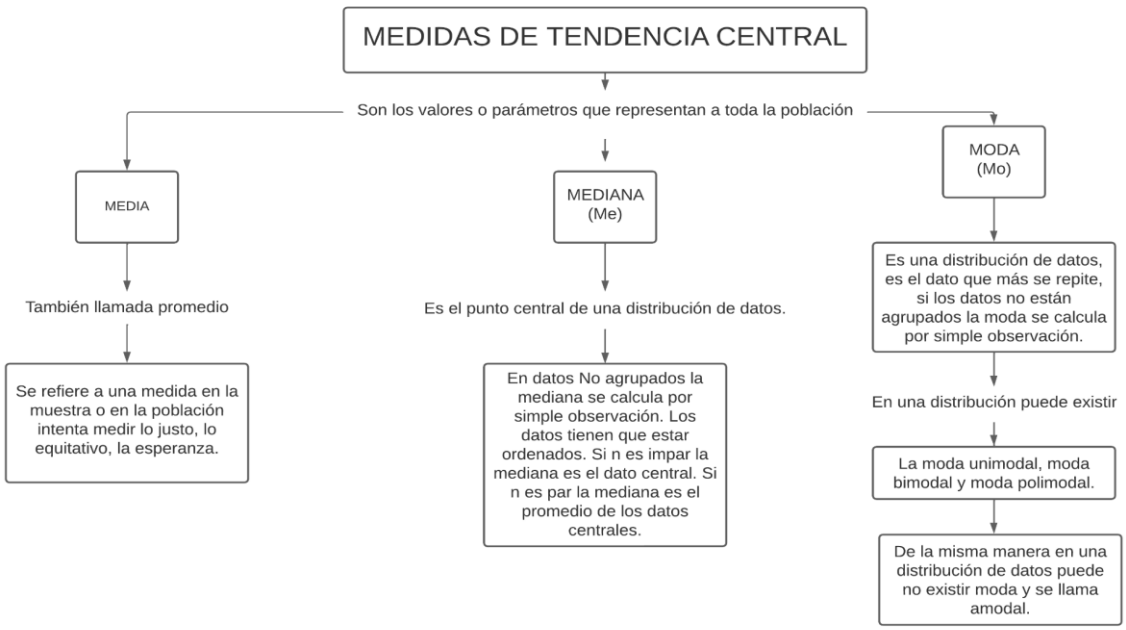
Source: CD-1417 (epn.edu.ec) .

Together, these UML diagrams provide a complete and structured view of the game's design and functionality, providing a valuable communication tool between game developers and designers, allowing them to work more effectively and achieve a successful final product.

To some extent, in subjects such as statistics, students need to learn basic concepts of measures of central tendency to mention a specific topic.

To present this area of knowledge to students, one could start with a conceptual map, as shown in Figure 3, which is especially useful for visualizing the important concepts and definitions involved in the most representative types in a simple and synthesized way.

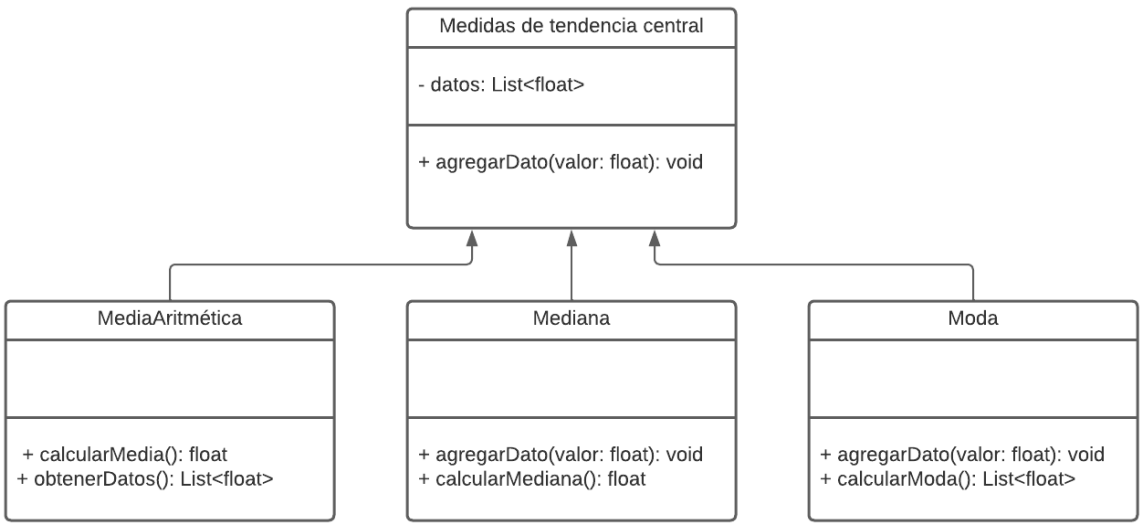
Figure 3. Conceptual diagram of measures of central tendency.



