



PENSAMIENTO LÓGICO Y PROGRAMÁTICA COMPUTACIONAL EN ESTUDIANTES DE INGENIERÍA

Logical thinking and computational programmatics in engineering students

RUBÉN ALEXANDER MORE VALENCIA , JUAN MANUEL TUME RUIZ , ANTIA RANGEL VEGA
Universidad Nacional de Piura, Perú

KEYWORDS

Thought
Logical Thinking
Computational Programmatic
Communication
Teaching Engineering
Engineering students

ABSTRACT

The purposes of studies on computational logical thinking foresee that it is possible, through experiment work, to do research on previous skills of logic and mathematics in engineering students, by using packages of software and applications and supporting these students in parallel. It was possible that teachers achieved impact by using co-creation, and, since their work was collaborative, this showed leadership, motivation, and guidance. By using computer programming and analytical tools integrated into the capacities of students, this not only allowed collaboration, but also development and progress for building their own knowledge, in a process according to the pedagogical theory of constructionism.

PALABRAS CLAVE

Pensamiento
Pensamiento Lógico
Programática Computacional
Comunicación
Enseñanza Ingeniería
Estudiantes Ingeniería

RESUMEN

Los propósitos de los estudios sobre pensamiento lógico computacional prevén que es posible investigar con esfuerzos experimentales, las habilidades previas de lógica y matemática para estudiantes de ingeniería con paquetes de software y aplicaciones, en forma paralela dar soporte a los estudios universitarios de nivel superior, se logró que docentes impacten con co-creación, siendo colaborativo su trabajo de liderazgo, así como motivador y director, con usos de programática computacional, las herramientas analíticas integradas en capacidades de estudiantes, no solo permitió colaboración, también desarrollos y progreso para construir su propio conocimiento, en un proceso descrito por la teoría pedagógica del construccionismo.

Recibido: 06/ 06 / 2022

Aceptado: 25/ 08 / 2022

1. Introducción

La distancia de los centros donde la asistencia física para la enseñanza de los estudiantes, permite el uso de laboratorios y su desarrollo con competencias en lógica computacional, en todo nivel educativo, incluyendo a los estudiantes de educación básica, con herramientas computacionales, que asegura que los mismos docentes diseñen guías y puedan apoyarse en habilidades de programación y pensamiento computacional (Computational thinking). Estos factores reforzaron la idea de utilizar la tecnología en la virtualidad para las clases y, sobre todo, evaluar su efectividad. Los productos académicos resultantes implican no solo la forma y desarrollo del aprendizaje, también la profundidad de pensamiento de estudiantes y docentes para procesar, y ejecutar actividades en entornos digitales y de comunicación digital asíncrona, así como mejora continua, con planes que antes de pandemia, instituciones académicas lograron implementar (Marcelino et al., 2018).

Los efectos evaluados a temprana edad, especialmente con niños, generan una apreciación del aprendizaje basado en enfrentar casos y problemas para codificar, programar y buscar una sistematización a los problemas, así como a casos o ejercicios académicos a los que se enfrentan, logrando efectos físicos neurológicos (Yadav et al., 2014). Los estudios longitudinales que se han realizado, prevén que es posible administrar estudios, que enfoquen los esfuerzos de investigación en niveles experimentales y evaluar los cambios donde sin habilidades previas de lógica y matemática, se pueda conseguir aplicación y utilidad desde el aprendizaje de aplicaciones computacionales, entornos de desarrollo programático, paquetes de software que permitan a los docentes y maestros desarrollar su enseñanza en forma paralela, tanto de sus contenidos académicos como de uso de lenguajes y actividades de programación computacional y, con esto, conseguir pensamiento computacional en las escuelas como actividades de rutina para habilidades en niños (Relkin et al., 2021).

En estudios con alumnos que han terminado sus estudios de educación básica regular, e iniciado su formación en niveles complementarios antes de la universidad, se ha logrado que los docentes impacten con propuestas de co-creación puesto que, en las aulas los estudiantes reciben conocimiento aplicado a contenidos y materias que son reguladas por sumillas, planificación académica y actividades de retroalimentación de su nivel. En este caso, el rol complementario del docente es ser un agente colaborativo, con trabajo funcional de liderazgo, motivador, director y guía, con recursos de programación computacional, que impliquen y afecten significativamente en la formación de sus estudiantes, con la incrustación de contenido multimedia en los proyectos que realizan, como por ejemplo las pruebas de comunicación P2P (comunicación punto a punto), lo que promueve crear y diseñar redes de comunicación en las actividades con resultados de los estudiantes, evaluados por eficiencia en su comunicación y con alcances diferentes, pues se apoyan en la metodología del líder docente, con índices y filtros en la información de apoyo, tutoriales, recursos de práctica y autoevaluación, guías y soluciones.

