



Método Multicriterio Neutrosófico para la evaluación del impacto de Scratch: una herramienta de apoyo en el proceso de desarrollo de la lógica de programación.

Neutrosophic Multicriteria Method for Scratch impact assessment: a support tool in the programming logic development process.

Jorge Lenin Acosta Espinoza ¹, Rita Azucena Díaz Vásquez ², Andrés Roberto León Yacelga ³ and Karla Abigail Ayala Díaz ⁴

¹ Universidad Regional Autónoma de Los Andes, Ibarra. Ecuador. E-mail: ui.jorgeacosta@uniandes.edu.ec

² Universidad Regional Autónoma de Los Andes, Ibarra. Ecuador. E-mail: ui.ritadiaz@uniandes.edu.ec

³ Universidad Regional Autónoma de Los Andes, Ibarra. Ecuador. E-mail: ui.andresleon@uniandes.edu.ec

⁴ Universidad Regional Autónoma de Los Andes, Ibarra. Ecuador. E-mail: karlaad47@uniandes.edu.ec

Resumen. El conocimiento de la programación se ha convertido en una habilidad fundamental. Sin embargo, para los estudiantes que recién se están aventurando en este emocionante campo, el lenguaje de programación puede parecer intimidante y complejo. Es aquí donde Scratch entra en juego como una herramienta poderosa y amigable para iniciar el proceso de aprendizaje en la lógica de programación. Scratch, desarrollado por el MIT Media Lab, es un lenguaje de programación visual diseñado especialmente para jóvenes y principiantes. Su interfaz intuitiva utiliza bloques de código que se arrastran y sueltan para crear programas, eliminando la necesidad de aprender la sintaxis complicada que a menudo disuade a los recién llegados. Esta característica hace que Scratch sea una opción ideal para que los estudiantes exploren los conceptos fundamentales de la programación de manera lúdica y sin frustraciones. La presente investigación propone el desarrollo de un método multicriterio neutrosófico para la evaluación del impacto de Scratch. El método propuesto basa su funcionamiento mediante un enfoque multicriterio para la evaluación. La propuesta favorece la recomendación de su uso e implementación de Scratch. Entre los resultados más relevantes es que en corto tiempo los estudiantes en su mayoría lograron crear animaciones, juegos, historias y más al combinar bloques que representan diferentes acciones y lógica. En conclusión, Scratch se presenta como un pilar en el proceso de aprendizaje de la lógica de programación para estudiantes. Su enfoque en la accesibilidad, la creatividad y la comunidad brinda a los principiantes una introducción suave y atractiva al mundo de la programación.

Palabras Claves: método multicriterio neutrosófico, lógica de programación, lenguaje de programación, scratch.

Summary. Knowledge of programming has become a fundamental skill. However, for students who are just venturing into this exciting field, the programming language can seem intimidating and complex. This is where Scratch comes into play as a powerful and friendly tool to start the learning process in programming logic. Scratch, developed by the MIT Media Lab, is a visual programming language designed especially for young people and beginners. Its intuitive interface uses drag-and-drop blocks of code to create programs, eliminating the need to learn complicated syntax that often deters newcomers. This feature makes Scratch an ideal choice for students to explore the fundamental concepts of programming in a playful and frustration-free way. The present research proposes the development of a neutrosophic multicriteria method for evaluating the impact of Scratch. The proposed method bases its operation on a multi-criteria approach to evaluation. The proposal favors the recommendation of its use and implementation of Scratch. Among the most relevant results is that in a short time the students mostly managed to create animations, games, stories and more by combining blocks that represent different actions and logic. In conclusion, Scratch is presented as a pillar in the learning process of programming logic for students. Its focus on accessibility, creativity, and community gives beginners a gentle and engaging introduction to the world of programming.

Keywords: neutrosophic multicriteria method, programming logic, programming language, scratch.

1 Introducción

Ante la demanda de los sistemas informáticos de producción, servicios y consumo, particularmente en el presente mundo desarrollado, se presenta un hecho sorprendente: “no se pueden cubrir los puestos de trabajo de ingenieros de software, desarrolladores de aplicaciones, documentalistas digitales” [1-34-35], por falta de profesionales, de potenciales alumnos y sobre todo por carencia de personal capacitado [2-33].

En el mundo contemporáneo, la programación se ha convertido en una habilidad esencial que impulsa la innovación y la transformación en diversas industrias [3]. Sin embargo, más allá de la sintaxis y los lenguajes específicos, hay un pilar fundamental que sustenta todo el proceso de aprendizaje: la lógica de programación. Por lo que se analiza qué es la lógica de programación y cómo su comprensión impacta de manera significativa en el proceso de aprendizaje de estudiantes de los primeros niveles de las carreras de software primer semestre.

La lógica de programación es el arte de pensar de manera estructurada y secuencial para resolver problemas y crear algoritmos [4-36]. Abarca la habilidad de dividir un problema complejo en pasos más pequeños y manejables, para luego organizarlos de manera que conduzcan a una solución coherente y eficiente. Este enfoque analítico y ordenado es la base sobre la cual se construyen todos los programas informáticos, desde simples aplicaciones hasta sistemas complejos [5].

Para los estudiantes que se involucran en el mundo de la programación, comprender y dominar la lógica es esencial. A menudo, los lenguajes de programación pueden parecer abrumadores y confusos para los principiantes, pero la lógica subyacente es universal. Al aprender a pensar de manera lógica, los estudiantes adquieren una base sólida que les permite abordar cualquier lenguaje con confianza y destreza [6-37-38].

La lógica de programación también fomenta habilidades cruciales en la resolución de problemas. A través de la descomposición de problemas en partes más pequeñas y la identificación de patrones y relaciones, los estudiantes aprenden a abordar desafíos de manera sistemática y eficiente. Esta habilidad trasciende el ámbito de la programación, ya que se convierte en una herramienta valiosa en la vida cotidiana y en la toma de decisiones informadas [7-39].

A medida que los estudiantes principiantes se sumergen en la lógica de programación, también cultivan su creatividad y capacidad de abstracción. La resolución de problemas requiere pensar fuera de la caja y considerar múltiples enfoques, lo que a su vez nutre la innovación y la adaptabilidad. Estas habilidades son esenciales en un mundo en constante evolución, donde las soluciones originales y flexibles son altamente valoradas.

La lógica de programación sirve como cimiento fundamental en el proceso de aprendizaje de programación para estudiantes principiantes [6-40-53]. Al comprender y aplicar principios lógicos, los estudiantes no solo se vuelven competentes en la escritura de código, sino que también desarrollan habilidades cruciales en la resolución de problemas, el pensamiento creativo y la toma de decisiones informadas. Estos beneficios trascienden la programación y enriquecen el crecimiento personal y profesional de los estudiantes en un mundo cada vez más orientado hacia la tecnología.

El presente artículo propicia el uso de herramientas que sirvan de apoyo al proceso de enseñanza – aprendizaje, y sobre todo afianzar el desarrollo de la lógica de programación utilizando herramientas como Scratch, permitiendo a los estudiantes el desarrollo de sus habilidades en el análisis y resolución de problemas.

