

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

Artículos científicos

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, México

mdelrocio.ramirez@academicos.udg.mx

<https://orcid.org/0000-0003-1709-5539>

Karime Pulido Hernández

Universidad de Guadalajara, México

karime.pulido@academicos.udg.mx

<https://orcid.org/0009-0005-5672-7510>

Carmen Elizabeth Rivera Orozco

Universidad de Guadalajara, México

carmen.rivera@academicos.udg.mx

<https://orcid.org/0000-0001-6588-6779>

Norma Alicia Gómez Torres

Universidad de Guadalajara, México

norma.gtorres@academicos.udg.mx

<https://orcid.org/0009-0004-0808-4111>

Liliana Serrano Zúñiga

Universidad de Guadalajara, México

liliana.serrano@academicos.udg.mx

<https://orcid.org/0009-0009-4370-3826>

Luis Mexitli Orozco Torres

Universidad de Guadalajara, México

mexitli.orozco@academicos.udg.mx

<https://orcid.org/0000-0001-6576-9767>

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, especially.



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.

Fecha Recepción: Junio 2024

Fecha Aceptación: Noviembre 2024

Introducción

El uso de UML como una herramienta de representación, interpretación, análisis y desarrollo del pensamiento computacional, es esencial en el mundo tecnológico actual ante la demanda de solución a desafíos globales por su capacidad para visualizar y comunicar conceptos complejos, así como para analizar y resolver problemas, convirtiéndolo en un recurso invaluable para los futuros profesionistas.

En este artículo se describen algunas de las características más relevantes del lenguaje UML, así mismo, diagramas utilizados en asignaturas del área de computación que pueden servir como ejemplo en situaciones prácticas para su uso y comprensión, destacando su importancia y el valor que tienen en el contexto del pensamiento computacional y finalmente un conjunto de herramientas digitales que disponen de una gran diversidad de elementos que apoyan en la diagramación y/o trabajo colaborativo.

En la actualidad, gran parte de nuestro entorno está relacionado con sistemas informáticos y las personas desean comunicar sus ideas a otros, o bien, plantear una solución a determinado problema. Para lograr transmitirlos, una de las herramientas más utilizadas es el lenguaje gráfico, ya que permite plasmar el resultado del análisis de algún proceso, evitar tecnicismos a los interlocutores, simplificar conceptos complejos, así como, estimular y fortalecer la abstracción de las ideas, entre otros.

Como menciona Zapata-Ros “Se trata de una forma de pensar que propicia el análisis y la relación de ideas para la organización y la representación lógica de procedimientos” (2015, p.1). Así pues, este tipo de representaciones: diseños gráficos, diagramas o modelos resultan de gran utilidad para los alumnos de nivel Licenciatura, dotándolos de habilidades que son determinantes en diversas etapas de la vida para alcanzar un éxito personal y profesional.

Los informáticos resuelven problemas basándose en principios básicos planteados por Seymour Papert en los 80's, quien es considerado el pionero en el uso de lenguajes de programación y creador del lenguaje de programación LOGO. Estos principios se basan en una técnica para subdividir un problema, fragmentar las dificultades o descomponer un proceso en subprocesos, dicho de otra manera: pensarlo como un todo, pero, comprenderlo de forma separada (Urcop, 2018). El objetivo de este trabajo como se describió con anterioridad es plantear la utilidad del lenguaje UML para representar, interpretar, analizar y desarrollar el pensamiento computacional basándose en tales principios.

Dicho concepto fue planteado en 2006 por Jeannette Wing en su artículo Computational thinking, donde lo describe como una competencia fundamental para

todos y no únicamente para los científicos de computación, a fin de desenvolverse de manera efectiva en la sociedad del conocimiento. Desarrollar esta habilidad implica la resolución de problemas, el diseño de sistemas y la comprensión de los comportamientos, basándose en los conceptos fundamentales de la informática. Wing consideraba que independientemente si fuese para lectura, escritura y la aritmética, se debería añadir esta habilidad a la capacidad analítica de cada niño (Wing, 2006, pp.33-35).

Así pues, esta forma de discernimiento aborda los proyectos o problemas con una perspectiva crítica, teniendo en cuenta nociones de computación y tecnologías digitales de manera complementaria a otros razonamientos que posibilitan visualizar un problema en partes más pequeñas que sean fáciles de manejar y comprender, reconocer patrones, realizar abstracciones y desarrollar algoritmos para su resolución. La práctica de estas acciones contribuye a que haya una mejor confianza al fragmentar la complejidad, aumentando la persistencia en el trabajo y búsqueda de soluciones, así como tolerar y resolver situaciones ambiguas (ISTE y CSTA, como se citó How I Learn Code, 2018).