Por lo expuesto anteriormente, podemos decir que las herramientas analíticas integradas permiten evaluaciones no solamente colectivas, sino que van permitiendo acciones visibles de los estudiantes como colaboración, también desarrollos individuales, con evaluación del progreso y aplicación controlada del pensamiento computacional, puesto en ejecución por los estudiantes que construyen su propio conocimiento académico, en un proceso descrito por la teoría pedagógica del constructivismo (Papert & Harel, 1991), en la que los estudiantes desarrollan su conocimiento mediante la construcción de un 'artefacto' que los motive, y más importante aún, pueden compartir esa "construcción motivadora".

Para la construcción de conocimientos, en niveles universitarios, para el desarrollo de cursos que implican lógica, matemática, diseño de decisiones en entornos de datos para ingeniería, se utilizan entornos de programación computacional y lógica como actividades prácticas, los estudiantes pueden experimentar con los contenidos conceptuales desarrollados por el docente, los materiales y guías audiovisuales (videos), además los estudiantes diseñan soluciones generalizadas y pueden construir, con planificación grupal, y aportes individuales, un proyecto para resolver una situación o un problema propuesto (More Valencia, 2018). Por lo que la forma de integración de contenidos con actividades de pensamiento lógico, se ejecuta desde los principios de un lenguaje de programación para computadoras, y comunicarse lógicamente para crear ideas de resolución de problemas (Disessa, 2004).

2. Revisión de la literatura

En el ámbito académico el pensamiento computacional está relacionado con el conocimiento computacional, por un pensamiento lineal para desarrollar tareas y procesos, esto quiere decir, programación, que permite en un período sostenido, actividades académicas específicas que se centran, en llevar a la práctica reflexiva, soluciones a casos y actividades dentro del ámbito académico, la reflexión implica ser eficientes en mayor frecuencia con los cursos que utilizan tecnologías, aplicaciones informáticas y paquetes de software. Las transformaciones y cambios, que en principio son competencias de los maestros o docentes, que por capacidades, se les exige la dependencia exclusiva y extraordinaria de medidas auto informadas, puesto que antes del conocimiento computacional se requiere una estructura de pensamiento el cual se diferencia, para usar la linealidad de una causa y efecto, tareas predeterminadas y fin decisor o resultado funcional, como proceso para desarrollar el enfrentamiento a problemas (S.-C. Kong et al., 2020).

La autorregulación es importante para las habilidades del pensamiento computacional, ya que se consideran un factor o constructo que permite la idea de la persistencia selectiva, es decir, que aunque tendrían metas o indicadores basados en procesos, pueden formalizar una búsqueda de procesos no agregados, de tal manera que revelen medidas que parten del aprendizaje del estudiante para que utilice la programación computacional cuando recopila un flujo de datos; que luego procesa de forma codificada de tal manera que pueda evaluarse por partes y de forma continua en el progreso del estudiante en sus estudios de los cursos de ingeniería. La explicación de los resultados permite que los docentes o profesores entiendan que, aunque se tiene un proceso lineal de pensamiento, este puede ser selectivo y discriminatorio sin dejar de tener alternativas de solución que puedan ser evaluadas para el desarrollo del conocimiento computacional de los estudiantes (Chan et al., 2021).

Los estudiantes deben comprender que están usando un tipo de pensamiento y distinguirlo de otros, es decir, qué lugar ocupa la tecnología y la abstracción como principio inicial de usar la programación para discriminar y resolver. El estudiante de ingeniería, debe sentirse cómodo para aplicar a problemas de cursos de nivel universitario, el aprendizaje de conceptos informáticos, también debe tener interés en ejecutar sus habilidades para resolver problemas con el apoyo computacional, pero el interés de utilizar, tanto la problemática propuesta, como la información, requiere de una forma de pensamiento útil, eficiente y de interés, donde el estudiante amplíe y tome oportunidades para usar algoritmos y programación. En cualquier clase se puede incorporar dentro de los planes silábicos, la lógica programática y el uso del pensamiento computacional, donde el estudiante resuelve por análisis de un dominio, el problema de forma algorítmica matemática, el uso de algún tipo de lenguaje de programación y que puede pasar de la idea abstracta a modelar un dominio, esto es no olvidar el marco conceptual, el contexto y la implementación, que permita ser evaluado y además cotejado con el contenido teórico y práctico de las materias estudiadas, esta evaluación se puede generar para dar las soluciones a lo planteado (Leonard et al., 2018; Yadav et al., 2014).