Scratch es un lenguaje de programación visual diseñado especialmente para enseñar a niños y principiantes los conceptos básicos de la programación de una manera lúdica e interactiva. Fue desarrollado por el Grupo Lifelong Kindergarten del MIT Media Lab. En lugar de escribir líneas de código tradicionales, los usuarios de Scratch manipulan bloques de programación en forma de iconos gráficos que representan diferentes comandos y acciones [8-41].

A continuación, se describen algunas características básicas de Scratch.

- 1 Interfaz gráfica: Scratch presenta una interfaz gráfica en la que los usuarios pueden arrastrar y soltar bloques de programación desde una barra lateral hacia el área de trabajo principal. Estos bloques encajan como piezas de un rompecabezas para construir programas.
- 2 Bloques de programación: Los bloques de programación en Scratch representan diferentes comandos y acciones que se pueden llevar a cabo en un programa. Hay bloques para control de flujo (como bucles y condicionales), bloques para manejo de eventos (como clics de ratón y pulsaciones de teclas), y bloques para operaciones matemáticas y lógicas, entre otros.
- 3 Programación por eventos: En Scratch, los programas se crean en base a eventos. Los eventos pueden ser acciones del usuario, como hacer clic en un objeto o presionar una tecla, o eventos de tiempo, como esperar cierta cantidad de segundos. Los bloques de código se pueden enlazar a estos

- eventos para que se ejecuten cuando ocurra el evento específico.
- 4 Bloques de código: Los bloques de código en Scratch están organizados en diferentes categorías, como movimiento, apariencia, sonido, eventos, control, entre otros. Cada bloque tiene una forma específica y encaja con otros bloques de manera lógica y estructurada.
 - 5 Arrastrar y soltar: Los usuarios pueden seleccionar los bloques de programación que necesitan y arrastrarlos hacia el área de trabajo. Los bloques se conectan entre sí de forma lógica para construir el flujo de la aplicación.
 - 6 Conexiones lógicas: Los bloques de programación tienen conectores magnéticos que permiten unirlos de manera lógica. Por ejemplo, puedes conectar un bloque de "si" (if) a un bloque de "entonces" (then) para establecer una condición y su acción correspondiente.
 - 7 Sprites y escenario: En Scratch, los programas se basan en personajes llamados "sprites". Los usuarios pueden crear sus propios sprites o elegir entre los proporcionados por Scratch. También pueden diseñar fondos o "escenarios" para sus programas.
 - 8 Eventos y acciones: Los programas en Scratch se basan en eventos y acciones. Los eventos son desencadenantes que hacen que ocurran ciertas acciones. Por ejemplo, hacer clic en un sprite puede ser un evento. Los bloques de programación se utilizan para definir cómo el sprite debe responder a un evento específico.
 - 9 Ejecución interactiva: A medida que los bloques de programación se conectan, el programa cobra vida en el área de trabajo. Los usuarios pueden probar su programa haciendo clic en los bloques de activación o simulando eventos como clics de ratón.
 - 10 Creación de animaciones e interacciones: Scratch es especialmente adecuado para la creación de animaciones interactivas, juegos y otras aplicaciones visuales. Los usuarios pueden combinar bloques para mover, girar, cambiar el tamaño y cambiar la apariencia de los sprites, lo que permite la creación de interacciones divertidas.
 - 11 Compartir proyectos: Una de las características destacadas de Scratch es la posibilidad de compartir los proyectos creados en línea. Los usuarios pueden cargar sus proyectos en la plataforma Scratch y permitir que otros los vean, jueguen con ellos y aprendan de ellos.

Las ventajas de la programación por bloques:

- La programación por bloques nos permite generar programas sin necesidad de escribir código y con lo cual es más difícil equivocarse y aprendemos más rápido.
- Además, los bloques están clasificados por categorías y colores que nos hacen mucho más intuitivo el código para poder entenderlo.
- Podemos ver justo al lado el resultado de nuestro código con lo cual a medida que avanzamos ya notamos los resultados.
- Adquirimos el pensamiento computacional de forma mucho más rápida que con otro tipo de lenguajes.
- Scratch plantea un modo diferente de enseñanza/aprendizaje, con Scratch el alumno es protagonista, necesita pensar, puede plantear preguntas y soluciones a esas preguntas. El aprendizaje es activo y constructivo (López-Escribano & Sánchez-Montoya, 2012).
- Facilita mucho la tarea para luego aprender otros lenguajes más complejos.

Teniendo en cuenta estas ventajas, la presente investigación tiene como objetivo desarrollar un Método Multicriterio Neutrosófico para la evaluación del impacto de Scratch.

2 Materiales y métodos

La presente sección describe el funcionamiento del método multicriterio neutrosófico para la evaluación del impacto de Scratch. Se presentan las características generales de la solución propuesta. Se describen las principales etapas y actividades que conforman el método.

El método multicriterio neutrosófico para la evaluación del impacto de Scratch está diseñado bajo las siguientes cualidades:

Las cualidades que distinguen al modelo son:

- Integración: el método garantiza la interconexión de los diferentes componentes en combinación para la evaluación del impacto de Scratch.
- Flexibilidad: utiliza 2-tuplas para representar la incertidumbre de modo que aumente la interoperabilidad de las personas que interactúan con el método.
- Interdependencia: el método utiliza como punto de partida los datos de entrada proporcionados por los expertos del proceso. Los resultados analizados contribuyen a una base de experiencia que conforma el núcleo del procesamiento para la inferencia.

El método se sustenta en los siguientes principios:

- Identificación mediante el equipo de expertos de los indicadores para la evaluación del impacto de Scratch.
- El empleo de métodos multicriterios en la evaluación.

El método para la evaluación del impacto de Scratch, está estructurado para gestionar el flujo de trabajo del proceso de evaluación a partir de un método de inferencia multicriterio, posee tres etapas fundamentales: entrada, procesamiento y salida de información. La Figura 1 muestra un esquema que ilustra el funcionamiento general del método.

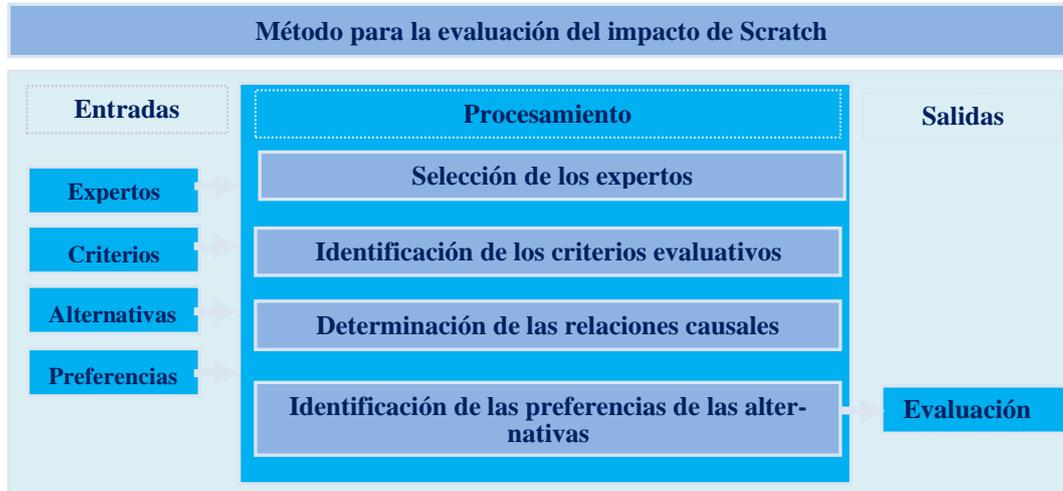


Figura 1. Esquema general del funcionamiento del método.