UML se define como un lenguaje estándar para especificar, construir, visualizar y documentar los componentes de un sistema de software. Surge en el año 1994 por la empresa de desarrollo de software *Rational* con el propósito de reunir métodos de diseño antecesores; y son Grady Booch, James Rumbaugh e Ivar Jacobson sus principales referentes. Ellos buscaban una manera de unificar y representar el modelo planteado para resolver algún problema relacionado con el desarrollo de software utilizando una notación gráfica. UML 1.0 finalmente fue presentado a la OMG (Object-Management Group) en 1997 a través de un consorcio llamado UML Partners (Booch et al., 2006, pp.15-18).

Desde entonces han existido distintas revisiones, pero sigue aplicándose de manera extensa a diversos campos de desarrollo de software (Jiménez, 2021, p.20), adicionalmente, es considerada una herramienta poderosa debido a que proporciona una variedad de elementos visuales y de tipo conceptual que estimulan el pensamiento computacional en los jóvenes universitarios.

Materiales y Métodos

Es importante mencionar que algunos de los diagramas de UML pueden resultar más útiles que otros en determinado momento debido a sus características; sin embargo, tienen como propósito esencial crear un modelo para simplificar la realidad, ofreciendo una visión desde distintas perspectivas (Booch et al., 2006, p.50).

Los tipos de diagramas UML planteados por Booch et al. (2006, p.50) son:

1. Diagrama de clases.
2. Diagrama de objetos.



3. Diagrama de componentes.
4. Diagrama de estructura compuesta.
5. Diagrama de casos de uso.
6. Diagrama de estados.
7. Diagrama de actividades.
8. Diagrama de despliegue.
9. Diagrama de paquetes.
10. Diagrama de secuencia.
11. Diagrama de tiempos.
12. Diagrama de comunicación.
13. Diagrama global de interacciones.

Como puede observarse, esta amplia variedad de diagramas permite esquematizar las más diversas situaciones. A continuación, se presentan algunos ejemplos de diagramas que pueden emplearse, mismos que pueden ajustarse dependiendo de casos más específicos.

En los planes de estudio de áreas computacionales, se incluyen asignaturas que involucran el manejo de información mediante tecnologías que implementan el uso de bases de datos BD o el diseño y desarrollo de videojuegos por mencionar algunas. Los especialistas en tales áreas de conocimiento deben garantizar que se obtenga una comprensión clara de la naturaleza de los datos, el modo en que éstos se utilizan, y disponer de algún modelo de comunicación que sea no técnico y que esté libre de ambigüedades (Connolly & Begg, 2005).

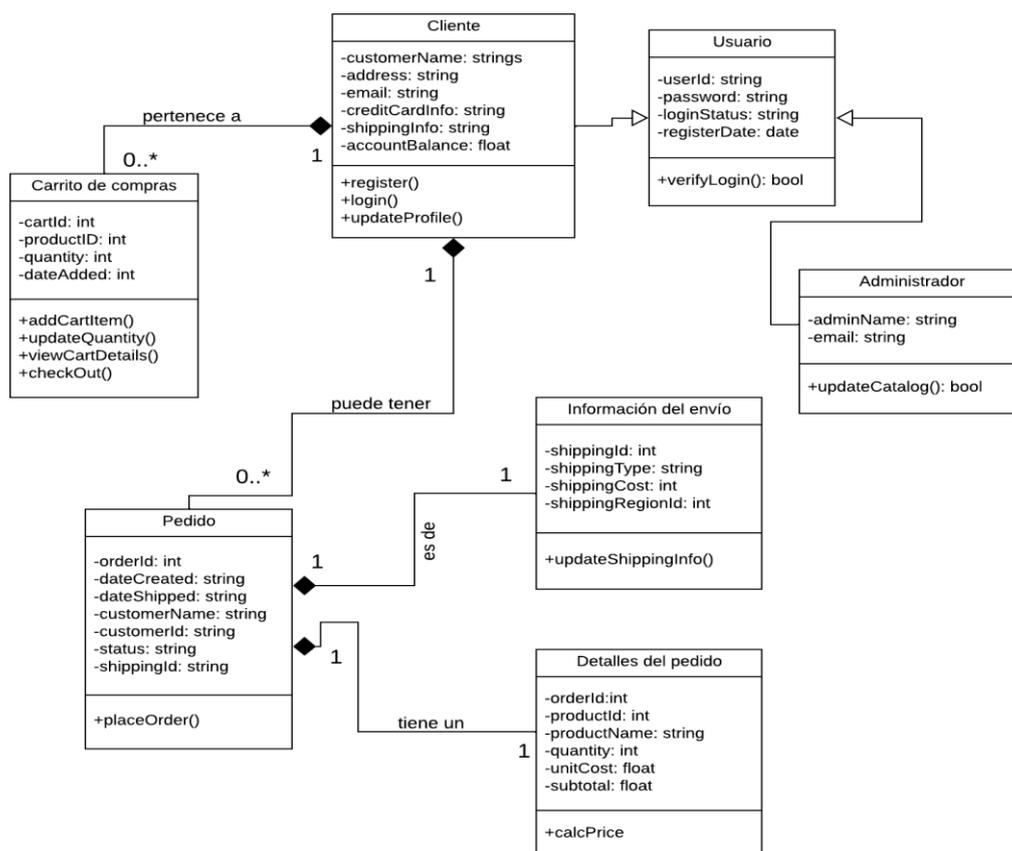
De este modo, de manera específica los diagramas de clases de UML resultan especialmente útiles para representar los datos comprendidos en la solución informática para diseñar los modelos entidad-relación (E-R) enriquecidos, haciendo énfasis en sus ventajas como lo menciona Booch et al.

