

EL PENSAMIENTO FORMAL, UN FACTOR DETERMINANTE EN LA ENSEÑANZA DEL DISEÑO DE CLASES BAJO UML EN LA PROGRAMACIÓN

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>

¹Estudiante del Doctorado en Ambientes y Sistemas Educativos Multimodales. 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, Ciudad de México. <https://orcid.org/0000-0003-0754-7140>

Resumen-- Este artículo pretende mostrar la investigación documental de algunos de los trabajos de investigación que se han llevado a cabo en cuanto al proceso de enseñanza del diseño y modelado en programación, teniendo como base al pensamiento formal, y se socializa el análisis de la revisión bibliográfica descriptiva, así como los resultados cuantitativos obtenidos de la aplicación de un método estadístico. De esta forma, interesa en este documento mostrar la importancia de desarrollar o fortalecer el pensamiento formal en el proceso de la enseñanza del diseño y modelado de clases en programas bajo el paradigma orientado a objetos bajo el estándar UML (Unified Model Language). Para ello, el lector podrá encontrar cómo y cuáles fueron los criterios que se consideraron para seleccionar los trabajos revisados en el apartado del método, así como la metodología para la obtención de datos cuantitativos que se obtuvieron a través de tres cuestionarios, que ayudaron a determinar si los sujetos de estudio tienen desarrollado el pensamiento formal, o solo tienen desarrollado el pensamiento concreto. Por consiguiente, en el apartado de resultados y discusión, se exponen los trabajos seleccionados y revisados, presentando a través de una síntesis los detalles más sobresalientes. Al final se presenta la perspectiva y relevancia que toma este tema para la investigación principal de la cual se desprende este artículo. Y, por último, en las conclusiones se da respuesta a la pregunta eje de este documento, ¿Por qué es importante que las y los alumnos de educación superior tengan fortalecido el pensamiento formal en el proceso de enseñanza del diseño y modelado de clases en programas bajo la técnica orientada a objetos con UML?

Palabras Clave: Estrategias enseñanza, pensamiento formal, pensamiento concreto, recursos educativos, diseño de programas orientados a objetos, clases abstractas.

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.

INTRODUCCIÓN

Este artículo se genera como un producto derivado de la propuesta de investigación que lleva por título “Análisis de la eficiencia de una metodología didáctica basada en el desarrollo del pensamiento formal en estudiantes de Ingeniería en Sistemas Computacionales en Milpa Alta, Ciudad de México”. Cuyo objetivo principal es, analizar la eficiencia de la implementación de una metodología didáctica que desarrolle el pensamiento formal para lograr el diseño y modelado de clases abstractas bajo el estándar UML, dicha investigación está dirigida a las y los estudiantes que cursan la asignatura de Programación Orientada a Objetos de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Milpa Alta.

Es por ello que interesa establecer en este artículo, el por qué es importante considerar si las y los alumnos cuentan con el pensamiento formal desarrollado, ya que se considera como uno de los factores que influyen en el logro de los aprendizajes enfocados en el modelado y diseño de diagramas de clases. Debido a que, las y los alumnos deben aprender la semiótica del estándar UML (Unified Model Language). Y, que, a través de sus tres ramas principales: la semántica, la sintáctica y la pragmática; deben aprender no solo a interpretar, sino que también a resolver problemas por medio de diagramas o esquemas.

Esto atiende a que, la enseñanza de la técnica de diseño de software orientada a objetos en los últimos tiempos se ha vuelto una necesidad de las carreras universitarias enfocadas en informática o sistemas computacionales, ya que esta técnica promueve el análisis y diseño de soluciones a problemas relacionados con el desarrollo de software [1]. Es así que, en el ámbito profesional, contar con esta competencia es de suma importancia, puesto que actualmente, la mayoría de las páginas web, aplicaciones de computadoras y dispositivos móviles, así como el software en general, están desarrollados bajo este paradigma [2].

Actualmente la técnica de Orientación a Objetos, se utiliza para desarrollar sistemas complejos, a través de los cuales interactúan diferentes disciplinas como: el desarrollo de software, el desarrollo de hardware, las ciencias de la información (internet) y la minería de datos (manejo de la información a través de bases de datos), entre otras [3]. Todos ellos trabajando en conjunto han permitido avances en la tecnología en sectores como: las telecomunicaciones, la medicina, la banca, la agricultura, el comercio, la educación, la cultura y muchas áreas del sector productivo y de servicios que tienen como fin el automatizar procesos en casi cualquier industria [4].

En concordancia, México no es la excepción, hoy en día de acuerdo al Observatorio Laboral del Gobierno de México [5], los profesionistas ocupados del área de las ingenierías con especialidad en ciencias de la computación, son aquellos que tienen mayor campo laboral. Y de acuerdo al comunicado de prensa del INEGI (Instituto Nacional de Estadística y Geografía) NÚM. 310/19, publicado en junio de 2019 [6], la Ciudad de México es la entidad que presenta mayor población ocupada en trabajos que tienen que ver con las TIC (Tecnologías de la Información y las Comunicaciones), y además el 58% de empleados tienen como nivel de escolaridad a la ES (Educación Superior), es decir, una gran cantidad de profesionistas egresados de las IES (Instituciones de Educación Superior) llegan a ser contratados dentro de la CDMX.