2.1 Descripción de las etapas del modelo

El método propuesto está diseñado para garantizar la gestión del flujo de trabajo en el proceso para la evaluación del impacto de Scratch. Utiliza un enfoque multicriterio multiexperto donde se identifican indicadores evaluativos para determinar el funcionamiento del procesamiento del método.

La etapa de procesamiento está estructurada por cuatro actividades que rigen el proceso de inferencia del procesamiento. A continuación se detalla su funcionamiento:

Actividad 1: Selección de los expertos.

La actividad consiste en determinar el grupo de expertos que intervienen en el proceso. Para su selección se emplea la metodología propuesta por Fernández [9-42-43]. Para comenzar el proceso se envía un modelo a los posibles expertos con una explicación breve sobre los objetivos del trabajo y el área del conocimiento en el que se enmarca la investigación. Se realizan las siguientes actividades:

1. Se establece contacto con los expertos conocedores y se les pide que participen en el panel. La actividad obtiene como resultado la captación del grupo de expertos que participará en la aplicación del método.

El proceso debe filtrar los expertos con bajo nivel de experticia, participando solamente los de mayor conocimiento y prestigio en el área del conocimiento que se enmarca el objeto de estudio de la investigación. Para realizar el proceso de filtraje se realiza un cuestionario de autoevaluación para expertos. El objetivo es determinar el coeficiente de conocimiento o información (K_c), la ecuación 1 expresa el método para determinar el nivel de experticia.

$$K_c = n(0,1) \quad (1)$$

Where:

K_c : coeficiente de conocimiento o información

n : rango seleccionado por el experto

Actividad 2 Identificación de los criterios de evaluación

Una vez identificados los expertos que intervienen en el proceso, se procede a la identificación de los criterios evaluativos. Los criterios nutren el método, representan parámetros de entrada que se utilizan en la etapa de procesamiento. A partir del trabajo en grupo de los expertos se realizan las siguientes actividades:

1. Se envía un cuestionario a los miembros del panel y se les pide su opinión para la selección de los criterios evaluativos que sustenten la investigación. A partir de un cuestionario previamente elaborado, se obtiene como resultado el conjunto de criterios de los expertos.
2. Se analizan las respuestas y se identifican las áreas en que están de acuerdo y en las que difieren. La actividad permite realizar un análisis del comportamiento de las respuestas emitidas por los expertos y se identifican los elementos comunes.
3. Se envía el análisis resumido de todas las respuestas a los miembros del panel, se les pide que llenen de nuevo el cuestionario y que den sus razones respecto a las opiniones en que difieren. La actividad permite obtener una nueva valoración del grupo de expertos sobre el conocimiento recogido y resumido.
4. Se repite el proceso hasta que se estabilizan las respuestas. La actividad representa la condición de parada del método, a partir de que se establezcan las respuestas se concluye su aplicación considerándose este el resultado general.

En la actividad se obtiene como resultado el conjunto de criterios evaluativos del método. Emplea un enfoque multicriterio expresado como muestra la ecuación 1.

$$C = \{c_1, c_2, \dots, c_m\} \quad (2)$$

Donde:

$$m > 1, \quad (3)$$

Actividad 3 Determinación de los pesos de los criterios.

Para determinar los pesos atribuidos a los criterios evaluativos se utiliza el grupo de expertos que intervienen en el proceso. Se les pide que determinen el nivel de importancia atribuido a los criterios evaluativos identificados en la actividad previa.

Los pesos de los criterios evaluativos son expresados mediante un dominio de valores difusos. Los conjuntos difusos dan un valor cuantitativo a cada elemento, el cual representa el grado de pertenencia al conjunto. Un conjunto difuso A es una aplicación de un conjunto referencial S en el intervalo [0, 1], Tal que:

$$A: S \rightarrow [0,1],$$

y se define por medio de una función de pertenencia:

$$0 \leq \mu_A(x) \leq 1. \quad (4)$$

Para aumentar la interpretatividad en la determinación de los vectores de pesos asociados a los criterios se utilizan términos lingüísticos basados en 2-tuplas Neutrosófica [10], [11]. El uso de etiquetas lingüísticas en modelos de decisión supone, en la mayoría de los casos, la realización de operaciones con etiquetas lingüísticas. La tabla 1 muestra el conjunto de términos lingüísticos con sus respectivos valores.

Tabla 1: Dominio de valores para expresar causalidad.

Término lingüístico	Números SVN
Extremadamente buena (EB)	[1,0,0]
Muy muy buena (MMB)	[0.9, 0.1, 0.1]
Muy buena (MB)	[0.8,0.15,0.20]
Buena (B)	[0.70,0.25,0.30]
Medianamente buena (MDB)	[0.60,0.35,0.40]
Media (M)	[0.50,0.50,0.50]
Medianamente mala (MDM)	[0.40,0.65,0.60]
Mala (MA)	[0.30,0.75,0.70]
Muy mala (MM)	[0.20,0.85,0.80]
Muy muy mala (MMM)	[0.10,0.90,0.90]
Extremadamente mala (EM)	[0,1,1]

Una vez obtenidos los vectores de pesos de los diferentes expertos que intervienen en el proceso se realiza un proceso de agregación de información a partir de una función promedio tal como muestra la ecuación 5.

$$VA = \frac{\sum_{i=1}^n C_{ij}}{E} \quad (5)$$

Where:

VA: valor agregado,

E: cantidad de expertos que participan en el proceso,

C_{ij} : vector de pesos expresado por los expertos para los criterios C .

Actividad 4 determinación de las preferencias de las alternativas.

La actividad para la determinación de las preferencias consiste en identificar la preferencia que poseen los criterios evaluativos para la evaluación del impacto de Scratch. El proceso de evaluación es realizado mediante una escala numérica de modo que se exprese el nivel de pertenencia de los indicadores. La figura 2 muestra una gráfica con los conjuntos de etiquetas lingüísticas utilizados.