El estudiante puede resolver por abstracción, un problema, caso o situación como procesos principales y luego modular o especificar, esto también es parte de la matemática como ciencia. Este proceso cognitivo, que permite obtener entradas y salidas, como si de un sistema de pruebas y depuración se tratara, los resultados irán a la aplicación de problemas, lo cual permite un orden y una lógica que ha sido discriminada y utilizada por el estudiante (S. C. Kong & Wang, 2021; S.-C. Kong et al., 2020), además las carreras profesionales y su tipificación pueden ayudar a tener una utilidad mejorada de lo que es el pensamiento computacional. Como parte de su desempeño el estudiante en su carrera puede entender que el logro de objetivos profesionales también implica, habilidades computacionales e informáticas y es valioso en sí mismo como conocimiento previo y comprendido, para la solución de problemas que se pueden enfrentar en la ejecución de su carrera profesional o en la vida diaria, lo que implicaría el desarrollo de competencias profesionales.

El estudio de Kong & Wang (2020), sobre identidad computacional a través del desarrollo de perspectivas de pensamiento computacional, considera las experiencias en edades tempranas. Las investigaciones tienen como propósito que herramientas, instrumentos y análisis documental validados, puedan medir lo que teóricamente se estructura, cómo emergen competencias y la utilidad necesaria de medios directos, que se sugieren para el uso de las tecnologías de información, comunicación y el pensamiento computacional, los que servirán como instrumentos o líneas de funcionalidad eficiente de problemas resueltos y también aquellas que se evalúan por estándares de la psicología, como observación y medición por psicometría. Las perspectivas en las dimensiones de capacidad de los estudiantes son para relacionar las herramientas que tiene para enfrentar deberes, como ejercicios, cuestionar la idea, así como la participación, la afiliación y actualización para objetivos con soluciones programáticas. Las acciones anteriores permiten que se tengan factores que convergen en el pensamiento computacional, lo cual apoya entonces el trabajo por el carácter distintivo asumido por estudiantes a las variaciones entre tecnología y pensamiento computacional, el pensamiento de los estudiantes logra distinguir y compara, ideas iniciales de entrada para enfrentar problemas y ejercicios resueltos básicos que apoyan soluciones y luego trabaja con los que están relacionados, esto es que debe modelarlas como formas distintas, pues si se fusionaran el uso de tecnologías con pensamiento computacional, no tendrían un modelo que permita revisar su influencia en la formación de la inteligencia y el conocimiento humano propio, que se tiene con la programación computacional.

La relación de la pedagogía con conceptos y práctica para la programación computacional en los docentes lo estudia Willermark. La revisión teórica de su estudio indica que el docente debe lograr articular el diseño y los resultados para que el aprendizaje sea significativo, esto es que utilice la comprensión del concepto dentro de cada caso o ejemplo, como parte del contenido silábico, lo que contribuye a un conocimiento de los cursos como enseñanza eficaz y esta es mejorada con el uso de la tecnología. En ese contexto, el pensamiento computacional se relaciona con la parte programática que representa el conocimiento como concepto, y la práctica del pensamiento computacional que es resolutoria, esto indica que tiene un contexto muy parecido a la realidad.

Es una parte del proceso de enseñanza aprendizaje, donde la algoritmia o las partes de solución dentro de un proceso de pensamiento lineal, se pueden enfrentar tanto de forma individual como colectiva, sin embargo no se debe olvidar el conocimiento pedagógico y tecnológico que debe tener el docente, por lo tanto la relación de estos tres elementos en el proceso, comprensión y definición del concepto problema, programación de tareas para

el contexto y práctica resolutoria, representan un conocimiento de la enseñanza del pensamiento computacional en relación a una problemática que puede tratarse sin el uso de tecnología o computador, y como actividades desconectadas en la enseñanza con comunicación remota y no presencial en las clases universitarias, tal como se dio por pandemia, podría entonces establecerse que las actividades donde no están en forma presencial los estudiantes, puedan trabajar y dedicarse a tareas que le permitan desarrollar actividades desconectadas y luego comparadas de forma grupal (Willermark, 2018).