Entonces este estudio toma relevancia, ya que el Instituto Tecnológico de Milpa Alta, es una universidad pública perteneciente al Tecnológico Nacional de México, que se encuentra al Sur de la CDMX, ubicada en la Alcaldía de Milpa Alta. Esta institución se caracteriza por recibir alumnos y alumnas de nuevo ingreso en dos periodos al año, enero-junio y agosto-diciembre. Cuenta con 4 carreras: Ingeniería en Gestión Empresarial (IGE), Ingeniería en Bioquímica (IBQ), Ingeniería en Industrias Alimentarias (IIA) e Ingeniería en Sistemas Computacionales (ISC), en donde el programa de ISC, es el segundo plan de los cuatro que presenta una alta demanda de ingreso en comparación de los demás planes ofertados por dicha institución.

A continuación, se muestran las estadísticas obtenidas del Sistema de Control Escolar, entre el periodo Enero-Junio/2018 y Agosto-Diciembre/2022, 583 alumnos y alumnas cursaron asignaturas de primer semestre, Fundamentos de Bases de Datos, de los cuales 145 no aprobaron; 509 alumnos y alumnas cursaron la asignatura de segundo semestre, Programación Orientada a Objetos, de los cuales 153 no aprobaron; 377 alumnos y alumnas cursaron la asignatura de tercer semestre, Estructura de Datos, de los cuales 75 no aprobaron; y 284 alumnos y alumnas cursaron la asignatura de cuarto semestre, Tópicos Avanzados de Programación, de los cuales 36 no aprobaron [7].

Como se puede observar, en primera instancia el número de alumnos y alumnas que cursaron asignaturas de programación de primer a cuarto periodo va disminuyendo conforme se va avanzando en los semestres, lo que indica que hubo deserción en promedio de casi el 50% de la matrícula, en la primera mitad de la carrera. En segunda instancia, la asignatura de programación que presentó un índice de reprobación mayor fue la de segundo semestre, Programación Orientada a Objetos, en la cual se observa una disminución considerable de la matrícula al concluir el semestre dos. Y, por último, se consideran las estadísticas de estos cuatro primeros semestres, ya que, de acuerdo al plan de estudios, son las asignaturas obligatorias de la carrera que aplican la técnica orientada a objetos en el diseño de software, y que, además, sirven como base para las asignaturas de toda la carrera, por lo que las competencias desarrolladas en ellas son fundamentales para el proceso formativo de las y los alumnos.

En este caso, la población a la que se pretende beneficiar es a aquellas y aquellos alumnos de educación superior que estén cursando una carrera de Ingeniería en Sistemas Computacionales, cuyo modelo curricular se encuentre enfocado en desarrollar y/o fortalecer el pensamiento formal, para así aprender a diseñar programas bajo la técnica de programación orientada a objetos, identificando en una primera instancia la importancia de

la resolución de problemas a través del modelado de clases abstractas.

DESARROLLO

Metodología

En primera instancia, se llevó a cabo una revisión sistemática bibliográfica, para la cual se tomó como referencia la propuesta de [8], por lo que se realizaron de manera general las siguientes actividades:

1) Pregunta: en este artículo el objetivo será establecer la pertinencia del desarrollo o fortalecimiento del pensamiento formal en la enseñanza del diseño de clases en programas orientados a objetos.

2) Protocolo: Se establecieron los criterios para elegir o excluir literatura, se determinaron los buscadores en los cuáles se llevaron a cabo las búsquedas, como: Redalyc, Scielo, Google académico y Dialnet. Así como el periodo de búsqueda, el cual está entre 2010 y 2024.

3) Búsqueda: se retomaron antecedentes, tomando como referente proyectos enfocados en tomar el desarrollo del pensamiento formal como un factor a considerar en el proceso enseñanza-aprendizaje del diseño y modelado de programas, así como estrategias de enseñanza en la programación. Se utilizaron palabras clave como: enseñanza-aprendizaje, programación orientada a objetos, pensamiento abstracto, pensamiento formal, diagramas de clases, enseñanza diagramas de clase UML.

4) Selección: Se seleccionaron estudios que cumplieron con los criterios de elegibilidad, los cuales fueron, aquellos estudios que intenten explicar qué se considera como pensamiento abstracto o formal en la disciplina de la programación y la resolución de problemas, y cómo se puede promover una mejor experiencia del proceso enseñanza en el aula, en este caso, a través del uso de un esquema que permita diseñar y modelar un programa bajo la técnica orientada a objetos.

5) Extracción: De cada literatura seleccionada, se extrajeron los datos principales de cada uno, como: autores, tema de investigación, fuentes de información, objetivo general de estudio, metodología, resultados obtenidos, recomendaciones para estudios futuros y palabras clave.

6) Evaluación: En este paso se evaluaron tanto la calidad metodológica, es decir, si se consideran tanto el sesgo, como la validación de los estudios encontrados. De igual forma, las referencias utilizadas en cada estudio encontrado.

7) Síntesis: Se elaboró una tabla con los elementos obtenidos de la extracción, resumiendo a cada uno de los elementos de la literatura seleccionada. Y, a partir de ahí

se identificaron y clasificaron los contenidos, para obtener las categorías mostradas en el apartado de discusión y análisis de resultados.

8) Redacción: se hizo un análisis y discusión de los resultados obtenidos, así como la conclusión que se hace al revisar la literatura de la importancia y pertinencia del desarrollo del pensamiento formal en el proceso de enseñanza-aprendizaje del modelado de soluciones orientadas a objetos a través de esquemas llamados, modelos de clases abstractas.