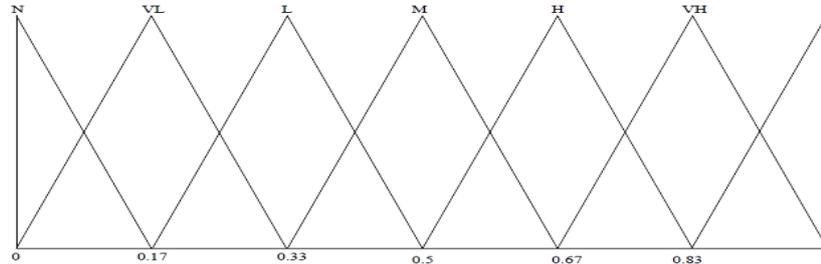


Figura 2. Conjunto de etiquetas lingüísticas.

Donde:

- N: Nulo
- VL: Muy Bajo
- L: Bajo
- M: Medio
- H: Alto
- VH: Muy Alto
- P: Preferido

Para la evaluación del impacto de Scratch, se describe el problema y la evaluación de cada alternativa a partir del cual se forma la matriz de evaluación [12-44-45], [13], [14]. La matriz está compuesta por las alternativas, los criterios y la valoración de cada criterio para cada alternativa [15], [16], [17].

A partir de obtener las preferencias de cada criterio evaluativo sobre el objeto de estudio, se realiza el proceso de inferencia de información [18], [19], [20]. La inferencia es guiada mediante el uso de operadores de agregación de información.

Se parte del conjunto de alternativas A :

$$A = \{A_1, A_2, \dots, A_m\} \quad (6)$$

A las cuales se les obtienen las preferencias P :

$$P = C_1, \dots, C_n \quad (7)$$

A los criterios evaluativos se les aplica un método multicriterio para procesar las alternativas a partir de los vectores de pesos W definidos por los expertos sobre los criterios evaluativos.

$$W = \{w_1, w_2, \dots, w_n\} \quad (8)$$

El proceso de agregación se realiza con la utilización de operadores de agregación de información [21], [22], [23]. El objetivo fundamental consiste en obtener valoraciones colectivas a partir de valoraciones individuales mediante el uso de operadores de agregación. Para el procesamiento del método propuesto se utiliza el operador de agregación OWA (*Ordered Weighted Averaging*) [24],[25], [26-46-47].

Los operadores OWA funcionan similar a los operadores media ponderada, aunque los valores que toman las variables se ordenan previamente de forma decreciente y, contrariamente a lo que ocurre en las medias ponderadas, los pesos no están asociados a ninguna variable en concreto [27], [28], [29].

Definición 1: Dado un vector de pesos $W = w_1, \dots, w_n \in [0,1]^n$ tal que: $\sum_{i=1}^n w_i = 1$, el operador (OWA) asociado a w es el operador de agregación $f_n^w: \rightarrow R$ definido por:

$$f_n^w(u) = \sum_{i=1}^n w_i v_i \quad (9)$$

donde v_i es el i -ésimo mayor elemento de $\{u_1, \dots, u_n\}$

Para la presente investigación se define el proceso de agregación de la información empleado, tal como expresa la ecuación 10.

$$F(p_1, p_2, \dots, p_n) = \sum_{j=1}^n w_j b_j \quad (10)$$

Donde:

P : conjunto de preferencias obtenidas de la evaluación de los criterios para la evaluación del impacto de Scratch.

w_j : son los vectores de pesos atribuidos a los criterios evaluativos.

b_j : es el j -ésimo más grande de las preferencias p_n ordenados.

3 Resultados y discusión

Para la implementación del método propuesto se ha realizado un estudio de caso donde se representa un instrumento enfocado hacia el caso específico que se modela. El objeto de análisis es la evaluación del impacto de Scratch como una herramienta de apoyo en el proceso de desarrollo de la lógica de programación. A continuación se presentan las valoraciones alcanzadas por cada actividad:

Actividad 1: Selección de los expertos.

Para la aplicación del método, se aplicó un cuestionario con el objetivo de seleccionar el grupo de expertos a intervenir en el proceso. Se logró el compromiso desinteresado de 9 expertos. Se les aplicó el cuestionario de autoevaluación a los 9 expertos donde se obtuvieron los siguientes resultados:

- 3 expertos se autoevalúan con un nivel de competencia sobre el tema objeto de estudio de 10 puntos.
- 2 expertos se autoevalúan con un nivel de competencia de 9 puntos.
- 2 expertos se autoevalúa con un nivel de competencia de 8 puntos.
- 2 expertos se autoevalúan con un nivel de competencia de 6 puntos.

El coeficiente de conocimiento K_c representa un parámetro importante en la aplicación del método propuesto. Para la investigación se obtienen los K_c por experto tal como refiere la tabla 3

Tabla :

Tabla 3. Coeficiente de conocimiento por expertos.

1	2	3	4	5	6	7	8	9
1	0,60	0,80	1	1	0,60	0,80	0,90	0,90

Se aplicaron 4 preguntas a los expertos donde se obtuvieron los siguientes resultados para identificar los niveles de conocimientos sobre el tema:

- Sobre la pregunta 1. Análisis teóricos realizados por usted sobre el tema: se obtuvo una autoevaluación de *Alta* para 5 expertos y *Media* para 2 expertos.
- Sobre la pregunta 2. Estudio de trabajos publicados por autores Ecuatorianos: se obtuvo una autoevaluación de *Alta* para 5 expertos, *Media* para 2 expertos y *Baja* para 2 expertos.
- Sobre la pregunta 3. Trabajo directo en el proceso de desarrollo de la lógica de programación: se obtuvo una autoevaluación de *Alta* para 5 expertos, *Media* para 3 expertos y *Baja* para 1 experto.
- Sobre la pregunta 4. Conocimiento del proceso de desarrollo de la lógica de programación: se obtuvo una autoevaluación de *Alta* para 4 expertos, *Media* para 4 expertos y *Baja* para 1 experto.

La figura 3 muestra una gráfica con el comportamiento de los coeficientes de conocimiento de los expertos. A partir del análisis de los resultados se determina utilizar 7 de los 9 expertos previstos inicialmente.

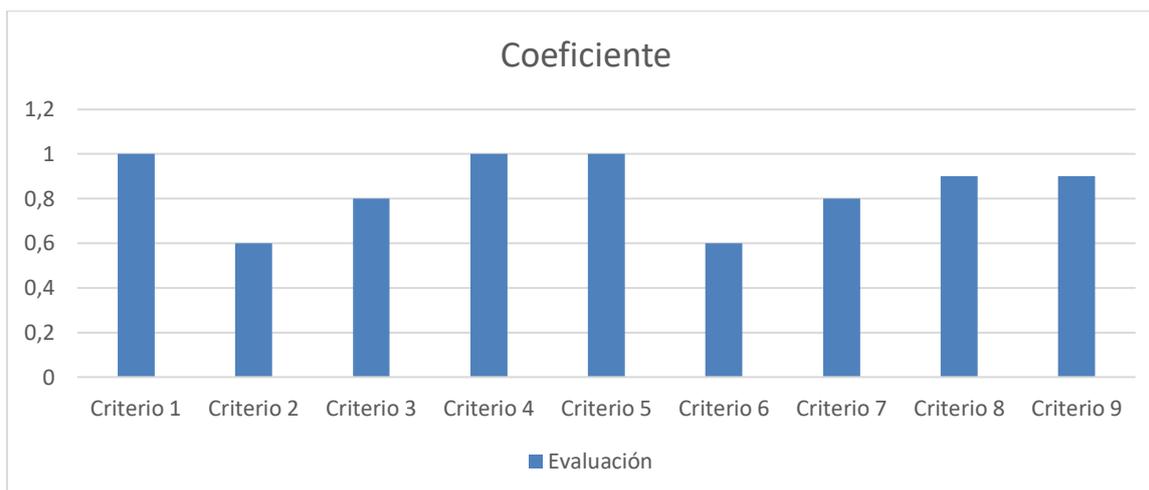


Figura 3. Representación del coeficiente de conocimiento de los expertos.

