



DESARROLLO DEL PENSAMIENTO LÓGICO MATEMÁTICO A TRAVÉS DE LA PROGRAMACIÓN EN PYTHON

Bryan Alejandro Collazos Estrada
Daniela Correa Santiago

Universidad del Valle
Facultad de Educación y Pedagogía
Licenciatura en Matemáticas
2024



DESARROLLO DEL PENSAMIENTO LÓGICO MATEMÁTICO A TRAVÉS DE LA
PROGRAMACIÓN EN PYTHON

Bryan Alejandro Collazos Estrada

Daniela Correa Santiago

Yonathan Bonelo Ayala

Director Trabajo de Grado

Universidad del Valle

Facultad de Educación y Pedagogía

Licenciatura en Matemáticas

2024

Tabla de Contenido

	Pág.
Resumen	7
Introducción	9
1. Definición del problema	11
1.1 Objetivo general	17
1.1.1 Objetivos específicos	17
1.1.2 Preguntas auxiliares	17
1.2 Revisión de la literatura	18
2. Marco teórico	26
2.1 Marco teórico orientador: el enfoque fenomenológico	27
2.2 Marco teórico para la acción: la Teoría de Mediación Semiótica (TMS)	28
2.3 Marco de dominio específico: sobre el pensamiento lógico matemático, el pensamiento computacional y la programación en Python	31
2.3.1 Pensamiento lógico matemático	32
2.3.2 Pensamiento computacional	34
2.3.3 Programación computacional	36
3. Marco metodológico	40
3.1 Sobre la estructura metodológica	40
3.2 Naturaleza de la investigación	41
3.2.1 Enfoque: fenomenológico	43
3.2.2 Aproximación: interpretativa	43
3.2.3 Recursos (información y análisis de datos)	44
3.2.4 Estrategia cualitativa: estudio de caso	45
3.3 Recursos: captura información	46
3.3.1 Escenario ¿Dónde?	46
3.3.2 Informantes ¿A quién?	46
3.3.3 Eventos ¿Cuándo?	47
3.3.4 Procesos ¿haciendo qué?	48
3.3.5 Recursos: Actividades	50

3.3.6 Sobre la estructura del análisis	51
4. Análisis de datos	53
4.1 Sobre el análisis	53
4.2 Encuesta previa a las actividades	53
4.3 Análisis actividad 0	60
4.3.1 Descripción de la actividad	60
4.3.2 Desarrollo de la actividad	61
4.3.3 Objetivo	62
4.3.4 Análisis local A0	62
4.4 Análisis actividad 1	66
4.4.1 Descripción de la actividad	66
4.4.2 Objetivo	68
4.4.3 Análisis local A1	68
4.5 Análisis actividad 2	73
4.5.1 Descripción de la actividad	73
4.5.2 Objetivo	76
4.5.3 Análisis local A2	76
4.6 Encuesta final	82
4.7 Análisis retrospectivo	90
4.7.1 Descripciones sobre los análisis realizados	90
4.7.2 El rol de la TMS y el vínculo entre el pensamiento matemático y computacional	92
4.7.3 La resolución de problemas y el pensamiento lógico matemático a través de Python	93
5. Conclusiones y consideraciones finales	95
5.1 Objetivo específico 1	95
5.2 Objetivo específico 2	96
5.3 Objetivo específico 3	98
5.4 Consideraciones finales	100
5.4.1 A la investigación	100
5.4.2 A las prácticas pedagógicas	101
Referencias	102
Anexos	109

Índice de tablas

<i>Tabla 1.</i> Habilidades del pensamiento matemático y elementos computacionales que las desarrollan	15
<i>Tabla 2.</i> Ficha panorámica metodológica	40
<i>Tabla 3.</i> Cronograma de actividades de la investigación	47
<i>Tabla 4.</i> Respuestas de la pregunta 1 de la encuesta introductoria	54
<i>Tabla 5.</i> Respuestas de la pregunta 2 de la encuesta introductoria	56
<i>Tabla 6.</i> Respuestas de la pregunta 3 de la encuesta introductoria	57
<i>Tabla 7.</i> Respuestas de la pregunta 4 de la encuesta introductoria	58
<i>Tabla 8.</i> Respuestas de la pregunta 5 de la encuesta introductoria	59
<i>Tabla 9.</i> Respuestas de la pregunta 1 de la encuesta final	82
<i>Tabla 10.</i> Respuestas de la pregunta 2 de la encuesta final	84
<i>Tabla 11.</i> Respuestas de la pregunta 3 de la encuesta final	85
<i>Tabla 12.</i> Respuestas de la pregunta 4 de la encuesta final	86
<i>Tabla 13.</i> Respuestas de la pregunta 5 de la encuesta final	88