Y de forma conjunta a esta revisión sistemática documental, se aplicó un método estadístico sobre una investigación descriptiva; cuyo objetivo, es establecer un diagnóstico de la presencia o no del pensamiento formal en las y los alumnos, para evidenciarlo como un factor, de los muchos que existen, a considerar en el proceso enseñanza-aprendizaje, y establecer que la ausencia de este pensamiento formal fortalecido puede significar retrasos en el proceso de enseñanza. Es así, que la hipótesis que se plantea es: La presencia del pensamiento formal en las y los alumnos permite situarlos en el campo de lo abstracto para que puedan resolver problemas haciendo uso de la semiótica de diagramas de clases con UML en la asignatura de programación orientada a objetos de segundo semestre de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Milpa Alta. Por lo que se llevaron a cabo las siguientes actividades:

1) Recolección de datos: Se visitó a la fuente primaria al inicio del semestre, y para evitar sesgos, la aplicación de todos y cada uno de los instrumentos se llevó a cabo en el mismo día, y a la misma hora, lo que dio validez interna al estudio. Los instrumentos utilizados para la recolección de datos fueron un cuestionario socio-académico, un test para medir el nivel de pensamiento concreto existente, un test para medir la presencia o ausencia del pensamiento formal, y, un test para medir el nivel de fortalecimiento del pensamiento formal. Estos, se aplicaron de forma sistemática haciendo uso de una herramienta en línea. También se utilizó a la observación simple, la cual fue registrada en una bitácora digital, en la cual se respaldaron todos los sucesos que se generaron durante la recolección de datos. La población de alumnos que se encontraba cursando la asignatura en el periodo Enero-Junio/2024, fue de 56. Considerando un nivel de confianza de 95 por ciento, y un error del 5 por ciento, el tamaño de la muestra considerada es de 49 alumnas y alumnos, los cuales se eligen de forma aleatoria.

Datos:

Población, $N = 56$

Margen de error, $e = 0.05$

Nivel de confianza, 95% por lo tanto, $k = 1.96$

Probabilidad de éxito, $p = .5$

Probabilidad de fracaso, $q = 1 - p = .5$

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

2) Recuento y Análisis: Una vez que se aplicaron los instrumentos, se llevó a cabo un procedimiento de análisis de los datos que reflejan de forma estadística la información, y así poder identificar aquellos elementos que servirán para interpretar los datos obtenidos.

3) Presentación e interpretación: Una vez que se hizo el tratamiento de datos, y que se representaron a través de tablas, gráficos y/o figuras, se llevó a cabo un proceso para interpretar la información obtenida, y así determinar, si la hipótesis se cumplió o no.

DISCUSIÓN Y ANÁLISIS DE RESULTADOS

Importancia del pensamiento formal en la resolución de problemas

El perfil de egreso de un estudiante de Ingeniería en Sistemas Computacionales del TecNM, se conforma de 15 competencias de las cuales, se toma como referente para esta investigación la capacidad de aplicar nuevas tecnologías en la solución de problemas de su entorno laboral [9]. Para dar solución a estos problemas, una de las tantas habilidades que todo ingeniero debe fortalecer es el pensamiento abstracto, lógico y deductivo; que en su conjunto llamaremos en esta investigación pensamiento formal.

El que las y los alumnos de educación superior cuenten con este pensamiento formal, les podría permitir comprender, analizar y modelar los problemas de la información y el conocimiento en la sociedad actual, y de esta forma puedan llegar a presentar soluciones eficientes y eficaces [10].

Vale la pena en este punto aclarar qué se entiende por pensamiento formal para esta investigación, y su diferencia con el pensamiento concreto, que, si bien en su conceptualización son dicotómicos, en el proceso de enseñanza deberían complementarse, para así poder plantear una serie de estrategias de enseñanza del diseño y modelado de programas bajo la técnica orientada a objetos.

A pesar de esta consideración que debe hacerse, se ha identificado que, en la mayoría de los estudios seleccionados para este artículo, los docentes-investigadores prefieren quedarse solo en la enseñanza del pensamiento concreto, el cual consiste en observar un fenómeno de la vida real, al cual se le clasifica y explica únicamente de manera lógica en cuanto a cómo se compone y sus características. De acuerdo a Piaget, este pensamiento concreto se desarrolla desde la primera infancia hasta la adolescencia [11].

A diferencia del pensamiento concreto, el pensamiento formal, según Piaget, se desarrolla entre los 11 y 15 años, donde las y los alumnos son capaces de resolver problemas más complejos haciendo uso de conceptos y teorías para la comprensión del comportamiento de dichos fenómenos, los cuales ahora los conceptualizará como objetos, y podrá identificar y determinar cómo se vincula con otros objetos, ya sea con las mismas o diferentes características [12].

Por su parte Vygotski, se refiere a esta etapa como conceptos en acción, puesto que, el estudiante de nivel superior toma características mentales de etapas anteriores, y, si existen nuevos procesos mentales pueden construir soluciones a problemas de tipo hipotético deductivo. Marta Uribe, describe que, en esta etapa universitaria, también se desarrolla la capacidad de utilizar símbolos para representar objetos, categorías, y operarlos entre ellos. Sin embargo, si en el proceso de enseñanza el docente sólo presenta secuencias didácticas enfocadas a utilizar únicamente el pensamiento concreto, las y los alumnos, no podrán desarrollar o fortalecer este pensamiento formal, ya que no habrá un estímulo para desarrollar nuevas habilidades [14].

[14], [15], [16] y [17] presentan estudios en los cuales señalan que, no importando la edad, o el entorno y estímulos a los que han sido sometidos las y los estudiantes universitarios, aproximadamente un 30% no ha desarrollado el pensamiento formal, y casi el 50% ha logrado desarrollar dicho pensamiento formal. Esto implica que, las y los docentes deben establecer estrategias o metodologías didácticas que promuevan el desarrollo del pensamiento formal, ya que, de otra forma, el impacto negativo que se tiene en la trayectoria de las y los alumnos es muy marcado, llegando a ser uno de los tantos factores que influyen en la deserción escolar.