Los diagramas de clases de UML son un superconjunto de los diagramas entidad-relación (E-R), una herramienta de modelado para el diseño lógico de bases de datos utilizada con mucha frecuencia. Mientras que los diagramas de E-R clásicos se centran solo en los datos, los diagramas de clases van un poco más allá pues permiten el modelado del comportamiento. En la base de datos física, estas operaciones lógicas normalmente se convierten en disparadores (triggers) o procedimientos almacenados (2006, p.140-141).

La Figura 1 muestra un diseño lógico que satisface las necesidades de información de un sistema de bases de datos aplicado a un carrito de compras en línea. En él se pueden apreciar diversas clases, sus relaciones y la cardinalidad.

Cada clase se representa con un recuadro dividido en 3 secciones, la parte superior corresponde al nombre de la clase, la intermedia define los atributos y su alcance, se usa el signo de menos (-) para definir un atributo privado mientras que se usa un signo de más (+) para definir un atributo público. En la sección inferior se declaran los métodos o acciones que los objetos de dicha clase podrán realizar.

Figura 1. Ejemplo de un Diagrama de clases.



Fuente: Elaboración propia con Lucidchart

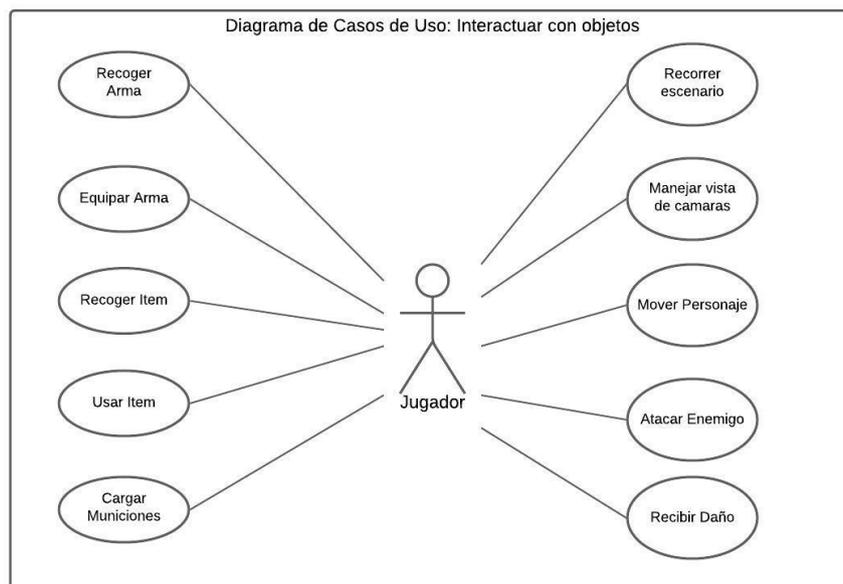
A manera de ejemplo, se puede explicar que la clase llamada Usuario está compuesta de cuatro atributos privados (userId, password, loginStatus, registerDate) y un método público llamado *verifyLogin()* que devuelve un valor de verdadero o falso (booleano) tras su ejecución. De esta clase se heredan dos más: la clase Cliente y la clase Administrador, dicho de otra manera, estas clases son subclases de la clase usuario o tipos especiales de usuarios.