Actividad 2 Identificación de los criterios de evaluación

Para la actividad se realizó una encuesta a los expertos que intervienen en el proceso. El objetivo consistió en identificar los criterios para la evaluación del impacto de Scratch. Los indicadores constituyen el elemento fundamental sobre el cual se realiza el procesamiento en etapas siguientes.

La tabla 4 visualiza los criterios evaluativos obtenidos de la actividad.

Tabla 4: Criterios evaluativos obtenidos.

Número	Criterios evaluativos
C_1	1. Facilidad de uso: Evaluar la facilidad con la que los estudiantes pueden utilizar la herramienta para desarrollar la lógica de programación, considerando su interfaz, funciones y capacidades de personalización.
C_2	Flexibilidad: Analizar la capacidad de la herramienta para adaptarse a diferentes estilos y enfoques de desarrollo de la lógica de programación, permitiendo la implementación de diferentes algoritmos y estructuras de control.
C_3	Eficiencia en la resolución de problemas: Evaluar la capacidad de la herramienta para ayudar a los estudiantes a resolver problemas complejos mediante el desarrollo de algoritmos eficientes y lógicos.
C_4	Capacidad para enseñar: Evaluar si la herramienta es efectiva para enseñar conceptos clave de lógica de programación, como estructuras condicionales, bucles, recursividad, entre otros, de manera clara y comprensible.
C_5	Integración con otros recursos: Analizar la capacidad de la herramienta para integrarse con otros recursos educativos o de desarrollo, como tutoriales, documentación y ejemplos prácticos que enriquezcan el aprendizaje y la aplicación de la lógica de programación.
C_6	Retroalimentación y soporte: Evaluar si la herramienta proporciona retroalimentación efectiva a los estudiantes, como: mensajes de error claros, sugerencias para mejorar el código y soporte técnico para resolver dudas o problemas relacionados con el desarrollo de la lógica de programación.

Actividad 3 Determinación de los pesos de los criterios

Para determinar los pesos sobre los criterios se utilizó un enfoque multiexperto, en el que participaron los 7 seleccionados en la actividad 1. Con el empleo de 2-tuplas tal como propone la tabla 1 se realizó el trabajo por el grupo de expertos.

A partir de la agregación realizada mediante la ecuación 9 se unifica los pesos de los 7 expertos en un valor agregado. La tabla 5 muestra el resultado de los vectores de pesos resultantes de la actividad.

Tabla 5: Pesos de los criterios a partir del criterio de expertos.

Número	Vectores de pesos W para los criterios C
C_1	[0,9, 0,1, 0,1]
C_2	[1,0,0]
C_3	[0,75,0,25,0,30]
C_4	[1,0,0]
C_5	[0,9, 0,1, 0,1]
C_6	[0,9, 0,1, 0,1]

Se llegó al consenso en la segunda iteración del proceso. A partir de lo cual se tomó como valor de parada.

Actividad 4 determinación de las preferencias de las alternativas.

Para el estudio de caso propuesto con el objetivo de evaluar del impacto de Scratch, se realizó una evaluación del cumplimiento de los criterios. Se tomó como información de partida los vectores de pesos atribuidos a cada criterio evaluativo. Se evaluó el cumplimiento de los indicadores con el empleo del conjunto de etiquetas lingüísticas. Se obtuvo como resultado un sistema con valores difusos que se agregan como valores de salidas. La tabla 6 muestra el resultado del procesamiento realizado.

Tabla 6: Resultado de las evaluaciones obtenidas por los expertos

Número	W	Preferencia	$\sum w_j b_j$
C_1	[0,9, 0,1, 0,1]	[0,9, 0,1, 0,1]	[0,9, 0,1, 0,1]
C_2	[1,0,0]	[1,0,0]	[1,0,0]
C_3	[0,75,0,25,0,30]	[1,0,0]	[0,87, 0,1, 0,1]
C_4	[1,0,0]	[0,70,0,25,0,30]	[0,85, 0,1, 0,1]
C_5	[0,9, 0,1, 0,1]	[1,0,0]	[0,95, 0,1, 0,1]
C_6	[0,9, 0,1, 0,1]	[1,0,0]	[0,95, 0,1, 0,1]
Índice			[0,92, 15,0,20]

La figura 4 muestra el comportamiento de las inferencias sobre los criterios evaluativos para el caso de estudio propuesto.

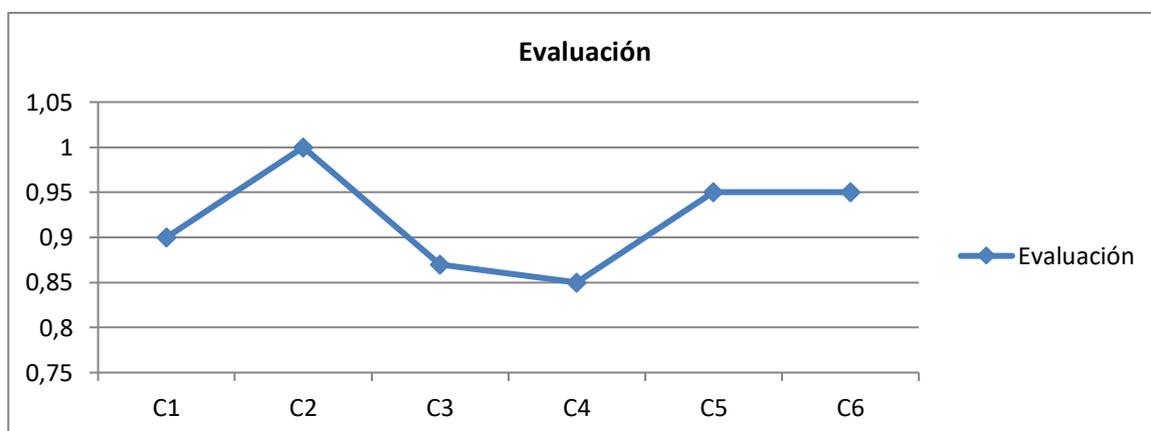


Figura 4. Comportamiento de las inferencias.