Source: Own elaboration.

These concepts are then translated into a UML class diagram, where the superclass Central Tendency Measures are presented and the Arithmetic Mean, Median, and Mode as subclasses since these share or inherit characteristics from the main class (Measures of central tendency) and its different methods (see figure 4).

Figure 4. Class diagram for calculating measures of central tendency.



Source: Own elaboration.

Similarly, the aim is to abstract from the real world the data needed to solve a problem where the application of statistics provides important information for decision-making. Thus, the student will learn to follow a series of steps depending on the selected

central tendency measure, differentiating the treatment and formula applied. With these steps, in a simplified manner, the student develops a skill applicable not only to this area of knowledge but to any other.

These cases are just examples of how computer science principles can be translated and applied in problem-solving and how UML diagrams provide a more complete and clearer view of the proposed solutions or possible functionalities of certain situations, where the integration between the conceptual and visual parts of an idea can be seen in a more simplified way.

This paper analyzes the most commonly used software programs for creating UML diagrams. Although the features contain several useful tools for software developers, the goal is for any user to be able to identify those that best suit their needs. Various criteria were considered, such as ease of use, flexibility, and type of license, among others.

Some of them are described below:

MagicDraw: This is a handy UML editor with a friendly and customizable user interface, supports UML and CASE (Computer Aided Software Engineering) with teamwork support, which facilitates the analysis and design of object-oriented systems and databases. Among its main characteristics are that it can run on different operating systems and create the design of real-time, client/server, and n-tier distributed applications (ElSoft, 2021).

Lucidchart: Lucidchart is an online UML tool for teamwork; its diagramming is useful for making better decisions together, as well as optimizing ideas, projects, and processes. “One of the advantages of Lucidchart is its intuitive operation, as it allows you to simultaneously share and edit diagrams as a team and integrate comments directly into the tool” (Ionos Inc., 2021).

As described by Ionos Inc. (2021), with the free account, you can access seven UML diagram templates:

1. Class diagram
2. State diagram
3. Activity diagram
4. Sequential diagram
5. Component diagram
6. Use case diagram
7. Deployment diagram

One of the important features is that it has the most useful diagrams for database design since it is capable of importing its structure directly from the Database Management System (DBMS) to Lucidchart and is compatible with the most popular platforms such as MySQL, Oracle, PostgreSQL and SQL Server (Lucid Software Inc., 2023).

IBM Engineering Rhapsody Model Manager: formerly called Rational Rhapsody Model Manager, is a graphical integrated development environment (IDE) for systems engineers and software developers who create real-time and embedded systems and software. This UML modeling tool covers object-based software development for web applications, as well as embedded systems and real-time systems based on C++ and Java EE. UML/ SysML modeling allows you to quickly create source code in the IDE for the specified languages, C, C#, and Ada (Smarter Process, n.d.).

You can also draw models freely, import existing code to view it as a diagram, or download requirements from a management tool. It also supports all current UML 2 diagram types, but its disadvantage is that it is relatively expensive to use (Smarter Process, n.d.).