Por otro lado, en el contexto del desarrollo de videojuegos, los diagramas de UML desempeñan un papel esencial para comprender, organizar y visualizar los elementos clave del juego en conjunto.

Entre los diagramas más comunes destaca el diagrama de clases, que representa los componentes fundamentales que dan forma a la estructura del mismo. También podrían utilizarse los diagramas de despliegue, que permiten visualizar las relaciones físicas entre los nodos, así como la distribución de los componentes en ellos.

Otros diagramas relevantes son los de casos de uso y de actividad. Ambos se emplean para especificar el diseño detallado de los elementos del juego y cómo interactúan entre sí (ver figura 2). Mientras el diagrama de casos de uso ayuda a entender las distintas situaciones y acciones que pueden darse; otro tipo de diagramas como lo es el de actividad permite visualizar los flujos y secuencias de acciones que los elementos realizan durante su desarrollo, incluyendo el intercambio de mensajes entre ellos (Fernández et al., 2014).

Figura 2. Diagrama de casos de uso de un videojuego. Retomado de la tesis “Desarrollo de un videojuego tridimensional”.



Fuente: (Macas & Mena V., 2008)

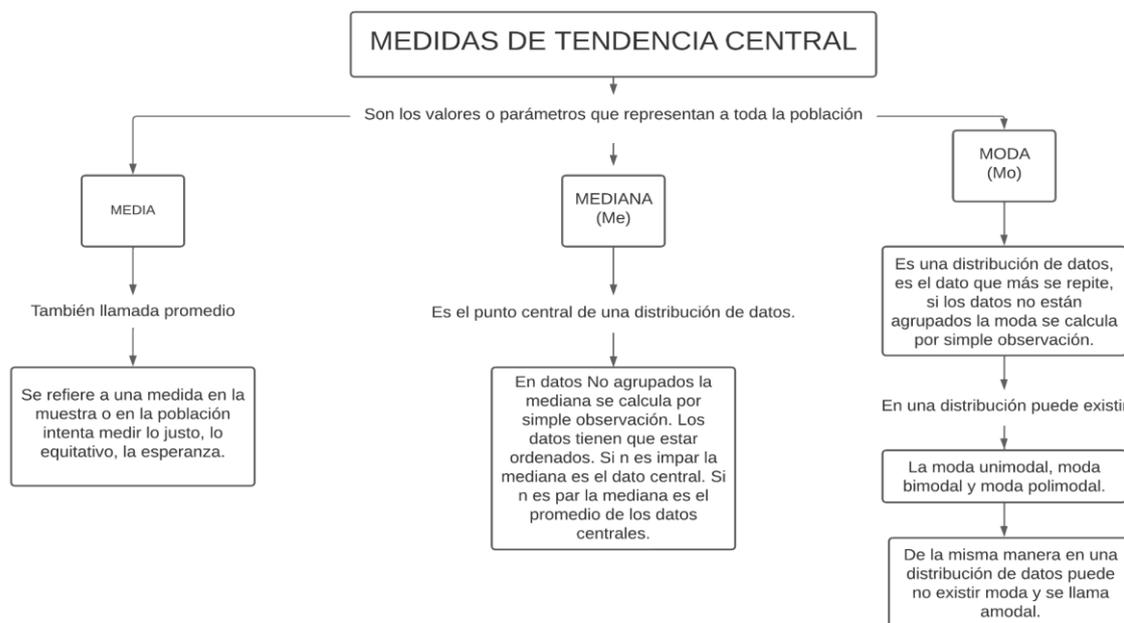
Fuente: CD-1417 (epn.edu.ec).

En conjunto, estos diagramas de UML ofrecen una visión completa y estructurada del diseño y la funcionalidad del videojuego, los cuales proporcionan una valiosa herramienta de comunicación entre los desarrolladores y diseñadores del juego, permitiéndoles trabajar de manera más efectiva y lograr un producto final exitoso.

Por otro lado, en asignaturas como la estadística, es importante que los alumnos aprendan conceptos básicos de medidas de tendencia central por mencionar un tema en específico.

Para presentar esta área del conocimiento a los estudiantes, se podría iniciar con un mapa conceptual como se muestra en la figura 3, especialmente útil para visualizar los conceptos y definiciones importantes involucradas de los tipos más representativos de una manera sencilla y sintetizada.