A partir de los datos presentados en la tabla 5, se identifica un índice de la propuesta de método multicriterio neutrosófico para la evaluación del impacto de Scratch con un II 0,92. Los resultados obtenidos son valorados como un Alto índice de impacto.

4 Aplicación de un instrumento investigativo

Adicionalmente se aplicó una encuesta de conocimiento del nivel de desarrollo en la lógica de programación en base a un producto elaborado en Scratch. La población tomada en cuenta para esta investigación fueron los 30 estudiantes activos de primer semestre de la carrera de software modalidad en línea de la Universidad Regional Autónoma de los Andes – UNIANDES matriculados en el periodo académico Abril – Septiembre 2023.

De los estudiantes encuestados, el 96% afirman tener un conocimiento altamente desarrollado en temas sobre algoritmos, especialmente el relacionar los algoritmos con las actividades diarias que cada uno realiza en su vida cotidiana. Específicamente en el desarrollo de algoritmos utilizando estructuras condicionales y repetitivas utilizando herramientas como PSEINT y SCRATH, los estudiantes encuentran un poco de dificultad al intentar trasladar un algoritmo narrado en lenguaje natural a un lenguaje un poco más con tecnicismo como lo es PSEINT, al trabajarlo en bloques con SCRATH. En un inicio les resultó complicado sobre todo conocer la herramienta y la ubicación de las respectivas opciones, el trabajo con las estructuras condicionales y repetitivas les resultó más sencillo utilizarlo. De los estudiantes encuestados, un 68% alcanzan un nivel de comprensión entre mediana y altamente desarrollado. Con el 32% restante se requiere tener un trabajo más personalizado, el cual alcanza un nivel de comprensión poco desarrollado.

Asimismo, los estudiantes manifiestan que comprender y dar una solución a un problema cotidiano no es muy difícil, pero al tratar de entender y dar solución a un problema que involucre algo de razonamiento matemático su nivel de comprensión del problema y posible explicación de la solución a un 52% le parece entre fácil y muy fácil, al 44% indica que es difícil comprender y dar solución a un problema y al 4% realmente es muy difícil para este grupo entender y sobre todo dar explicación de la solución al problema.

Discusión

Según el estudio realizado por [30-48-49], Scratch constituye una herramienta propicia para el desarrollo del pensamiento lógico y algorítmico para niños y estudiantes de Chile, y presenta un ambiente en el cual los estudiantes se motivan y participan en la propuesta de soluciones a las situaciones planteadas sin temor al error, posibilita el análisis de problemas y la propuesta, desarrollo y aplicación de soluciones lógicas y algorítmicas, las que se pueden probar y mejorar. Es decir, mediante pruebas de ensayo y error, los estudiantes pueden desarrollar y mejorar un pensamiento algorítmico. Cómo lo señala [30-50-51-52], mediante el uso de objetos, es posible el desarrollo de razonamiento algorítmico, y Scratch permite trabajar directamente con las propiedades y acciones de objetos. De esta forma, Scratch es una herramienta adecuada para la enseñanza de algoritmos y programación. Justamente, en la realización de este experimento, fue posible comprobar que las alternativas de respuesta a una pregunta lógica generan el razonamiento del estudiante sobre cuál es la opción correcta así como validar sus soluciones.

[31] En su investigación sobre el uso de Scratch en el aprendizaje de programación en Educación Superior menciona en sus conclusiones que la validez y el potencial que posee la herramienta Scratch no sea tan evidente en todas las dimensiones del pensamiento computacional, pero si en ciertos aspectos como mejorar la motivación, interés, creatividad e imaginación y facilitar el manejo de las sentencias al no concentrar la atención de los estudiantes en la sintaxis como en otros lenguajes de programación.

[32] En sus conclusiones indica que el inicio del proceso de desarrollo de la lógica de programación es necesario que los docentes utilicemos métodos que fomenten el desarrollo del razonamiento y no directamente a la práctica como lo evidencia los resultados de su investigación, para lo cual señala la necesidad de utilizar métodos como el aprendizaje basado en problemas y aprendizaje basado en investigación, en los cuales el estudiante aprende a inferir.

En base a la investigación realizada y en base a las experiencias expuestas por otros autores se puede indicar que a los estudiantes les hace falta desarrollar su razonamiento lógico matemático, que de este se desprende la capacidad de análisis para la resolución efectiva de problemas a través del uso y desarrollo de programas informáticos. El uso de Scratch en el desarrollo de la lógica de programación es de mucha ayuda, pero al carecer de razonamiento lógico por más sencillas y simples que sean las herramientas para este grupo de personas siempre será un grande desafío poderlas utilizar.

Conclusión

A partir de la implementación del método propuesto, se obtienen vectores de pesos de agregación para los criterios evaluativos que representó la base del proceso de análisis del impacto de Scratch. Se obtuvo como resultado del método la participación desinteresada de 9 expertos de los cuales 7 se utilizaron a partir de su coeficiente de competencia para la implementación del método propuesto que permitió la implementación del método propuesto.

Los resultados de la implementación del método propuesto sugieren que el uso de Scratch puede ser beneficioso para el desarrollo de la lógica de programación en estudiantes universitarios. Tanto para principiantes como para estudiantes con experiencia previa en programación, Scratch ha demostrado ser una herramienta efectiva para enseñar y desarrollar los conceptos fundamentales de programación. Además, su versatilidad y accesibilidad hacen que sea una opción viable para estudiantes de diferentes disciplinas. Sin embargo, es importante tener en cuenta que estos resultados se basan en estudios específicos y pueden variar dependiendo del contexto y las características de los estudiantes. Por lo tanto, es necesario realizar más investigaciones para obtener una visión más completa y precisa sobre el impacto del uso de Scratch en el desarrollo de la lógica de programación en estudiantes universitarios y especialmente en Carreras como Ingeniería de Software.