Gliffy is a cloud-based application that makes it easy to visualize software projects with UML diagrams, features built-in templates, and allows you to create all kinds of graphics such as flowcharts, Venn diagrams, UML diagrams, organizational charts, entity-relationship diagrams, wireframes, floor plans, and mind maps, as well as business process models and notations (BPMN) among other things. The online tool was written in HTML5 and is ideal for visual collaboration and teamwork as read-only versions can be shared, work in parallel, and exchange information thanks to the integrated comment function (Digital Guide IONOS, 2021).

Gliffy has a free UML version that imports diagrams in GON, Gliffy, and gXML formats. However, other formats require a paid account. Certain image formats such as JPG, PNG, and SVG can also be exported, and a Google Drive connection can be made. The free account also provides 2 MB of cloud space for saving graphics, and it is possible to share up to five models at the same time. Of course, the paid account offers further possibilities, such as unlimited import of memory or diagrams from Visio and export to the image formats mentioned above (Digital Guide IONOS, 2021).

Draw.io: Is a free diagramming tool that allows you to draw any type of mind maps, concept maps, diagrams, or different graphical representations, such as hierarchy or set diagrams, and enrich them with images, icons, and links; in addition to collaborating and cooperating with others thanks to the simplicity of sharing them. It has enterprise-

level privacy and security standards, as well as advanced and useful tools that are integrated into the diagram editor to facilitate designs (INTEF, n.d.), and is available on the site https://app.diagrams.net/ .

According to Jorge Vázquez Fernando (as cited in INTEF, n.d.), its main advantages at an educational level are:

- Integration: This tool can be used in online or desktop versions. For online versions, you can link to different accounts and save your work in cloud storage systems (Google Drive, OneDrive, Dropbox) or download it, and you can also import diagrams from other applications.
- Templates and libraries: a series of templates organized by type are available to make the work easier and faster. It consists of a series of libraries with a large number of shapes, lines, drawings, and icons, which can be expanded. And if these options are not enough, you can import or create the ones you need.
- Ease of use: It has basic tools to develop the experience, it uses the drag and drop system, with which you can quickly bring shapes, arrows, images, etc., to the work area. Another important feature of the application is that the work area expands as the diagram, representation, or drawing grows without the need to include pages. It also includes automatic saving, so work is never lost due to unexpected closures. And finally, something that may be attractive to students is the ease of exporting work.
- Sharing and cooperating: It is possible to create files that can be edited by several people at the same time. In this way, students are encouraged to use collaborative environments and cooperative methodologies. Finally, when the product is finished, they can publish or share the file through a simple link (INTEF, n.d.).

Given these characteristics, it can be concluded that Draw.io is a simple and useful tool with which students can improve teacher resources for the classroom that require diagrams or a structured organization. In this sense, results are obtained with significant time savings since its integration with cloud platforms makes the work simple and worry-free (INTEF, n.d.).

Results

The use of UML as a tool for the representation, interpretation, and analysis of information has proven to be an essential element, especially in those fields related to engineering or even integrating into any area, promoting and strengthening computational thinking. As established in the main objective of this study, the key characteristics of the UML language were explored, where it has been possible to observe that the versatility of the diagrams that comprise it provides a significant advantage both in the understanding and in the solution of complex problems, by providing through a graphical language the understanding and solutions to said problems, facilitating their understanding in the face of various problems.

Through the results obtained in these studies and the agreement on the benefits of UML, it is evident that students can benefit from its use in multidisciplinary areas, thus demonstrating the great versatility that it can have as a modeling tool, breaking down disciplinary barriers.