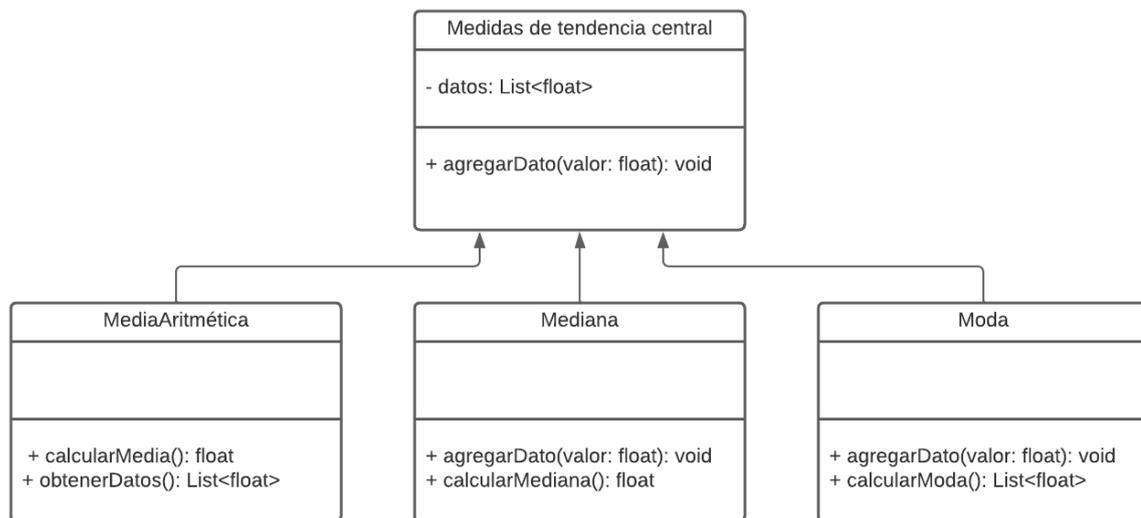
Figura 3. Diagrama conceptual de medidas de tendencia central.



Fuente: Elaboración propia.

Posteriormente, estos conceptos se traducen a diagrama de clases de UML, donde se presenta la superclase Medidas de tendencia central y como subclasses la Media Aritmética, Mediana y Moda, ya que éstas comparten o heredan características de la clase principal (Medidas de tendencia central) y sus distintos métodos ver figura 4.

Figura 4. Diagrama de clases para el cálculo de medidas de tendencia central.



Fuente: Elaboración propia.

De igual manera, se pretende abstraer del mundo real los datos necesarios para resolver algún problema donde la aplicación de la estadística proporciona información importante para la toma de decisiones. Así pues, el estudiante aprenderá a seguir una serie de pasos según sea la medida de tendencia central seleccionada, diferenciando el tratamiento y fórmula aplicada. Con estos pasos, de manera simplificada se desarrolla en el estudiante una habilidad aplicable no sólo a esta área del conocimiento sino a cualquier otra.

Estos casos, son solo ejemplos de cómo pueden traducirse y aplicarse los principios informáticos en la resolución de problemas y la forma en que los diagramas UML aportan una visión más completa y clara de las soluciones planteadas o posibles funcionalidades de determinadas situaciones, donde la integración entre la parte conceptual y visual de una idea, pueden verse plasmadas de manera más simplificada.

En este trabajo se efectuó un análisis de los programas de software más utilizados para la creación de diagramas UML; y aun cuando las características contienen en mayor medida una serie de herramientas útiles para los desarrolladores de software, el objetivo es que cualquier usuario sea capaz de identificar aquellos que mejor se adecúen a sus necesidades. Se consideraron diversos criterios, tales como la facilidad de uso, flexibilidad, tipo de licencia, entre otras.

A continuación, se describen algunos de ellos:

MagicDraw: es un práctico editor de UML con una interfaz de usuario amigable y personalizable, admite el modelado UML y CASE (Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora) con soporte de trabajo

en equipo, lo que facilita el análisis y el diseño de sistemas orientados a objetos y bases de datos. Entre sus principales características están que puede ejecutarse sobre distintos sistemas operativos y crear el diseño de aplicaciones en tiempo real, cliente/servidor y distribuidas de n-capas (ElSoft, 2021).

Lucidchart: es una herramienta UML en línea para el trabajo en equipo; su diagramación es útil para que se tomen mejores decisiones en conjunto, así como optimizar ideas, proyectos y procesos. “Una de las ventajas de Lucidchart es su funcionamiento intuitivo, ya que permite compartir y editar simultáneamente diagramas en equipo e integrar comentarios directamente en la herramienta” (Ionos Inc., 2021).