Índice de figuras

<i>Figura 1.</i> Proceso de la investigación de desarrollo	44
<i>Figura 2.</i> Plan de ejecución de la estrategia “estudio de caso”	47
<i>Figura 3.</i> Trabajo realizado en A0 por E1, E3, E4	64
<i>Figura 4.</i> Trabajo realizado en A0 por E1, E4	66
<i>Figura 5.</i> A1 - Sección 1	67
<i>Figura 6.</i> A1 - Sección 2	68
<i>Figura 7.</i> A1 - Sección 3	68
<i>Figura 8.</i> Respuesta E4. A1 - Sección 3. Pregunta 1	70
<i>Figura 9.</i> Respuesta E2. A1 - Sección 3. Pregunta 1	71
<i>Figura 10.</i> Respuesta E4. A1 - Sección 3. Pregunta 2	72
<i>Figura 11.</i> Respuesta E1. A1 - Sección 3. Pregunta 2	73
<i>Figura 12.</i> A2 - Sección 1	75
<i>Figura 13.</i> A2 - Sección 2	76
<i>Figura 14.</i> A2 - Sección 3	76
<i>Figura 15.</i> Respuesta E4. A2 - Sección 2. Pregunta 2	78
<i>Figura 16.</i> Respuesta E1. A2 - Sección 2. Pregunta 2	79
<i>Figura 17.</i> Respuesta E1. A2 - Sección 3. Pregunta 1	81
<i>Figura 18.</i> Respuesta E6. A2 - Sección 3. Pregunta 1	82

Resumen

Este trabajo se centra en investigar cómo la programación en Python puede fortalecer el desarrollo del pensamiento lógico-matemático en estudiantes de secundaria. Mediante un enfoque cualitativo fenomenológico, se implementaron tres actividades en un estudio de caso con siete estudiantes de noveno a undécimo grado, utilizando Google Colaboratory. El análisis de las actividades permitió observar que la integración del pensamiento computacional a través de la programación con el lenguaje Python facilitó la resolución de problemas, la abstracción y la estructuración de procesos lógicos, lo que contribuye al aprendizaje significativo de las matemáticas. El estudio concluye que el uso de la programación no solo potencia habilidades matemáticas, sino que también promueve una mayor autonomía y creatividad en los estudiantes.

Palabras clave: pensamiento lógico-matemático, programación en Python, educación matemática, pensamiento computacional, resolución de problemas, estudio de caso

Introducción

El creciente interés de la educación por la implementación de tecnología computacional en las aulas da paso a diversas y emergentes oportunidades en el campo de la Educación Matemática, permitiendo así el desarrollo de metodologías y recursos novedosos en los procesos de enseñanza y aprendizaje de las matemáticas escolares. En este sentido, surge la necesidad de presentar una investigación en torno a la forma en que la programación computacional, específicamente el lenguaje Python, puede constituir un recurso pedagógico para fortalecer el aprendizaje de las matemáticas y el desarrollo del pensamiento lógico matemático en estudiantes de la educación secundaria en Colombia, específicamente de grados noveno, décimo y undécimo.

Este trabajo de investigación está estructurado en cinco capítulos. El primero presenta la definición del problema de investigación, estableciendo el objetivo general: caracterizar el rol de la programación computacional en la enseñanza de las matemáticas y en el desarrollo de habilidades propias al pensamiento matemático del estudiante. Además de esto, el primer capítulo también da cuenta de la revisión de la literatura realizada para la investigación, así como el establecimiento de categorías que fueron necesarias para la escogencia de los documentos referenciales.

El segundo capítulo expone el marco teórico, en donde se abordan conceptos clave como pensamiento lógico-matemático, pensamiento computacional y la programación como elemento integrador de los dos referentes conceptuales anteriores.