[18] y [19], coinciden en que hay un vínculo entre el pensamiento formal y el desarrollo de la capacidad abstractiva, ya que esta habilidad es clave en la formación de cualquier ingeniería, y para este caso en el de las ciencias computacionales que implica la resolución de problemas a partir de la observación de fenómenos [19].

Entonces, para poder plantear posibles soluciones, se tendría que enseñar a las y los alumnos a realizar un proceso de razonamiento que permita reducir la complejidad de un problema, dividirlo en módulos y a su vez cada módulo se podría conceptualizar o generalizar a través del uso de los patrones que se extraigan. Y una vez, que se tenga el diseño de todos y cada uno de los módulos, el proceso es vincularlos a todos ellos. La intención tendría que ser, que todas estas soluciones se puedan abordar desde cualquier disciplina de las ciencias computacionales, como: las telecomunicaciones, las

bases de datos, la inteligencia artificial, y por supuesto, la programación en cualquiera de sus paradigmas [18].

Entonces resulta pertinente tomar la definición de Garrido, quien define que el pensamiento formal es un proceso que tiene que ver con las estructuras mentales, y que este proceso toma del objeto de estudio detalles que son relevantes para poder simplificar un problema. Y, para poder conceptualizar las posibles soluciones a un problema se lleva a cabo a través del desarrollo de modelos en diferentes niveles de abstracción [20].

Semiótica del UML para el diseño y modelado de clases de programas orientados a objetos

En el apartado anterior, se identificó que el desarrollo del pensamiento formal implica que las y los alumnos puedan operar objetos a través de teorías, y para ello deben tener la capacidad de sintetizar grandes cantidades de información. Por lo que resulta conveniente el uso de esquemas que, por un lado, permiten conjuntar en módulos o bloques que representen de manera sintética la conceptualización de la información, y, por otro lado, relacionen de forma más compleja a dichos bloques de información.

Es por ello que la competencia de llevar a cabo representaciones por medio de sistemas figurativos, tampoco debe quedarse en la etapa del pensamiento concreto. Brandi, muestra a través de su trabajo de grado, la importancia de la alfabetización gráfica, la cual debe consistir en la enseñanza de las características sintácticas y la taxonomía de un sistema figurativo [21]. Esto implica, el aprender a generar e interpretar desarrollos conceptuales complejos, que se sintetizan a través de la semiótica.

Por lo que, si se habla desde la perspectiva de la semiótica, los diagramas UML (Unified Model Language) permiten representar de forma gráfica conjuntos lógicos de elementos de uno más modelos relacionados entre sí [22]. Estos diagramas UML, han sido utilizados en la industria de las ciencias computacionales por su fácil uso y representación, debido a que, al ser un lenguaje universal, y bajo una sintaxis definida, el mismo elemento figurativo se puede interpretar a partir de diferentes perspectivas, donde cada una de ellas, es una parte del área de especialidad con la cual se abordará la solución, por ejemplo: base de datos, telecomunicaciones, programación, ingeniería de software, servidores, cómputo en la nube, seguridad informática, etc.

Consideremos ahora que en la programación existen dos formas de llevar a cabo el proceso de modelado. Por una parte, está la abstracción de especificación, y por otra la abstracción de implementación [20]. Donde el primer tipo de abstracción, sirve para identificar las características y

así determinar qué se debe desarrollar; para este tipo de abstracción regularmente se utilizan estructuras bajo el estándar UML como: lluvia de ideas, diagrama de niveles de abstracción, diagramas de casos de uso o diagramas de clase, entre otros. Mientras que el segundo tipo de abstracción, sirve para determinar cómo van a interactuar los módulos y así dar funcionalidad al programa; para este tipo de abstracción también se realizan bajo el estándar UML, y regularmente se utilizan estructuras como: diagramas de flujo, diagramas de estados, diagramas de secuencias, diagramas de actividad, entre otros [23].

Entonces, los diagramas de UML pueden elaborarse para representar estructuras estáticas, casos de uso, secuencias, colaboraciones entre usuario-maquina-usuario, y de estados [24]. Dentro de la clasificación de los diagramas para representar estructuras estáticas, se encuentran los diagramas de clases, los cuales intervienen en los procesos de ingeniería de software, y, que se desarrollan como parte del diseño de una propuesta de solución.

A estos modelos de clases, se les puede ver como la representación de la realidad, y cada signo utilizado representa una parte de dicha realidad, y se les define como, modelos orientados a objetos. Es por ello que, los diagramas de clases UML tienen una ventaja por encima de los lenguajes de modelado imperativos, declarativos o funcionales [23]. Debido a que el desarrollo del diagrama de clases, se inicia haciendo uso del pensamiento concreto, donde se identifica al objeto, su estructura y características; y, con el pensamiento formal se determina, para cada uno de los objetos identificados, cómo se crean, cómo funcionan o bien cómo interactúan entre sí ya en la programación.

Análisis de resultados estadísticos

Para este artículo se desarrolló un cuestionario, tres test, y se implementó una bitácora de observación. El cuestionario, se implementó para poder identificar el perfil socio-académico y de hábitos académicos de los sujetos estudio. Los test se implementaron para determinar el nivel de pensamiento que tenían desarrollado hasta ese momento los sujetos de estudio, y de esta forma determinar el porcentaje de alumnos que presentaban un pensamiento formal desarrollado o fortalecido. Y, la bitácora de observación, se implementó para anotar posibles mejoras en la implementación de la investigación principal, ideas, datos que no se reportan en los test, y observaciones en general de la aplicación de los test. Por lo que, a continuación, se describe el análisis y resultados obtenidos de los instrumentos de recolección de datos implementados.