Como lo describe Ionos Inc.(2021), con la cuenta gratuita se puede acceder a siete plantillas de diagramas UML:

1. Diagrama de clases
2. Diagrama de estados
3. Diagrama de actividades
4. Diagrama secuencial
5. Diagrama de componentes
6. Diagrama de casos de uso
7. Diagrama de despliegue

Una de las características importantes es que cuenta con los diagramas más útiles para el diseño de bases de datos, ya que es capaz de importar su estructura directamente del Sistema Gestor de Base de Datos (SGBD) a Lucidchart y es compatible con las plataformas más populares como son MySQL, Oracle, PostgreSQL y SQL Server (Lucid Software Inc., 2023)

IBM Engineering Rhapsody Model Manager antes llamado Rational Rhapsody Model Manager, es un entorno de desarrollo gráfico integrado (IDE) para los ingenieros de sistemas y desarrolladores de software que crean sistemas y software integrados o en tiempo real. Esta herramienta para modelado UML incluye el desarrollo de software basado en objetos para aplicaciones web, así como sistemas embebidos y sistemas en tiempo real basados en C++ y Java EE. El modelado UML/SysML permite crear rápidamente código fuente en el IDE para los lenguajes especificados, C, C# y Ada (Smarter Process, s.f.).

Así mismo, se pueden dibujar los modelos libremente, importar código existente para visualizarlo en forma de diagrama o descargar los requisitos de una herramienta de gestión. Además, es compatible con todos los tipos de diagramas UML 2 actuales, sin

embargo, su desventaja es que el precio para su uso es relativamente alto (Smarter Process, s.f.).

Gliffy. Es una aplicación basada en la nube que facilita la visualización de proyectos de software con diagramas UML, cuenta con plantillas integradas y permite crear todo tipo de gráficos tales como diagramas de flujo, diagramas de Venn, diagramas UML, organigramas, diagramas entidad-relación, estructuras alámbricas, planos de planta y mapas mentales, así como también modelos y notaciones de procesos de negocio (Business Process Models and Notations, BPMN) entre otras cosas. La herramienta en línea fue escrita en HTML5 y es ideal en la colaboración visual para el trabajo en equipo ya que se pueden compartir versiones de solo lectura, así como trabajar de forma paralela e intercambiar información gracias a la función de comentarios integrada (Digital Guide IONOS, 2021).

Gliffy cuenta con una versión UML gratuita que importa diagramas en formatos GON, Gliffy y gXML, sin embargo, para otros formatos se requiere una cuenta de pago. También se pueden exportar ciertos formatos de imagen como JPG, PNG, SVG y la conexión a Google Drive. La cuenta gratuita además proporciona 2 MB de espacio en la nube para guardar gráficos y es posible compartir hasta cinco modelos al mismo tiempo. Por supuesto, la cuenta de pago ofrece más posibilidades tales como la importación ilimitada de memoria o diagramas desde Visio y la exportación a los formatos de imagen mencionados anteriormente (Digital Guide IONOS, 2021).

Draw.io. Es una herramienta de diagramación gratuita con la que se puede dibujar cualquier tipo de mapas mentales, mapas conceptuales, esquemas o diferentes representaciones gráficas, como diagrama de jerarquía o conjuntos y enriquecerlos con imágenes, iconos y enlaces; además de colaborar y cooperar con otros gracias a la sencillez para compartirlos. Cuenta con estándares de privacidad y seguridad de nivel empresarial, así como con herramientas avanzadas y útiles que están integradas en el editor de diagramas para facilitar los diseños (INTEF, s.f.) y se encuentra disponible en el sitio <https://app.diagrams.net/>.

Según Jorge Vázquez Fernando (como se citó en INTEF, s.f.) sus principales ventajas a nivel educativo son:

- **Integración:** esta herramienta puede utilizarse en versión online o de escritorio. Para las versiones online se puede vincular a diferentes cuentas y guardar los trabajos en sistemas de almacenamiento en la nube (Google Drive, OneDrive, Dropbox) o descargarlos, además permite importar diagramas desde otras aplicaciones.