En el tercer capítulo se presenta la metodología a seguir en la investigación a través de una ficha panorámica, en la que se detalla la naturaleza cualitativa e interpretativa, que bajo el enfoque fenomenológico y la estrategia de estudio de caso permitirá hacer un análisis de los resultados con el fin de caracterizar y presentar el impacto de la programación en las aulas de matemáticas.

En el cuarto capítulo se recogen los análisis realizados a las tres actividades implementadas en el aula de clase, donde se describen las actividades, los objetivos y la relación con el marco teórico propuesto, además de un análisis retrospectivo donde se vinculan elementos teóricos con el panorama general de la investigación.

En el quinto y último capítulo se exponen las conclusiones encontradas en torno a los objetivos planteados, y cómo estos se relacionan con los análisis para caracterizar algunos de los hallazgos sobre el uso de la programación en el aula de matemáticas, de otra parte, se comparten algunas reflexiones con propósitos pedagógicos e investigativos para futuros estudios o docentes en práctica.

El propósito de este trabajo consiste en proporcionar una caracterización de la programación en Python y su rol en el desarrollo del pensamiento lógico-matemático en el ámbito de la enseñanza y el aprendizaje de las matemáticas escolares. Los hallazgos obtenidos tendrán como propósito ofrecer orientación en las consideraciones relacionadas con el análisis propuesto y los objetivos, así como en propuestas pedagógicas en este ámbito.

1. Definición del problema

En Colombia se han llevado a cabo diversas reformas y cambios curriculares en los enfoques de la educación y la enseñanza de las matemáticas a lo largo de los años (Vasco, 2018). A pesar de que en los últimos años ha aumentado la inclusión de dispositivos tecnológicos en las instituciones educativas colombianas, aún son escasas las experiencias pedagógicas que incorporan el uso efectivo de estas herramientas para potenciar el aprendizaje de las matemáticas. El aprovechamiento real de las TIC en las aulas de clase matemáticas del país es bastante restringido y su implementación para transformar los procesos de enseñanza-aprendizaje es un área muy poco explorada dentro del campo educativo nacional. Esto es una oportunidad para vislumbrar los alcances de estas herramientas, así como su interactividad a favor del desarrollo de destrezas lógico-matemáticas significativas de los estudiantes (Padilla, 2022).

De ahí que, autores como Reich (1991) aseguren que las reformas educativas deberían preparar a muchas más personas con el fin de promover una educación autónoma y constante, con apoyo en el aprendizaje colaborativo¹ de tal forma que sus habilidades en conjunto sean más significativas que sus aportes individuales.

En el contexto local, los referentes curriculares de matemáticas en Educación Básica y Media en Colombia se encuentran desactualizados con relación a los intereses de la sociedad moderna, pues en su concepción original del siglo XX su principal enfoque era moldear al futuro empleado, quien solía permanecer en el mismo trabajo durante toda su

¹ Enfoque educativo que enfatiza la importancia de que los estudiantes trabajen en conjunto para resolver problemas, completar tareas y aprender los unos de los otros.

vida y no requería un amplio conjunto de habilidades o creatividad. Tal como lo afirman Salazar & Fernández (2017), las necesidades presentes en el contexto educativo actual difieren de las del siglo anterior, y es por esto por lo que los currículos deberían de ajustarse con base en objetivos que solventen las necesidades del contexto educativo en el que nos encontramos.

Investigaciones realizadas por los autores Cristancho (2019) y Mariotti (2002) han permitido entrever la importancia de promover nuevas metodologías educativas en el aula de matemáticas que potencien el desarrollo del pensamiento matemático de forma transversal a otras disciplinas. En este sentido, ha surgido la necesidad de hablar sobre la importancia de la integración y uso de las tecnologías de la información y comunicación (TIC) en los procesos de enseñanza y aprendizaje en las aulas de matemáticas, logrando así un avance en la educación de acuerdo con las necesidades del mundo moderno.

Esto implica la revisión de los planes de estudio actuales, metodologías educativas, enfoques pedagógicos y la exploración del papel que tiene la Educación Matemática con la sociedad moderna a medida que se incorporan nuevas tecnologías de la información y comunicación en la medida en que la inclusión de recursos digitales es cada vez más relevante en el ámbito educativo. Pico et al. (2017) afirman que la educación matemática se beneficia enormemente de la tecnología, puesto que, ayuda en la formación de estudiantes con un profundo entendimiento, con habilidades analíticas, estudiantes creativos, y con la capacidad para pensar crítica y activamente, así mismo, capaces de validar y abordar desafíos cotidianos con eficacia.