Discussions

Throughout this study, different UML diagrams and their applications in different contexts were analyzed. Among the main observations in the implementation of UML, it is possible to highlight that it allows students to develop a greater capacity to conceptualize problems and generate solutions. A clear example is the use of case diagrams that can provide a visual structure that helps students understand and communicate their ideas more effectively, even if they are complex. By integrating this type of diagram, it can help with their classification. This is consistent with what was proposed by Zapata-Ros (2015), who argues that graphical visualization facilitates the organization and logical representation of procedures, a crucial aspect in the development of computational thinking. On the other hand, Páez et al. (2021) raise the effectiveness and efficiency of using UML in the management of multidisciplinary projects. Their study addresses the creation of software for controlling the entry/exit of student bicycles at the Universidad Cooperativa de Colombia, where they emphasize the achievement of their results thanks to the application of UML and SCRUM that allowed them to obtain efficiency in the development of said project.

Furthermore, Juárez et al. (2020) in their study, mention that the use of UML in education improves the efficiency and quality in the delivery of projects, highlighting that technology and video game students who use UML show a greater ability to conceptualize and solve problems in various technological areas. Pérez-Castillo et al. (2021) also

contributes with their study focused on the representation of quantum circuits through activity diagrams that mention an ease in the integration of classical and quantum components in hybrid information systems.

On the other hand, there are also certain limitations in the use of UML, including the complexity of some diagrams, which can represent challenges for students who are not familiar with the concepts of graphical modeling. In addition, the adaptation of UML to specific problems may require a deep understanding of the use of the various types of diagrams and their applications, which implies another of the limitations, which refers to the learning curve regarding the use of the different tools, for which time must be invested to become familiar and know the functionalities in detail.

Conclusions

The study has confirmed that UML is a valuable tool for the development of computational thinking and the representation of solutions in higher education. Its implementation not only improves students' ability to conceptualize and communicate ideas but also provides a solid foundation for effective collaboration and problem-solving in various fields that students may face as challenges in the current and future digital world.

Considering that collaboration between disciplines has become an essential component regarding the resolution of complex problems in the academic and professional fields, UML emerges as a tool that facilitates communication and understanding between different areas of knowledge, promoting synergy between different specialties, highlighting the breaking of linguistic and disciplinary barriers, facilitating communication and integration of perspectives and strengthening not only efficiency by integrating it into multidisciplinary projects but also promoting the culture of collaboration.

Future lines of research

As future contributions to maximize the potential that UML can offer in various contexts, including technical, educational, teaching-learning, IT project modeling, and industrial, among others, the following lines of research are proposed:

Promoting the development of computational thinking: implementing a structured and detailed approach in students to develop systematic and analytical thinking when facing complex problems as a learning strategy to address abstract concepts, preparing them to join the labor market with directly applicable and in-demand skills.

Effectiveness of software design learning: assess students' understanding and mastery of software design principles, methodologies, and implementation through comparative studies between groups, contrasting results in terms of design quality and efficiency.

Impact on collaboration and communication: Refer to how, by using UML, students learn to communicate design ideas effectively through diagrams and models, where these skills are promoted, which are currently crucial among multidisciplinary teams, measuring the precision and clarity in the transmission of design ideas.

Finally, applying UML in higher education can enrich the learning experience of students by providing them with practical and theoretical skills that enable them to face various challenges in their professional development.

It is important to emphasize that the lines presented above are not exclusive, but due to the nature of UML, it can be widely applied to approaches as required by other researchers.

References

Booch, G., Rumbaugh, J. y Jacobson, I. (2006). *El lenguaje unificado de modelado. Guía del usuario*. Pearson Educación.

Connolly, T. M. y Begg, C. E. (2005). *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*. Pearson Educación.

Digital Guide IONOS. (1 de junio de 2021). *Las 6 mejores herramientas UML*. IONOS. https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/las-mejores-herramientas-uml/

Fernández, F., Rodillo, D., del Pino, F., Bajo, J. y Corchado, J. M. (2014). WirePET: Desarrollando videojuegos para dispositivos móviles con comunicación bluetooth.https://www.researchgate.net/profile/Javier-Bajo-2/publication/228646208_WirePET_Desarrollando_Videojuegos_para_Dispositivos_Moviles_con_Comunicacion_Bluetooth/links/02bfe50c8ba48a350c000000/WirePET-Desarrollando-Videojuegos-para-Dispositivos-Moviles-con-C