Un estudio del autor Tikva, sobre la sistematización de la enseñanza y el aprendizaje del Pensamiento Computacional a través de la programación en la educación superior, mostró que la identificación de factores y áreas de estudio del pensamiento computacional como el desarrollo de capacidades, en estudiantes de nivel superior y de investigaciones longitudinales, ha surgido recientemente. Los cursos a nivel superior deben proponer un rediseño para alinear el conocimiento disciplinar y tener pensamiento de historias de análisis, correspondencia de ideas si los casos de aprendizaje son grupales, así como la aceptación de lo que se usa para enfrentar problemas y situaciones, con el fin de proporcionar modelos de instrucción.

Proponer marcos para estrategias de aprendizaje, herramientas y métodos de evaluación no se discute ampliamente, el pensamiento computacional se implementa a través de entornos de programación de texto, o de consola. Pero hay una tendencia ascendente en la programación visual, dado que se aplica a los cursos de formación de docentes, que han estado a la vanguardia de la educación superior con pensamiento computacional en los últimos años. Además, recientemente han surgido herramientas y recursos para los docentes como microcontroladores, para robótica y sistemas de realidad aumentada, utilizados como recursos del pensamiento computacional (Tikva & Tambouris, 2021).

3. Objetivos

Hay funciones a ejecutar en la práctica académica apoyada con el pensamiento computacional, que incluyen categorización, toma de decisiones, abstracción, conocimiento, resolución de problemas, planificación y ejecución, estas actividades abren para el alumno, la oportunidad de aprender e inspirarse en nuevas soluciones para problemas futuros (Lu & Law, 2012).

Se debe experimentar con niveles de eficiencia, las capacidades de estudiantes de ingeniería, desde diferentes aspectos en los que el pensamiento computacional, y que el docente influya con su material didáctico y un trabajo que supere a la práctica docente tradicional, como son mecanismos de colaboración, de autoevaluación, y de niveles de progreso para solucionar problemas como conocimiento y valorar las soluciones, el reconocimiento es importante. Los juegos se han utilizado para facilitar el aprendizaje introductorio a herramientas computacionales y de programación, los conceptos introductorios son claves para poder mapear y programar en computadoras, ya que las características del pensamiento computacional a través de una lógica de juegos como experiencias, proporcionan comentarios sobre el uso y utilidad de enfoques de ayuda y colaboración, además de habilidades para solucionar casos y problemas, es aquí que se puede tener objetivos más amplios como que se puede retroalimentar desde la evidencia empírica estructurada cuando se enfoca en casos y juegos (Kazimoglu et al., 2012).

El estudio tiene como objetivo evaluar métodos para utilizar el pensamiento computacional en estudiantes de ingeniería, evaluar sus factores para enfocar actividades en el proceso de enseñanza y aprendizaje con soporte programático, utilizando un entorno de desarrollo integral (IDE), que se usa en lenguajes compilados por el motor computacional como el paquete de software R, en cursos de simulación de sistemas y juego de negocios para ingeniería.

4. Método y Aplicación

En el marco de desarrollo de pruebas de diseño centrado en la evidencia (ECD), los resultados del estudio muestran que la conceptualización es en cuatro dimensiones de las prácticas de pensamiento computacional: (1) reutilización y remezcla, por el hecho de contar con librerías en el software funcional R y el entorno de desarrollo de programación computacional y algorítmica Rstudio, lo que genera una base de contexto para reutilizar, ya que las librerías con código programático, son compiladas en el entorno y son reutilizadas; (2) abstracción y popularización, este proceso sí implica un contexto de caso, donde se es aplicado, en tiempo en que la enseñanza es en línea, la narrativa del caso debe ser específica, por ello la práctica debe señalar de forma específica acciones y objetivos secuenciales; (3) prueba y depuración, permite en los ámbitos de cursos de ingeniería; la simulación además de realizar las pruebas, incluyendo en su depuración, un proceso comparativo y (4) pensamiento algorítmico, que permite la secuencialidad del proceso resolutorio, planificado y estructural del plan de acciones tanto de abstracción, programática, depuración y consecuencia por logro de objetivo. Esta evidencia teórica y empírica respalda la aplicabilidad de la definición operativa para las prácticas del pensamiento computacional en el contexto de la programación visual (S. C. Kong & Wang, 2020).