El cuestionario socio-académico y de hábitos de estudio permitió identificar que:

- El 80% son de género masculino
- El 40% de los sujetos de estudio tienen como lugar de residencia a la Alcaldía de Xochimilco, a más de 30 minutos de distancia.
- El 96% de la muestra son solteros y no tienen hijos,
- El 74% de los sujetos de estudio no trabajan.
- El 55% proviene de la institución de Educación Media Superior, CETIS.
- El 76% de los sujetos de estudio tienen como estilos de aprendizaje predominantes la kinestésica y auditiva.
- El 80% de los sujetos de estudio cuenta con espacios aptos o adaptados para sus sesiones de estudio.
- El 40% de los sujetos de estudio no tienen un horario fijo para sus sesiones de estudio.
- El 30% realiza esquemas para reforzar sus conocimientos.

En cuanto a los test, el primer test permite determinar si los sujetos de estudio tienen desarrollado el pensamiento concreto al identificar conceptos y definiciones, así como los elementos que conforman a un diagrama de clases.

Entonces, al aplicar el primer test a los sujetos de estudio, se les solicitó que identificaran los elementos de una clase, la sintaxis, relacionar conceptos con su respectiva definición, y por último interpretar un diagrama completo. Para lo cual, se obtuvieron los siguientes resultados:

Tabla 1. Conceptos que permiten identificar si los sujetos de estudio tienen presente el pensamiento concreto, a través de la identificación de conceptos y clasificación de información.

Conceptos concretos	Porcentaje de alumnos que respondieron correctamente
Identifica elementos que conforman una clase y su orden	76%
Sintaxis de una clase	78%
Identifica la definición de una clase	93%
Identifica la definición de objeto	57%
Identifica la definición de un atributo	98%
Identifica la definición de un método	78%
Identifica el concepto de abstracción	72%
Identifica los elementos de un diagrama de clases: clase, nivel de acceso, tipo de dato, parámetros, asociaciones.	92%

Esto implica que, desde el punto de vista concreto, los sujetos de estudio identifican a este tipo de diagramas, sus elementos, definiciones, conceptos y cómo interactúan entre sí, es decir, el pensamiento concreto, está presente.

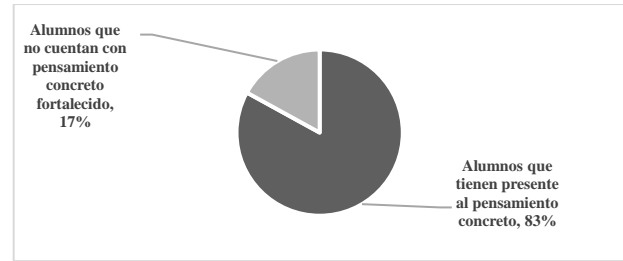


Figura 1. Gráfica comparativa de la presencia del pensamiento concreto en los sujetos de estudio.

El segundo test, se aplicó para identificar, si estos mismos sujetos de estudio tienen desarrollado el pensamiento formal para resolver un problema que tenga como resultado la elaboración de un diagrama de clases. Para ello, se dio una lectura que describe la simbología del estándar UML, en qué casos utilizar dicha simbología, y al final de la lectura se muestran tres ejemplos de aplicación.

Entonces, una vez concluida la lectura, se presentaron dos problemas, con planteamientos diferentes, tanto en datos como en redacción, el 48% pudo modelar únicamente los atributos de las clases; y solo el 32% pudo determinar atributos y métodos; establecer la interacción, vínculos y uso de sintaxis entre las diferentes clases de forma correcta, 20% respondió correctamente; el restante de cada criterio no respondió.

Tabla 2. Criterios que permiten identificar si los sujetos de estudio tienen presente al pensamiento formal, a través de la resolución de ejercicios de diseño y modelado de clases con UML

Criterio de medición	Porcentaje de alumnos que lograron realizar el ejercicio de resolución de problemas
Modelar una clase haciendo uso de la semiótica del estándar UML	48%
Modelar atributos y métodos haciendo uso de la semiótica del estándar UML	32%
Modelar la interacción, vínculos y uso de sintaxis entre las diferentes clases	20%

Esto implica, que, si bien saben los conceptos, al articularlos, no hay una verdadera comprensión de cómo hacerlo, y esto implica una barrera para la siguiente etapa, que es la de la implementación a través de la interpretación de estos diagramas haciendo uso de la programación.

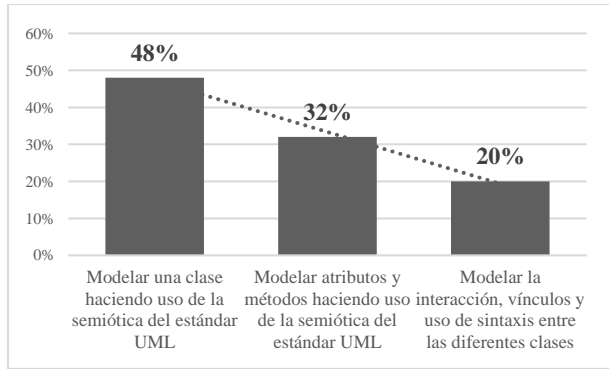


Figura 2. Gráfica de tendencia, en la cual se muestra como a mayor complejidad de los ejercicios menor número de alumnos responde de forma correcta.