Gliffy. (s.f.). *Build Better Applications with Software Engineering Diagrams*. Gliffy. https://www.gliffy.com/solutions/diagrams-for-software-engineering

How I Learn Code. (22 de marzo de 2018). *Jeannette M. Wing y el Pensamiento Computacional*. How I Learned Code. https://howilearnedcode.com/2016/10/el-pensamiento-computacional-jeannette-m-wing/

INTEF (s.f.). *Draw.io: mucho más que mapas mentales*. INTEF. https://intef.es/observatorio_tecno/draw-io-mucho-mas-que-mapas-mentales/

Ionos Inc. (1 de junio de 2021). *Las 6 mejores herramientas UML*. IONOS. https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/las-mejores-herramientas-uml/

Jiménez, C. (2021). *UML. Arquitectura de aplicaciones en Java, C++ y Python. 2ª Edición*. RA-MA S.A. Editorial y Publicaciones.

Lucid Software Inc. (2023). *Herramienta de diseño de base de datos - Diagrama esquemático en línea*. Lucidchart. https://www.lucidchart.com/pages/es/ejemplos/herramienta-de-diseno-de-bases-de-datos

Macas, J. P. y Mena V., A. M. (2008). *Desarrollo de un videojuego tridimensional* [Tesis de Licenciatura, Escuela Politécnica Nacional] [Digital]. bibdigital. Repositorio Institucional https://bibdigital.epn.edu.ec/bitstream/15000/956/1/CD-1417.pdf

Páez, J. A., Cortes, J. A., Simanca, F. A. y Blanco, F. (2021). Aplicación de UML y SCRUM al desarrollo del software sobre control de acceso. Información Tecnológica, 32(5), 57-66. https://doi.org/10.4067/s0718-07642021000500057

Smarter Process. (s.f.). *IBM Engineering Rhapsody Model Manager (IBM RMM)*. SmarterProcess. https://doorsnext.com/products/ibm-engineering-rhapsody-model-manager/

URCOP. (2018, 12 de diciembre). *Pensamiento computacional* [video]. YouTube. https://youtu.be/veXgaxaNICM

Wing, J. M. (2006). *Computational Thinking*. CMU School of Computer Science. https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf

Zapata-Ros, M. (23 de octubre de 2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia (RED)*. Revistas UM. https://revistas.um.es/red/article/view/240321

Contribution Role	Author(s)
Conceptualization	Maria del Rocio Ramirez Jimenez. <<same>> Karime Pulido Hernandez <<same>> Carmen Elizabeth Rivera Orozco <<same>> Liliana Serrano Zuñiga . <<same>> Luis Mexitli Orozco Torres. <<same>>
Methodology	Maria del Rocio Ramirez Jimenez. <<same>> Karime Pulido Hernandez <<same>> Norma Alicia Gomez Torres <<same>> Liliana Serrano Zuñiga . <<same>>
Investigation	Maria del Rocio Ramirez Jimenez. <<same>> Karime Pulido Hernandez <<same>> Norma Alicia Gomez Torres <<same>> Liliana Serrano Zuñiga . <<same>>
Writing - Preparing the original draft	Maria del Rocio Ramirez Jimenez. <<same>> Karime Pulido Hernandez <<same>> Norma Alicia Gomez Torres <<same>> Liliana Serrano Zuñiga . <<same>>
Writing - Review and editing	Maria del Rocio Ramirez Jimenez. <<same>> Carmen Elizabeth Rivera Orozco <<same>> Luis Mexitli Orozco Torres. <<same>>
Display	Maria del Rocio Ramirez Jimenez. <<same>> Carmen Elizabeth Rivera Orozco <<same>> Luis Mexitli Orozco Torres. <<same>>
Supervision	Maria del Rocio Ramirez Jimenez. <<same>> Luis Mexitli Orozco Torres. <<same>>
Project Management	Luis Mexitli Orozco Torres. <<main>>
Acquisition of funds	Luis Mexitli Orozco Torres. <<main>>