Para nuestro estudio se aplicó la siguiente metodología que consta de proponer problemas, casos y situaciones, evaluar los factores del pensamiento computacional y relacionar los resultados con el uso individual y colectivo

del pensamiento computacional en 45 estudiantes de los cursos de ingeniería para simulación de sistemas y juego de negocios. Por tanto, los resultados se evalúan durante las propuestas de casos de los cursos de ingeniería, además considerando una medición de las habilidades relacionadas al pensamiento computacional:

- A. Propuesta de caso contextual – Problema práctico académico.
- B. Dimensiones y niveles de pensamiento computacional evaluados.
- C. Soluciones presentadas por los estudiantes de forma individual y grupal.

La aplicación resultante se ejemplifica en el presente artículo, con un caso de ejemplo presentado y los objetivos que se les mostró a los estudiantes, así como el abordaje con el pensamiento computacional.

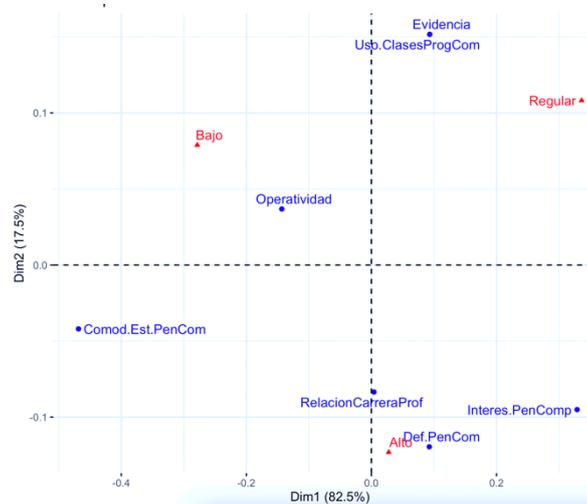
5. Resultados

El estudio propuso métodos para utilizar el pensamiento computacional en estudiantes de ingeniería, se logró evaluar los factores dentro del proceso de enseñanza y aprendizaje de estudiantes de ingeniería con programación funcional, reutilizable y compilada con el software de programación, en cursos de simulación de sistemas y juego de negocios. Se analizaron en prácticas y actividades resoluciones por abstracción, como proceso principal y modular, logrando una depuración de los casos expuestos y resueltos, a la aplicación de problemas con orden y una lógica discriminada y utilizada por el estudiante.

La aplicación del pensamiento computacional con herramientas analíticas integradas, con trabajo colaborativo para la resolución de problemas y casos específicos, generaron evidencia de desarrollos individuales y progreso controlable, así como la indicación que se usó el enfoque del construccionismo, ya que no solo parte de una aplicación de casos aplicados, sino también de la satisfacción del desarrollo de aspectos de ingeniería empresarial y experimentación de una dinámica de sistemas.

En la figura 1, el mapa dimensional definido y explicado por dos dimensiones de los datos estudiados, para la dimensión 1 que explica el 82.5 % de variabilidad y la dimensión 2 que explica el 17.5 % de variabilidad, representa para niveles altos de evaluación, los tres momentos secuenciales, en que los estudiantes desarrollan el pensamiento computacional, la forma es resultante en la teoría estudiada, el interés por usar la lógica, matemática y la computación, la definición como característica del proceso conceptual que especifica el diseño estratégico y de aplicación, convirtiéndose como parte académica del plan de sílabos de la carrera profesional.

Figura 1. Dimensión de Factores Evaluados del Pensamiento Computacional



Fuente: elaboración propia.

El problema real es específico, la adopción de la forma operativa, que permite revisar y la de aplicar el pensamiento computacional pues son de niveles bajos en el estudio, pero puede superarse las limitaciones específicas, al anotar en las clases prácticas, ciertos niveles de programación especializada en casos, aplicación y evidencia, en hechos y un contexto realizado, estudiado y analizado.

En la tabla 1, para los niveles de evaluación de los factores aplicados en estudiantes de ingeniería, la regularidad de la comodidad de los estudiantes para con el pensamiento computacional tiende a valores de nivel bajos como es el 51.1 %, y el de operatividad con el 40 %, entonces debe aprovecharse con los estudiantes el proceso mostrado dimensionalmente lo que implica disponer en los materiales y guías, propósitos conceptuales, las definiciones que tiene el estudiante sobre utilidad del pensamiento computacional y el interés que pudo medir con el 48.9 %, .Esto se aprecia por una secuencialidad inicial y de generación de aplicación, donde los estudiantes la aprovechan y utilizan en los casos presentados en la práctica académica.