Sin embargo, actualmente existen docentes que oponen resistencia hacia el uso de la tecnología en las aulas de clase, esto se debe tanto a factores internos del docente, así como factores externos socioculturales y de infraestructura de la institución. Pongamos por caso la falta de preparación en la formación de profesores en el uso de las TIC, fomentando miedo a lo desconocido y oposición al adaptar nuevas metodologías y oportunidades pedagógicas en el aula. Así mismo, la falta de interés presentada por los docentes por el uso de las tecnologías es un factor perjudicial para la Educación Matemática en Colombia dado que, al no existir un interés por el uso de estas herramientas lo más probable es que no se dé una implementación adecuada de las mismas, tal como lo afirman Tapia et al. (2017), es habitual observar que en las aulas de clase de matemáticas los docentes a menudo emplean las Tecnologías de la Información y la Comunicación (TIC) exclusiva y únicamente como una fuente de información y una herramienta de comunicación. De esta manera, dejando completamente de lado todas las oportunidades, potencialidades y virtudes que ofrecen las TIC en la enseñanza y aprendizaje de las matemáticas.

Así pues, nace un profundo interés por considerar herramientas tecnológicas que permitan cumplir con dichas exigencias, así como, es crucial que se dé un buen uso de estas en las aulas de matemática, y es aquí donde aparece la programación computacional y se establece como un factor para tener en cuenta en el desarrollo del currículo matemático moderno. Pues, como destaca Papert (1980), la programación no solamente es relevante en el ámbito tecnológico, sino que a su vez desarrolla habilidades importantes como la resolución de problemas, el pensamiento lógico y la creatividad, que son valiosas

en una amplia variedad de campos. La falta de inclusión de la programación computacional en el currículo de matemáticas también puede limitar la capacidad de los estudiantes para enfrentar los desafíos del mundo digital actual.

Considerando lo mencionado previamente, se observa que la programación propone diversos beneficios en el ámbito de la Educación Matemática, los cuales son de particular interés para nuestra investigación. Es posible decir que la programación es capaz de establecerse como un recurso a considerar en el fomento del pensamiento lógico matemático en los estudiantes al permitir la creación de algoritmos y códigos, facilitando el desarrollo de la secuencialidad y la lógica mientras los estudiantes los manipulan. Además, su eficacia en la resolución de problemas es notable, ya que los alumnos, con el apoyo de la programación, aprenden a descomponer situaciones complejas, fortaleciendo así sus habilidades y destrezas en la resolución de problemas.

Asimismo, durante el proceso de depuración del código, los estudiantes pueden mejorar su análisis lógico, al mismo tiempo que la creatividad en el momento de ingeniar soluciones aporta un componente innovador en el pensamiento lógico-matemático. Con lo cual podemos observar que, la programación parece promover el desarrollo de habilidades esenciales en los estudiantes, en un contexto práctico y aplicado en la educación.

No obstante, no es únicamente respecto al mundo digital actual donde la programación se establece como un referente óptimo para solucionar problemas, sino que, muchas de las competencias matemáticas actuales se relacionan directamente con habilidades propias de la programación, tal es el caso de la generalización, la resolución de problemas, formulación de algoritmos y la modelización. Por ende, es oportuno tomar en

consideración a la programación como una forma de desarrollar el pensamiento matemático de los estudiantes.

En relación con esto, Acevedo (2018) muestra algunos conceptos y habilidades matemáticas esenciales del pensamiento matemático que pueden ser vistos y desarrollados con la ayuda de herramientas o elementos computacionales, fomentando el desarrollo de habilidades del pensamiento computacional y relacionándolas con el pensamiento matemático del estudiante, como lo vemos en la siguiente tabla:

Tabla 1. *Habilidades del pensamiento matemático y elementos computacionales que las desarrollan*