Por último, el tercer test, permite identificar si los sujetos de estudio, son capaces de trasladar del diagrama de clases a la programación interpretando un modelo de clases que les fue proporcionado, con ello, se pretende determinar si hay un pensamiento formal fortalecido.

Al aplicar este tercer test a la misma muestra, el cual consiste en identificar si los sujetos de estudio, pueden implementar en la programación la interpretación del diagrama, se observó que, solo el 30%, pudo crear las clases y vincularlas conforme a el diagrama establecido de manera correcta, y el 46% se quedó únicamente en la construcción de las clases, sin lograr vincularlas a través de la programación; el 24% restante no respondió.

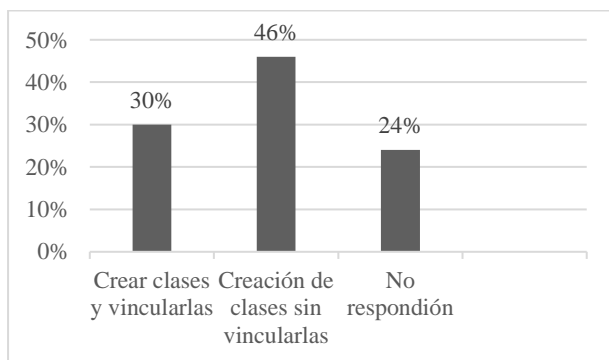


Figura 3. Gráfica del tercer test, donde se pretende identificar, si los sujetos de estudio tienen el pensamiento formal fortalecido.

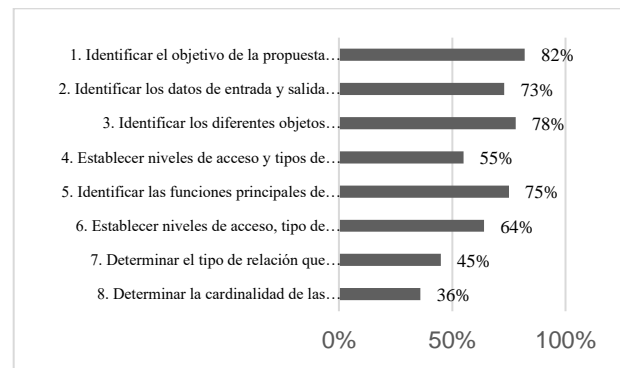
Esto, indica que el pensamiento formal aún no se ha fortalecido en la mayoría de las y los alumnos en esta etapa de su formación.

Otro factor importante que se observó y registró en la bitácora, fue que los sujetos de estudio, no cuentan con un método para resolver este tipo de problemas. En el segundo test, se preguntó a los sujetos de estudio: *Ordena los siguientes pasos que se llevan a cabo para elaborar un diagrama de clases.* En la tabla 4, se muestran los pasos y el número de personas que ubicaron la secuencia de manera correcta. Esto implica que, al no tener definida

la manera en la que se construye un diagrama de clases, repercute en el resultado final del diagrama, así como en el uso correcto de la semiótica.

Tabla 3. Numero de alumnos que respondieron de forma correcta el orden del paso a llevar a cabo en el planteamiento del desarrollo de un diagrama de clases.

Paso	Alumnos que respondieron en el orden correcto
1. Identificar el objetivo de la propuesta de solución	45
2. Identificar los datos de entrada y salida de la propuesta de solución	40
3. Identificar los diferentes objetos involucrados	43
4. Establecer niveles de acceso y tipos de datos de los atributos	30
5. Identificar las funciones principales de cada uno de los objetos	41
6. Establecer niveles de acceso, tipo de datos de salida y parámetros necesarios de cada uno de los métodos	35
7. Determinar el tipo de relación que puede o no existir entre los objetos	25
8. Determinar la cardinalidad de las relaciones entre objetos	19



Gráfica 1. Porcentaje de alumnos que ubicaron de forma correcta el orden en que debe llevarse a cabo cada uno de los pasos en el proceso del desarrollo de un diagrama de clases.

CONCLUSIONES

Al plantear la hipótesis de este artículo: “La presencia del pensamiento formal en las y los alumnos permite situarlos en el campo de lo abstracto para que puedan resolver problemas haciendo uso de la semiótica de diagramas de clases con UML en la asignatura de programación orientada a objetos de segundo semestre de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Milpa Alta”; se elaboraron los instrumentos de recolección de datos, con la intención de mostrar el porqué de la importancia de que las y los alumnos de educación superior cuenten con el pensamiento formal fortalecido en esta etapa de su

formación como ingenieros e ingenieras en sistemas computacionales.

El desarrollo o fortalecimiento del pensamiento formal no es un concepto de reciente creación, sin embargo, en los últimos tiempos se ha retomado con mayor énfasis en los procesos de enseñanza, debido a la complejidad de los problemas que deben resolverse en los últimos tiempos.

El poner énfasis en la forma en cómo serán representadas las soluciones a problemas que, en este caso, serán resueltos a través de las ciencias computacionales, es debido a que esta solución o soluciones se construyen mediante la abstracción de objetos del mundo real, que permiten el diseño y modelado de clases, que de forma posterior se programarán.

Como se pudo identificar tanto en la literatura seleccionada para este artículo, como en los test aplicados, y la bitácora de observación, las y los alumnos comprenden bien la sintaxis y taxonomía; sin embargo, al solicitar que ellos a partir de un problema planteado desarrollen un modelo, pocos son los que logran elaborar un diagrama con la semiótica que marca UML, así como la implementación en la programación. [25] resalta la importancia de primero comprender que la resolución de un problema que será resuelto a través de la programación, puede ser abordado utilizando estrategias mentales como el modelamiento, la abstracción, la descomposición y el razonamiento heurístico [25].