Tabla 1. Niveles para Factores Evaluados del Pensamiento Computacional

	Alto	Regular	Bajo
Comodidad	42.2%	6.7%	51.1%
Definiciones	48.9%	24.4%	26.7%
Evidencia	35.6%	31.1%	33.3%
Interés	48.9%	33.3%	17.8%
Operatividad	40.0%	20.0%	40.0%
Relación Profesión	46.7%	22.2%	31.1%
Uso Académico	35.6%	31.1%	33.3%

Fuente: elaboración propia

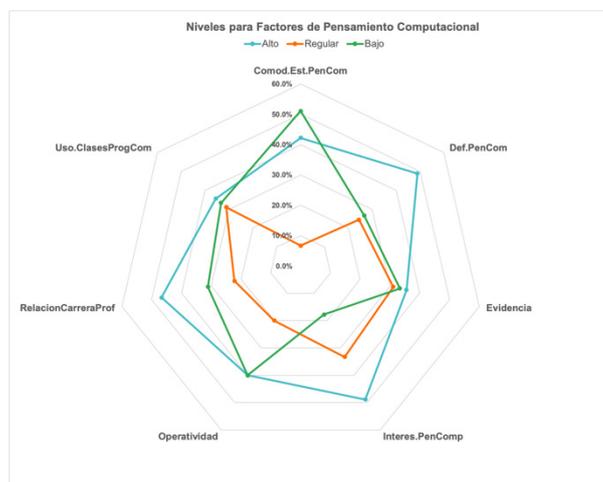
El análisis de los resultados motivó, una estructura secuencial de trabajo, esto es tener una organización general de los elementos del sistema cognitivo que utiliza el pensamiento computacional, aunque no había una base de datos, los estudiante tenían la guía propositiva y pudieron planificar por lógica de simulación de sistemas una matriz de datos, lo suficiente para permitir en un caso, menciona por ejemplo un caso de 1033 registros, que requería e involucraba aleatoriedad computacional demostrada, luego lograron la búsqueda de librerías y secuencia programática basada en semillas de números pseudoaleatorios en paquete de software R, que establezcan variabilidad inicial de un proceso estocástico, estas tareas fueron presentadas como soluciones individuales al caso expuesto. Aquí no solo es tener un ejemplo de aplicación, es también conocer y delimitar funciones, aplicadas a procesos de simulación.

Ya en las soluciones finales para el grupo, es tener por lo menos contextos diferentes que permitan la comparación, esto permite revisar otros métodos, como se observa en las soluciones explicadas por los grupos de trabajo, la toma de decisiones es el caso terminado que proviene de una explicación teórica y práctica revisada, y además propone que esta tenga base matemática y de estadística. Es interesante que los grupos se preocuparon en cada caso, por una condición de verdad y falsedad, que es lógica, pero por conteo se apoya en tener una comparativa numérica.

En la figura 2, los niveles mostrados al evaluar el pensamiento computacional, permiten establecer el criterio de convergencia de los niveles en los factores, se observa que los resultados convergen en la operatividad donde los estudiantes, no comparten un criterio establecido, esto es contraponen niveles bajos y altos de la operatividad, los niveles altos provienen de carreras profesionales ligadas a uso lógico, algorítmico y sistémico, mientras que los niveles bajos se deben gestionar en donde se implementan casos de práctica para procesos y automatización.

Mientras que la convergencia en evidencia, muestra que regularmente se toma casos que se operan y dan una evidencia de contexto conocido, donde los estudiantes se sienten cómodos, pues reconocen la realidad en una exposición de caso aplicado, por lo que es preciso anotar en este análisis de los resultados, que regularmente se tiene que usar en clase de ingeniería el pensamiento computacional, al tener regulares niveles con poca variabilidad y entender que puede encontrarse utilidad en el impacto del pensamiento computacional.