Referencias

- [1] M. Zapata-Ros, "Pensamiento computacional: Una nueva alfabetización digital," *Revista de Educación a Distancia (RED)*, no. 46, 2015.
- [2] E. C. Díaz, and G. L. Silvain, "El pensamiento computacional. Nuevos retos para la educación del siglo XXI," *Virtualidad, Educación y Ciencia*, vol. 11, no. 20, pp. 115-137, 2020.
- [3] L. X. F. Tello, J. F. L. Aguirre, J. C. P. Yuquilema, and J. L. L. Salazar, "Habilidades gerenciales para la revolución industrial 4.0 en el ámbito del capitalismo consciente," *Contribuciones a la Economía*, vol. 16, no. 3, pp. 5, 2018.
- [4] J. C. Casale, *Introducción a la programación: Aprenda a programar sin conocimientos previos*: RedUsers, 2016.
- [5] R. B. Capot, and R. M. Espinoza, "Desarrollo del pensamiento computacional con Scratch," *Nuevas Ideas en Informática Educativa*, vol. 11, pp. 616-620, 2015.
- [6] A. G. Rodríguez, "Enseñanza de la programación a través de Scratch para el desarrollo del pensamiento computacional en educación básica secundaria," *Revista Academia y Virtualidad*, vol. 15, no. 1, pp. 161-182, 2022.
- [7] A. S. Castillo, I. A. Berenguer, A. G. Sánchez, and Y. T. Fernández, "Lógica, algorítmica para la resolución de problemas de programación computacional: una propuesta didáctica," *Didasc@ lia: Didáctica y Educación*, vol. 4, no. 1, pp. 57-76, 2013.
- [8] S. A. Klotz, V. Ianas, and S. P. Elliott, "Cat-scratch disease," *American family physician*, vol. 83, no. 2, pp. 152-155, 2011.
- [9] S. H. d. M. Fernández. "Criterio de expertos. Su procesamiento a través del método Delphi," http://www.ub.edu/histodidactica/index.php?option=com_content&view=article&id=21:criterio-de-expertos-su-procesamiento-a-traves-del-metodo-delphi&catid=11.
- [10] Z.-S. Chen, K.-S. Chin, and K.-L. Tsui, "Constructing the geometric Bonferroni mean from the generalized Bonferroni mean with several extensions to linguistic 2-tuples for decision-making," *Applied Soft Computing*, vol. 78, pp. 595-613, 2019.
- [11] J. Giráldez - Cru, M. Chica, O. Córdón, and F. Herrera, "Modeling agent - based consumers decision - making with 2 - tuple fuzzy linguistic perceptions," *International Journal of Intelligent Systems*, vol. 35, no. 2, pp. 283-299, 2020.
- [12] S. Schmied, D. Großmann, S. G. Mathias, and S. Banerjee, "Vertical Integration via Dynamic Aggregation of Information in OPC UA." pp. 204-215.
- [13] P. T. Schultz, R. A. Sartini, and M. W. Mckee, "Aggregation and use of information relating to a users context for personalized advertisements," Google Patents, 2019.
- [14] N. Gospodinov, and E. Maasoumi, "Generalized Aggregation of Misspecified Models: With An Application to Asset Pricing," 2019.
- [15] L. B. Reyes, J. E. Suárez, and O. M. Cornelio, "Técnicas de Inteligencia artificial para el diagnóstico de pulsioximetría de apnea de sueño," *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 16, no. 4, pp. 1-10, 2023.
- [16] O. M. Cornelio, I. S. Ching, B. B. Fonseca, and P. M. P. Díaz, "Herramienta para la simulación de sistemas dinámicos integrado al sistema de laboratorios virtuales ya distancia."
- [17] V. V. Falcón, M. Y. L. Vázquez, and N. B. Hernández, "Desarrollo y validación de un cuestionario para evaluar el conocimiento en Metodología de la Investigación," *Revista Conrado*, vol. 19, no. S2, pp. 51-60., 2023.
- [18] M. Y. L. Vázquez, J. E. Ricardo, and V. Vega-Falcón, "La inteligencia artificial y su aplicación en la enseñanza del Derecho," *Estudios del Desarrollo Social: Cuba y América Latina*, vol. 10, pp. 368-380, 2022.
- [19] M. Y. L. Vázquez, J. E. Ricardo, and N. B. Hernández, "Investigación científica: perspectiva desde la neutrosofía y productividad," *Universidad y Sociedad*, vol. 14, no. S5, pp. 640-649., 2022.
- [20] J. L. G. González, and O. Mar Cornelio, "Propuesta de algoritmo de clasificación genética," *Revista Cubana de Ingeniería*, vol. 4, no. 2, pp. 37-42, 2013.
- [21] X. He, "Typhoon disaster assessment based on Dombi hesitant fuzzy information aggregation operators," *Natural Hazards*, vol. 90, no. 3, pp. 1153-1175, 2018.

- [22] O. Mar, I. Ching, and J. Gulín, "Competency assessment model for a virtual laboratory system at distance using fuzzy cognitive map," *Investigación Operacional*, vol. 38, no. 2, pp. 169-177, 2018.
- [23] P. Liu, H. Xu, and Y. Geng, "Normal wiggly hesitant fuzzy linguistic power Hamy mean aggregation operators and their application to multi-attribute decision-making," *Computers & Industrial Engineering*, vol. 140, pp. 106224, 2020.
- [24] R. R. Yager, and D. P. Filev, "Induced ordered weighted averaging operators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 2, pp. 141-150, 1999.
- [25] T. R. Sampson, C. Challis, N. Jain, A. Moiseyenko, M. S. Ladinsky, G. G. Shastri, T. Thron, B. D. Needham, I. Horvath, and J. W. Debelius, "A gut bacterial amyloid promotes α -synuclein aggregation and motor impairment in mice," *Elife*, vol. 9, pp. e53111, 2020.
- [26] O. M. Cornelio, and B. B. Fonseca, "Procedimiento multicriterio multiexperto para determinar el índice de control de una organización," *Scientia et Technica*, vol. 21, no. 3, pp. 234-238, 2016.
- [27] L. Jin, R. Mesiar, and R. Yager, "Ordered weighted averaging aggregation on convex poset," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 612-617, 2019.
- [28] X. Sha, Z. Xu, and C. Yin, "Elliptical distribution - based weight - determining method for ordered weighted averaging operators," *International Journal of Intelligent Systems*, vol. 34, no. 5, pp. 858-877, 2019.
- [29] H. Garg, N. Agarwal, and A. Tripathi, "Choquet integral-based information aggregation operators under the interval-valued intuitionistic fuzzy set and its applications to decision-making process," *International Journal for Uncertainty Quantification*, vol. 7, no. 3, 2017.
- [30] C. L. Vidal, C. Cabezas, J. H. Parra, and L. P. López, "Experiencias prácticas con el uso del lenguaje de programación Scratch para desarrollar el pensamiento algorítmico de estudiantes en Chile," *Formación universitaria*, vol. 8, no. 4, pp. 23-32, 2015.
- [31] H. O. Pérez Narváez, R. Roig-Vila, and L. Jaramillo-Naranjo, "Uso de SCRATCH en el aprendizaje de Programación en Educación Superior," 2020.
- [32] S. A. Machuca Vivar, C. R. Sampedro Guamán, D. P. Palma Rivera, and F. P. Cañizares Galarza, "Desarrollo de la lógica de programación en estudiantes de sistemas de Uniandes Santo Domingo," *Conrado*, vol. 17, no. 79, pp. 214-224, 2021.
- [33] Reyes, P. R. S. A., del Río, J. A. J., Sánchez, F. M., & Romero, A. V. "Hybrid and avant-garde methods for cost of capital evaluation". *Universidad y Sociedad*, vol 15 núm 4, pp 482-489, 2023. <https://rus.ucf.edu.cu/index.php/rus/article/view/4001/3919>
- [34] Márquez-Carriel, G., Márquez-Sánchez, F., & Vergara-Romero, A. "Relationship between the people's Republic of China and the Republic of Ecuador: a perspective from the dependency theory". *Universidad y Sociedad*, vol 15 núm 2, pp 49-62, 2023. <https://rus.ucf.edu.cu/index.php/rus/article/view/3605/3546>
- [35] Caveda, D. A., Sánchez, F. M., Ortega, R. A. S., & Chán, M. M. B. "El modelo pedagógico de la Universidad Tecnológica Ecotec: fundamentos epistemológicos, didácticos y metodológicos para su implementación". *Revista Científica ECOCIENCIA*, vol 2 núm 3, 2015. https://media.proquest.com/media/hms/ORIG/1/tUAsB?_s=dHk1ZlRf3SpGV%2B%2FY1rpgHfT9Br4%3D
- [36] Vergara-Romero, A., Morejón-Calixto, S., Márquez-Sánchez, F., & Medina-Burgos, J. "Economía del conocimiento desde la visión del territorio: Knowledge economy from the perspective of the territory". *Revista Científica ECOCIENCIA*, vol 9 núm 3, pp 37-62, 2022. <https://revistas.ecotec.edu.ec/index.php/ecociencia/article/view/680/430>
- [37] Carriel, G. M., Romero, A. V., Sánchez, F. M., & Molestina, G. A. "Bidependencia internacional en Ecuador: Estados Unidos y China: International bidependence in Ecuador: The United States and China". *REVISTA CIENTÍFICA ECOCIENCIA*, vol 9 núm 6, pp 1-19, 2022. <https://revistas.ecotec.edu.ec/index.php/ecociencia/article/view/747/457>
- [38] Sánchez, F. M. "EL SECTOR INFORMAL Y LA EVOLUCIÓN HISTÓRICA DE LA " PEQUEÑA EMPRESA" EN AMÉRICA LATINA, EN LA SEGUNDA MITAD DEL SIGLO XX". *Revista Científica Ecociencia*, vol 1 núm 1, pp 1-13, 2014.
- [39] Leonard, Y. G., Sánchez, F. M., del Río, J. A. J., & Romero, A. V. "Capítulo 4. Tourist destination management and cultural heritage: a perspective of the city of Havana". In *La gestión turística del patrimonio: una visión multidisciplinar* pp. 93, 2022. Thomson Reuters Aranzadi.
- [40] Ortega, R. S., del Río, J. A. J., Sánchez, F. M., & Romero, A. V. "Capítulo 18. Natural and cultural heritage un the turismo economy of the province of Guayas". In *La gestión turística del patrimonio: una visión multidisciplinar* pp. 421, 2022. Thomson Reuters Aranzadi.
- [41] Amores, E. R., Vega, L. L., Sánchez, F. M., & León, V. L. "Modelo econométrico de los gastos operativos de la banca en el Ecuador: Periodo 2012–2019: Econometric model of banking operating