Por lo que, la importancia del desarrollo de esta capacidad abstractiva deriva en la relevancia que toma el modelar interpretaciones para la resolución de problemas; entonces, esta habilidad, permite descomponer un problema, lo que ayuda en el análisis y comprensión del mismo, y de este modo crear modelos para plantear sus posibles soluciones. De esta forma el pensamiento formal se convierte en un factor importante para la ciencia y la ingeniería en cualquiera de sus ramas, sobre todo en el desarrollo de la ciencia y la tecnología.

[18] refiere que la habilidad de resolver un problema, implica que primero se identifican objetos de estudio del mundo real, y de ellos se obtienen características y funcionalidades, que estarán vinculados entre sí, y a cada objeto se le denomina módulo. Por lo que, al tener problemas más complejos, se identifican varios objetos de estudio que conforman un todo, y, por ende, de cada uno se debe llevar a cabo el mismo proceso. De ahí que, para un problema más complejo, más componentes van a ser parte de la solución. Entonces, es necesario que los programadores cuenten con una o más herramientas que les permita identificar a todos y cada uno de los módulos, así como las relaciones que hay entre ellos.

Como bien menciona Álvaro Carbajal, “los modelos son instrumentos o esquemas conceptuales por los cuales los seres humanos intentan articular de forma sistemática el conocimiento que se tiene de la experiencia mediante el proceso de investigación. Ambos contribuyen a tener una imagen de la realidad...” [26].

Dadas estas características, y considerando que los recursos a utilizar deben atender a las necesidades de contexto y actualidad, ahora toca llevar a cabo investigaciones más profundas, la primera que se propone, es una que analice si el implementar una metodología didáctica que fortalezca el pensamiento formal, haciendo uso de los nuevos ambientes de aprendizaje, puede servir como apoyo para aquellos docentes que deseen mejorar sus prácticas educativas.

Otra investigación que puede desprenderse de este artículo es, crear o proponer espacios de trabajo que tengan un sentido explícito, en los cuales se integren comunidades de práctica, donde la interacción entre pares y con los docentes tome relevancia. Donde aquellas actividades, recursos, instrumentos de evaluación y secuencias didácticas permitan proporcionar una guía que apoye a optimizar este proceso de enseñanza.

Por último, se concluye que, si bien los factores que influyen en el éxito o no del proceso de enseñanza de los diagramas de clases son multifactoriales, el pensamiento formal, es un factor que sí ayuda en la comprensión y construcción del andamiaje de las y los alumnos de educación superior, a través del fortalecimiento de estructuras mentales, habilidades y capacidades, que permiten una mejor adquisición y aprehensión de la semiótica de los diagramas de clases desarrollados con UML.

AGRADECIMIENTOS

Agradezco al Instituto Tecnológico de Milpa Alta, por permitirme llevar a cabo este proyecto de investigación, el cual busca aportar una propuesta de metodología didáctica que impacte en las áreas de oportunidad de los procesos de enseñanza-aprendizaje de la comunidad de dicha institución.

El presente trabajo se desarrolló en el marco de los estudios del Doctorado en Ambientes y Sistemas Educativos Multimodales de la Universidad Rosario Castellanos para la obtención del grado.

A la DTT. Dra. María Eugenia Rodríguez Paz. Y el CT. Conformado por la Dra. Ana Beatriz Vega Cruz y Dr. José Jacobo Gómez Quiroz.