Figura 2. Niveles para Factores Evaluados del Pensamiento Computacional



Fuente: elaboración propia

Se observa por la experiencia, que los docentes pueden integrar y reutilizar con la programática, computaciones complejas, como simulación de datos y organización del aprendizaje, centrado en individualidad, pues los estudiantes lograron crear una base de datos simulados con registros al caso expuesto, esto agencia también la participación fuera del contexto (aula física), aunque no sea exitosa la experiencia individual, la comunicación e integración en una segunda parte de la experiencia práctica promueve retroalimentación y dominio, en el ejemplo mostrado, los estudiantes mencionan, usar más de un método (Tucker-Raymond et al., 2021). Existe entonces regularidad de la comodidad de los estudiantes para con el pensamiento computacional que tiende a valores de nivel bajos como es el 51.1 %, y el de operatividad con el 40 %. Se utilizó entonces con los estudiantes un proceso mostrado dimensionalmente con propósitos conceptuales, definición y el interés con el 48.9 %, por una secuencialidad inicial y de generación de aplicación, que la aprovechan y utilizan en los casos presentados en la práctica académica.

6. Discusión y Conclusiones

Se ha demostrado que la aplicación de pensamiento computacional con herramientas analíticas integradas permite evaluaciones no solo colectivas, como colaboración para problemas y casos específicos, sino también hay evidencia de desarrollos individuales y progreso controlable, para detalles de los casos expuestos como estudio. Asimismo, el construccionismo es propuesto no solo como parte de una aplicación organizacional o reglamentada por los materiales y guías de un ente estatal o particular que direcciona como hacerlo, sino que se establecen casos aplicados, inherentes a la realidad social y laboral más cercana, que los estudiantes de ingeniería conocen y deben resolver, que puedan simularse, esto es experimentar una dinámica de sistemas, para los casos planteados en el estudio.

La práctica recurre a un entorno de programación (software de programación, diseño funcional y ejecución), por tanto, se cumple tener entornos de programación práctica, donde los alumnos experimentan. Esta explicación es guiada por el docente que aprecia y valora su realidad, por problemas aplicados, en el contexto del estudio era una viabilidad económica, en un negocio que da mucho trabajo a personas en la comunidad, y la determinación que es establecida también por el docente como creador y co autor de las soluciones junto a sus estudiantes, y es evaluada con los contenidos conceptuales desarrollados, en materiales y guías visuales como video, además diseñan y construyen no solo un modelo para un proyecto, pues también se modela diversas soluciones para una situación o un problema propuesto, desde ideas individuales divergentes pero convergentes por apreciación y valoración medible a nivel de grupo. Y la forma de integración de contenidos se ejecuta desde los principios de un lenguaje de programación para computadoras, y logra comunicarse lógicamente, como secuencias de tareas medidas y descartadas para crear ideas de resolución de problemas.

Las habilidades para el pensamiento computacional son propuestas como pensamiento algorítmico, reconocimiento de patrones, depuración y simulación, es un proceso adaptativo que demuestra, el reconocimiento de patrones y la habilidad de depuración, tal como si se tratara de un juego, la aplicación de casos en los cursos de simulación de sistemas y juego de negocios permite explicar la capacidad de los estudiantes para desarrollar estrategias repetibles y generalizadas, monitorear su pensamiento lineal y objetivo como un proceso de solución y detectar posibles errores en su lógica. En el aspecto individual, el método permite que los estudiantes enfrenten una situación contextualizada y de caso, para luego recibir retroalimentación, sugerencias y materiales de aprendizaje, de forma colectiva, por tanto esta metodología le permite conseguir estrategias reutilizables, así como monitorear sus soluciones y detectar errores relacionados con el pensamiento computacional (Hooshyar et al., 2021).

El énfasis de integrar el pensamiento computacional en los procesos de enseñanza de ingeniería, no solo se vuelve natural o adoptado por el uso de la tecnología, se debe comprender como experiencia efectiva desde pensamiento y cognición que utiliza procesos de abstracción cómo las ciencias matemáticas lo utilizan para detectar patrones de reconocimiento de la realidad o contexto que se les presenta en su ejercicio pre profesional (Lei et al., 2020).

La comodidad de los estudiantes para utilizar el pensamiento computacional debe lograrse para mejora, con equipos de trabajo computacionales que tengan suficiencia para soportar miles de operaciones, proporcionados por las entidades académicas. Por esto desde la planificación de clases debe entenderse que sin herramientas computacionales, el pensamiento lógico que es propio del ser humano, no puede adaptarse en los casos que se enfrentan estudiantes de ingeniería aplicada, la rapidez de cálculo, desde una conceptualización de ideas abstractas y luego secuenciales así como de decisión, requieren de ser analizados por cada institución, es lógico por tanto la presteza del estudiante de usar su pensamiento lógico y programar computacionalmente una secuencia, pero en la mitad de las veces en la realidad del estudio requiere de comodidad para desarrollar soluciones.