- expenses in Ecuador: Period 2012-2019”. *REVISTA CIENTÍFICA ECOCIENCIA*, vol 8 núm 3, pp 80-99, 2021. <https://revistas.ecotec.edu.ec/index.php/ecociencia/article/view/514/342>
- [42] Pozo-Estupiñan, C., Sorhegui-Ortega, R., Márquez-Sánchez, F., & Vergara-Romero, A. “Pensamiento Económico: Sostenibilidad y Economía Agraria (Economic Thinking: Sustainability and Agricultural Economy)”. In Forthcoming, En IX Congreso Internacional “Tecnología, Universidad y Sociedad”. Samborondón, Ecuador, 2021.
- [43] del Valle Blanco, D., & von Feigenblatt, O. F. “Similitudes y complementariedades entre “La nueva agenda juventudes” y “La cumbre del futuro””. *Revista Internacional de Pedagogía e Innovación Educativa*, vol 4 núm 1, pp 289-296, 2024. <https://editic.net/ripie/index.php/ripie/article/view/163/138>
- [44] Aparicio-Gómez, O. Y., Ostos-Ortiz, O. L., & von Feigenblatt, O. F. “Competencia digital y desarrollo humano en la era de la Inteligencia Artificial”. *Hallazgos*, vol 20 núm 40, pp 217-235, 2023. <https://revistas.usantotomas.edu.co/index.php/hallazgos/article/view/9254/8156>
- [45] Von Feigenblatt, O. F., & Gómez, Ó. Y. A. (Eds.). “Transcending the eternal debate between traditional and progressive education: A constructive scholarly dialogue”. Ediciones Octaedro, 2023.
- [46] Ortiz, O. L. O., Aparicio-Gómez, O. Y., & von Feigenblatt, O. F. “Assessing a country’s scientific contribution towards sustainability from higher education: a methodology for measuring progress towards the Sustainable Development Goals (SDG)”. *Revista Interamericana de Investigación Educación y Pedagogía RIIEP*, vol 16 núm 2, pp 343-361, 2023. <https://revistas.usantotomas.edu.co/index.php/riiep/article/view/8848/8052>
- [47] von Feigenblatt, O. F., & Ricardo, J. E. “The challenge of sustainability in developing countries: the case of Thailand”. *Universidad y Sociedad*, vol 15 núm 4, pp 394-402, 2023. <https://rus.ucf.edu.cu/index.php/rus/article/view/3992/3910>
- [48] von Feigenblatt, O. F. “Traditional Education in the Public Sphere: A Contested Terrain”. *Revista Internacional de Filosofía Teórica y Práctica*, vol 3 núm 2, pp 87-106, 2023. <https://www.editic.net/riftp/index.php/riftp/article/view/81/68>
- [49] von Feigenblatt Rojas, O. F. “Introducing the emerging field of Academic Diplomacy”. *Universidad y Sociedad*, vol 15 núm 2, pp 316-325, 2023. <https://rus.ucf.edu.cu/index.php/rus/article/view/3632/3573>
- [50] Pérez, G. R., Marqués, L. L., Poleo, A. J., Rivera, A., & von Feigenblatt, O. F. “El liderazgo educativo en los programas de educación especial: Una revisión de la literatura”. In *Anales de la Real Academia de Doctores* vol 8, núm 4, pp 785-801, 2023.
- [51] Negro, A. R., Marqués, L. L., Poleo, A. J., & von Feigenblatt, O. F. “La responsabilidad social corporativa, Una revisión histórica alineada a dos teorías que colaboran con el concepto”. In *Anales de la Real Academia de Doctores* vol 8, núm 4, pp 769-783, 2023
- [52] Solís, N. F., Marqués, L. L., Poleo, A. J., & von Feigenblatt, O. F. “Uso de factores resilientes en la gestión empresarial durante tiempos de crisis por Pandemia Covid-19”. In *Anales de la Real Academia de Doctores* vol. 8, núm. 3, pp 505-511, 2023
- [53] Verdezoto, M. I. M., Álvarez, D. R., & Falcón, V. V. “Neutrosophic Evaluation of Legal Strategies for Decision-making in a Digital Context”. *Neutrosophic Sets and Systems*, vol 62 num 1, pp 20, 2023. https://digitalrepository.unm.edu/cgi/viewcontent.cgi?article=2536&context=nss_journal

Recibido: noviembre 17, 2023. **Aceptado:** diciembre 05, 2023