- Plantillas y librerías: dispone de una serie de plantillas organizadas por tipos que hacen el trabajo más sencillo y rápido. Consta de una serie de librerías con una gran cantidad de formas, líneas, dibujos, iconos, que se pueden ir ampliando. Y si estas opciones no son suficientes, se pueden importar o crear las que se necesiten.
- Sencillez de uso: cuenta con herramientas básicas para poder desarrollar la experiencia, utiliza el sistema drag and drop (arrastrar y soltar), con el cual rápidamente se pueden llevar a la zona de trabajo formas, flechas, imágenes, etc. Otra de las características importantes de la aplicación es que el área de trabajo se extiende a medida que el diagrama, representación o dibujo crece, sin necesidad de incluir páginas. Además incluye el guardado automático, así nunca se pierde el trabajo ante cierres inesperados. Y por último, algo que puede resultar atractivo para los alumnos, es la facilidad para exportar los trabajos.
- Compartir y cooperar: es posible crear archivos que puedan ser editados por varias personas al mismo tiempo. De esta forma se fomenta entre los estudiantes el uso de entornos colaborativos y utilizar metodologías cooperativas. Finalmente, cuando se concluya el producto podrán publicar o compartir dicho archivo a través de un sencillo link (INTEF, s.f.).

Dadas estas características se puede concluir que Draw.io es una herramienta sencilla y útil con la que los alumnos pueden perfeccionar recursos del profesor para el aula que necesiten diagramas o una organización estructurada. En este sentido, se obtienen resultados con un ahorro de tiempo importante, ya que su integración con plataformas en la nube hace que el trabajo sea simple y sin preocupaciones (INTEF, s.f.).

Resultados

El uso de UML como una herramienta para la representación, interpretación y análisis de información ha demostrado ser un elemento esencial, sobre todo en aquellos campos relacionados con la ingeniería o incluso integrándose a cualquier área impulsando y fortaleciendo el pensamiento computacional. Como se estableció en el objetivo principal de este estudio, se exploraron las características claves del lenguaje UML, donde se han logrado observar que la versatilidad de los diagramas que lo conforman proporcionan una ventaja significativa tanto en la comprensión como en la solución de problemas complejos, al suministrar a través de un lenguaje gráfico la comprensión y

soluciones a dichos problemas, facilitando su entendimiento ante las diversas problemáticas.

A través de los resultados obtenidos en estos estudios y de la concordancia en los beneficios de UML se evidencia que los estudiantes pueden beneficiarse de su uso en áreas multidisciplinarias, demostrando así la gran versatilidad que puede llegar a tener como una herramienta de modelado traspasando barreras disciplinares.

Discusiones

A lo largo de este estudio, se analizaron diferentes diagramas UML y sus aplicaciones en distintos contextos. Dentro de las principales observaciones en la implementación de UML, es posible destacarse que les permite a los estudiantes desarrollar una mayor capacidad para conceptualizar los problemas y poder generar soluciones. Un claro ejemplo son los diagramas de casos de uso que tienen la facilidad de proporcionar una estructura visual que ayuda a los estudiantes a comprender y poder comunicar de manera más eficaz sus ideas, aunque éstas resulten complejas. Al integrar este tipo de diagramas puede ayudar con la clasificación de las mismas. Esto es consistente con lo planteado por Zapata-Ros (2015), quien argumenta que la visualización gráfica facilita la organización y representación lógica de procedimientos, un aspecto crucial en el desarrollo del pensamiento computacional. Por otra parte, Páez et al. (2021) plantea la eficacia y eficiencia del uso de UML en la gestión de proyectos multidisciplinarios. En su estudio se aborda la creación de un software de control de entrada/salida de bicicletas de estudiantes en la Universidad Cooperativa de Colombia, donde acentúan el logro de sus resultados gracias a la aplicación de UML y SCRUM que permitieron obtener una eficiencia en el desarrollo de dicho proyecto.

Además, Juárez et al. (2020) en su estudio, mencionan que el uso de UML en la educación mejora la eficiencia y calidad en la entrega de proyectos, destacando que los estudiantes de tecnologías y videojuegos que utilizan UML muestran una mayor capacidad para conceptualizar y resolver problemas en diversas áreas tecnológicas. A lo que Pérez-Castillo et al. (2021) contribuye también con su estudio centrado en la representación de circuitos cuánticos a través de diagramas de actividad que mencionan una facilidad en la integración de componentes clásicos y cuánticos en sistemas de información híbridos.

Por otro lado, también existen ciertas limitaciones en el uso de UML, entre ellas la complejidad de algunos diagramas que puede representar desafíos para aquellos estudiantes que no están familiarizados con los conceptos de la modelación gráfica.

Además, la adaptación de UML a problemas específicos puede requerir de un entendimiento profundo en la utilización de los diversos tipos de diagramas y sus aplicaciones, lo que implica otra de las limitaciones, que se refieren a la curva de aprendizaje respecto a la utilización de las diferentes herramientas ante la cual deberá invertirse tiempo para tener una familiaridad y conocer a detalle las funcionalidades.