BIBLIOGRAFÍA

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

- [2] Sánchez-García, J., Urías-Ruíz, M., Gutiérrez-Herrera, B. (2015). Análisis 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). La evolución de la automatización de procesos. Recuperado de: <https://www.ibm.com/downloads/cas/RJGDMZ2D>
- [5] Servicio Nacional de Empleo. (2023). Ingenierías. Recuperado de: <https://www.observatoriolaboral.gob.mx/static/estudios-publicaciones/Ingenierias.html>
- [6] INEGI. (2019). Estadísticas a propósito de las ocupaciones relacionadas con las tecnologías de la información y de la comunicación datos nacionales 1. Recuperado de: https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2019/OcupaTIC2019_Nal.pdf
- [7] Instituto Tecnológico de Milpa Alta. (2023). Índice de reprobación - Ingeniería en sistemas computacionales por periodo Enero-Junio/2018 a Agosto-Diciembre/2022. Recuperado de: Departamento de Servicios Escolares.
- [8] Universidad de Navarra. (2024). Pasos para realizar una revisión sistemática. Recuperado de: <https://biblioguias.unav.edu/revisionessistematicas/pasos-realizar-revisionsistematica>
- [9] TecNM - Campus Milpa Alta. (2024). Perfil de egreso de la carrera de Ingeniería en Sistemas Computacionales. Obtenido de: <https://itmilpaalta.edu.mx/pag/isc.php>
- [10] Serna, M., & Polo, J. (2014). Lógica y abstracción en la formación de ingenieros: una relación necesaria. Ingeniería Investigación y Tecnología, 15-2, pp. 299-310.
- [11] Ibarra, L. (2022). Pensar pedagógico y pensar concreto-abstracto. 9-1. Recuperado de: https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-78902021000800015
- [12] UNIR. (2022). Etapa preoperacional: en qué consiste e importancia en los niños. Recuperado de: <https://mexico.unir.net/noticias/educacion/etapa-preoperacional/>
- [13] Vygotski, L. S (2007). Pensamiento y lenguaje en Vygotski, L. S. Obras escogidas II, Madrid: Aprendizaje. Visor. citado en Cano, A., Cognición en el adolescente según Piaget y Vygotski. ¿Dos caras de la misma moneda?, Boletim Academia Paulistana de Psicología 27, pp. 148 - 166
- [14] Uribe, Marta (1993). El desarrollo del pensamiento formal y la docencia universitaria. Recuperado de: <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) Pensamiento matemático en estudiantes universitarios. 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. Recuperado de: <https://www.iisue.unam.mx/perfiles/articulo/1987-37-el-pensamiento-del-adolescente-y-el-diseno-curricular-en-educacion-media-superior.pdf>
- [17] Vázquez, L., & Segarra, P. (2017). Desarrollo del pensamiento formal mediante análisis y evaluación de información científica. Latin-American Journal of Physics Education, 11-2. Recuperado de: <https://dialnet.unirioja.es/servlet/articulo?codigo=6353424>
- [18] Serna, E. (2015). La capacidad lógico interpretativa y abstractiva. Libro electrónico. Medellín: Fondo Editorial ITM, Instituto Antioqueño de Investigación.
- [19] Zapotecatl, J. L. (coordinador) (2018). Introducción al pensamiento computacional: conceptos básicos para todos. Primera edición. México: Academia Mexicana de Computación, A.C.
- [20] Garrido, A. (2018). Abstracción en programación. Granada, España: Departamento de Ciencias de la Computación e I.A. ETS Ingeniería Informática. Recuperado de: <https://ccia.ugr.es/~jfv/abstraccion.pdf>
- [21] Brandi, J. P. (2016). Los gráficos en la enseñanza y el aprendizaje. Recuperado de: <https://core.ac.uk/download/pdf/296390387.pdf>
- [22] Génova, G., Valiente M. C., & Nubiola, J. (2006). Modelos en UML: un enfoque semiótico. Novática, 180, marzo-abril. Recuperado de: <http://www.ie.inf.uc3m.es/ggenova/pub-novatica2006a.pdf>
- [23] Sommerville, I. (2011). Capítulo 5. Modelado del sistema. En I. Sommerville, Ingeniería de Software, pp. 118-147. México, México: Pearson Educación.
- [24] Ferré, X., & Sánchez, I. (2011). Desarrollo Orientado a Objetos con UML. Universidad Veracruzana. Recuperado de: <https://www.uv.mx/personal/maymendez/files/2011/05/umltotal.pdf>
- [25] B@UNAM (2022). Pensamiento computacional y creatividad. 172-9. Obtenido de: <https://boletin1020.bunam.unam.mx/pdf/172.pdf>
- [26] Carvajal, Á. (2002). Teorías y modelos: formas de representación de la realidad. Comunicación, 01 (12), pp. 1-14. Recuperado de: <https://www.redalyc.org/pdf/166/16612103.pdf>
- [27] Arenas, L. (2008). Asignatura de Programación Orientada a Objetos. Recuperado de: <http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/opt/java.pdf>
- [28] Weisfeld, M. (2013). The Object-Oriented Thought Process. 3a edición. USA: Pearson Education.
- [29] Bustos, G. (1999). Modelado Orientado a Objetos: Una Evaluación Crítica. Recuperado de: https://www.sites.upiicsa.ipn.mx/archivos/profesores/hmontes/cursos/orientada-objetos/23-2/tareas/tarea_2b.pdf

- [30] Muñoz, J., Álvarez, F., Garza, L. A., & Pinales, F. J. (2005). Modelo para el aprendizaje colaborativo del análisis y diseño orientado a objetos. 5-1, pp.73-82.
- [31] Gonzáles, E. I., López, A., Trujillo, V., & Rojas, R. (2018). Estrategia didáctica de enseñanza y aprendizaje para programadores de software. RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo. Recuperado de: <https://ride.org.mx/index.php/RIDE/article/view/402/2079>.
- [32] López, L. (2007). Metodología para la enseñanza aprendizaje de la lógica de la programación orientada a objetos. Recuperado de: <https://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2007/lometo.pdf>
- [33] Rodríguez, C. (2014). Algunos elementos de la didáctica de la programación y el pensamiento computacional. [Tesis de licenciatura]. Facultad de Ciencias, UNAM, CDMX. Recuperado de: https://repositorio.unam.mx/contenidos/ficha/algunos-elementos-de-la-didactica-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). Recursos educativos didácticos en el proceso de enseñanza-aprendizaje. Cuadernos, 58-1, pp. 68-74. Recuperado de: http://www.scielo.org.bo/pdf/chc/v58n1/v58n1_a11.pdf
- [35] Marrero, A. L., & Díaz, Maiky. (2022). Elaboración e implementación de recursos educativos para estimular las competencias profesionales. Recuperado 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). Recursos educativos. Generalidades para su desarrollo y evaluación. Recuperado de: https://suayed.facmed.unam.mx/wp-content/uploads/2022/03/recursos_educativos_mar29.pdf
- [37] Universidad Rosario Castellanos. (2024). Morfología de los PLE. Recuperado de: https://edu.rcastellanos.cdmx.gob.mx/posgrado/pluginfile.php/200/mod_resource/content/16/index.html
- [38] Organismo de cooperación para el desarrollo en educación, ciencia, tecnología e innovación A.C. (2017). Modelo Socioconstructivista de Educación Integral. Recuperado de: <http://www.uno.edu.mx/downloads/DocumentosOficiales/ModeloEducativoSocioconstructivistaparalaEducacionIntegral.pdf>



Esta obra está bajo
una licencia internacional
Creative Commons Atribución 4.0.