Se pueden utilizar diversidad de formas para configurar la experiencia en el aula, de tal manera que no solo sea individualizada, sino más bien incorporar sistemas de aprendizaje con tareas y actividades que diferencien lo individual y colectivo, pero con decisiones retroalimentadas por comparativa y alternativas medidas y analizadas, que pueda vincular las experiencias de estudios anteriores donde, desde edades tempranas ,se materializa el

pensamiento computacional y el rendimiento práctico académico de estudiantes universitarios de carreras de ingeniería.

7. Agradecimientos

El presente texto nace de la experiencia en docencia universitaria, durante los años 2020 y 2021, como investigación resuelta para la Universidad Nacional de Piura y del apoyo de docentes del Departamento de Ingeniería y Ciencias, así como el soporte del Rectorado de la Universidad Nacional de Piura.

Referencias

- Chan, S.-W., Looi, C.-K., Ho, W. K., Huang, W., Seow, P., & Wu, L. (2021). Learning number patterns through computational thinking activities: A Rasch model analysis. *Heliyon*, 7(9), e07922. <https://doi.org/10.1016/j.heliyon.2021.e07922>
- Disessa, A. A. (2004). Metarepresentation: Native competence and targets for instruction. *Cognition and instruction*, 22(3), 293-331.
- Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M., & Lim, H. (2021). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior*, 114, 106575. <https://doi.org/10.1016/j.chb.2020.106575>
- Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming. *Procedia - Social and Behavioral Sciences*, 47, 1991-1999. <https://doi.org/10.1016/j.sbspro.2012.06.938>
- Kong, S. C., & Wang, Y. Q. (2020). Formation of computational identity through computational thinking perspectives development in programming learning: A mediation analysis among primary school students. *Computers in Human Behavior*, 106, 106230. <https://doi.org/10.1016/j.chb.2019.106230>
- Kong, S. C., & Wang, Y. Q. (2021). Item response analysis of computational thinking practices: Test characteristics and students' learning abilities in visual programming contexts. *Computers in Human Behavior*, 122, 106836. <https://doi.org/10.1016/j.chb.2021.106836>
- Kong, S.-C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872. <https://doi.org/10.1016/j.compedu.2020.103872>
- Lei, H., Chiu, M. M., Li, F., Wang, X., & Geng, Y. (2020). Computational thinking and academic achievement: A meta-analysis among students. *Children and Youth Services Review*, 118, 105439. <https://doi.org/10.1016/j.childyouth.2020.105439>
- Leonard, J., Mitchell, M., Barnes-Johnson, J., Unertl, A., Outka-Hill, J., Robinson, R., & Hester-Croff, C. (2018). Preparing Teachers to Engage Rural Students in Computational Thinking Through Robotics, Game Design, and Culturally Responsive Teaching. *Journal of Teacher Education*, 69(4), 386-407. <https://doi.org/10.1177/0022487117732317>
- Lu, J., & Law, N. (2012). Online peer assessment: Effects of cognitive and affective feedback. *Instructional Science*, 40(2), 257-275. <https://doi.org/10.1007/s11251-011-9177-2>
- Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational Thinking and scratch at distance. *Computers in Human Behavior*, 80, 470-477. <https://doi.org/10.1016/j.chb.2017.09.025>
- More, R. A. (2018). Engineering techniques and methods for the process of teaching and learning in Information and Communication Technologies using social video in students of the University of Piura. 2, 139-143. Scopus.
- Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36(2), 1-11.
- Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 169, 104222. <https://doi.org/10.1016/j.compedu.2021.104222>
- Tikva, C., & Tambouris, E. (2021). A systematic mapping study on teaching and learning Computational Thinking through programming in higher education. *Thinking Skills and Creativity*, 41, 100849. <https://doi.org/10.1016/j.tsc.2021.100849>
- Tucker-Raymond, E., Cassidy, M., & Puttick, G. (2021). Science teachers can teach computational thinking through distributed expertise. *Computers & Education*, 173, 104284. <https://doi.org/10.1016/j.compedu.2021.104284>
- Willermark, S. (2018). Technological Pedagogical and Content Knowledge: A Review of Empirical Studies Published From 2011 to 2016. *Journal of Educational Computing Research*, 56(3), 315-343. <https://doi.org/10.1177/0735633117713114>
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, 14(1), 5:1-5:16. <https://doi.org/10.1145/2576872>