Conclusiones

El estudio ha confirmado que UML es una herramienta valiosa para el desarrollo del pensamiento computacional y la representación de soluciones en la educación superior. Su implementación no solo mejora la capacidad de los estudiantes para conceptualizar y comunicar ideas, sino que también proporciona una base sólida para la colaboración efectiva y la resolución de problemas en diversos campos que puedan enfrentar los estudiantes como desafíos en el mundo digital actual y futuro.

Tomando en consideración que la colaboración entre disciplinas se ha convertido en un componente esencial respecto a la resolución de problemas complejos en el ámbito académico y profesional, UML surge como una herramienta que facilita la comunicación y el entendimiento entre diferentes áreas de conocimiento promoviendo la sinergia entre diversas especialidades destacando el rompimiento de barreras lingüísticas y disciplinares, facilitando la comunicación e integración de perspectivas y fortaleciendo no solo la eficiencia al integrarla a proyectos multidisciplinarios, sino también promoviendo la cultura de colaboración.

Futuras líneas de Investigación

Como contribuciones futuras para maximizar el potencial que puede ofrecer UML en diversos contextos tanto técnicos, educativos, de enseñanza-aprendizaje, modelado de proyectos de TI, industriales, entre otros; se proponen las siguientes líneas de investigación:

Fomento del desarrollo del pensamiento computacional: implementar un enfoque estructurado y detallado en los estudiantes, para desarrollar el pensamiento sistemático y analítico al enfrentarse a problemas complejos como estrategia de aprendizaje para abordar conceptos abstractos, preparándolos para unirse al mercado laboral con habilidades directamente aplicables y demandadas.

Efectividad del aprendizaje de diseño de software: evaluar la comprensión y el dominio de los principios de diseño, metodologías e implementación de software por

parte de los estudiantes mediante estudios comparativos entre grupos, contrastando resultados en términos de calidad y eficiencia del diseño.

Impacto en la colaboración y comunicación: referir cómo al utilizar UML los estudiantes aprenden a comunicar ideas de diseño de manera efectiva a través de diagramas y modelos, donde se promueven estas habilidades que son cruciales actualmente entre los equipos multidisciplinares, midiendo la precisión y claridad en la transmisión de ideas de diseño.

Finalmente, al aplicar UML en la educación superior se puede enriquecer la experiencia de aprendizaje de los estudiantes al proporcionarles habilidades prácticas y teóricas que les permita enfrentar los diversos desafíos en su desarrollo profesional.

Es importante recalcar que las líneas antes presentadas no son exclusivas, sino que por la naturaleza de UML, puede ser ampliamente aplicado a enfoques según lo requieran otros investigadores.

Referencias

- 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>

Rol de Contribución	Autor (es)	
Conceptualización	María del Rocío Ramírez Jiménez. <<igual>> Karime Pulido Hernández <<igual>> Carmen Elizabeth Rivera Orozco <<igual>> Liliana Serrano Zuñiga. <<igual>> Luis Mexitli Orozco Torres. <<igual>>	
Metodología	María del Rocío Ramírez Jiménez. <<igual>> Karime Pulido Hernández <<igual>> Norma Alicia Gómez Torres <<igual>> Liliana Serrano Zuñiga. <<igual>>	
Investigación	María del Rocío Ramírez Jiménez. <<igual>> Karime Pulido Hernández <<igual>> Norma Alicia Gómez Torres <<igual>> Liliana Serrano Zuñiga. <<igual>>	
Escritura - Preparación del borrador original	María del Rocío Ramírez Jiménez. <<igual>> Karime Pulido Hernández <<igual>> Norma Alicia Gómez Torres <<igual>> Liliana Serrano Zuñiga. <<igual>>	
Escritura - Revisión y edición	María del Rocío Ramírez Jiménez. <<igual>> Carmen Elizabeth Rivera Orozco <<igual>> Luis Mexitli Orozco Torres. <<igual>>	
Visualización	María del Rocío Ramírez Jiménez. <<igual>> Carmen Elizabeth Rivera Orozco <<igual>> Luis Mexitli Orozco Torres. <<igual>>	
Supervisión	María del Rocío Ramírez Jiménez. <<igual>> Luis Mexitli Orozco Torres. <<igual>>	
Administración de Proyectos	Luis Mexitli Orozco Torres. <<principal>>	
Adquisición de fondos	Luis Mexitli Orozco Torres. <<principal>>	