Habilidades del pensamiento matemático implementados	Elementos computacionales que facilitan el desarrollo de esta habilidad
Formular problemas de la vida cotidiana de manera simbólica para aplicarlos en computadoras y otras herramientas computacionales que permitan solucionarlos	Las representaciones abstractas y la organización de procesos mediante algoritmos permiten plantear problemas, llevar a cabo las tareas solicitadas y obtener soluciones. En computación, esto se refleja en el uso de pseudocódigos y diagramas de flujo, los cuales facilitan la comprensión de cómo el computador recibe, procesa y transmite la información.
Organizar datos de manera lógica para analizarlos	El empleo de variables y condicionales es crucial. Al comprender y utilizar el manejo de variables, podemos clasificar y depurar información de manera ordenada y eficiente. Del mismo modo, el manejo de condicionales nos ayuda a entender la relevancia de los datos en un momento dado, permitiendo ejecutar diferentes operaciones según las variaciones en estos datos.
Representar datos mediante abstracciones, como modelos y simulaciones	El empleo de pseudocódigos y diagramas de flujo no solo facilita que los alumnos comprendan y desarrollen representaciones abstractas de la realidad, sino que también establece una base inicial para simular situaciones dentro de un programa de computadora. Esto se logra mediante la implementación de pruebas que permiten verificar el funcionamiento correcto de un algoritmo específico en diferentes escenarios.
Desarrollo del pensamiento algorítmico y el razonamiento numérico	El manejo de ciclos repetitivos y solicitud de datos por teclado permite al alumno comprender y asimilar que algunos procesos que deben realizarse varias veces pueden ser definidos e implementados como ciclos repetitivos. Cuando se solicitan datos por teclado

	dentro de un ciclo repetitivo, no solo se comprende la ejecución de un proceso varias veces, sino que también se observa que sin importar las entradas o salidas de un proceso este puede llegar a ser generalizado y ejecutado por lo que el razonamiento numérico, ayuda en gran medida al desarrollo del pensamiento algorítmico.
Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva	El empleo de estructuras condicionales como "If" y "Else" permite al estudiante observar y ajustar el comportamiento de un programa de computadora en función de los parámetros requeridos. Esto facilita la identificación de recursos y secuencias necesarios para la solución de un problema específico, lo que conduce a la optimización de recursos al utilizar solo los necesarios.
Generalizar y transferir el proceso de solución de problemas a otros de la misma índole	Al replicar ejemplos en ejercicios similares, pero con diferentes niveles de dificultad, el estudiante consolida su comprensión de un tema y comprende cómo puede generalizar y adaptar una solución a distintos problemas del mismo tipo.

Nota. Tomado y adaptado de: Orjuela & Rodríguez (2022, p. 16 - 17).

Siendo la educación el motor del desarrollo de un país, el aprendizaje de la Matemáticas es uno de los pilares más importantes, ya que el pensamiento lógico, el pensamiento crítico, la argumentación fundamentada y la resolución de problemas son sustanciales para alcanzar los objetivos de la educación básica y media que Colombia propone (Farfan, 2014).

En este orden de ideas, en el presente trabajo de investigación se explorará el papel de la programación como una opción para el fortalecimiento de la Educación Matemática en Colombia, haciendo énfasis en el desarrollo del pensamiento lógico matemático. Así mismo, serán evidenciadas algunas ventajas que provee el uso de la programación en la educación secundaria y como esta puede fomentar la autonomía, la comunicación y la colaboración entre los estudiantes. A su vez, facilitar la comprensión de objetos

disciplinares y además permitiéndoles alcanzar un nivel de aprendizaje verdaderamente significativo.

Con este propósito, se propone la siguiente pregunta problema que pretende guiar la investigación:

¿De qué manera la integración del pensamiento computacional a través del uso de la programación en Python actúa como un recurso pedagógico en el desarrollo del pensamiento lógico matemático?

1.1 Objetivo general

Caracterizar el rol de la programación computacional como recurso pedagógico en el desarrollo del pensamiento lógico matemático en el aprendizaje de las matemáticas en la Educación Secundaria en Colombia.

1.1.1 *Objetivos específicos*

- Identificar las competencias del pensamiento lógico matemático que se desarrollan a través de la programación computacional.
- Analizar las percepciones de los estudiantes sobre el aprendizaje de la programación dentro del currículo de matemáticas en la Educación Secundaria en Colombia.
- Caracterizar el vínculo entre el pensamiento computacional y el pensamiento matemático y su relación con la Educación Matemática.

1.1.2 Preguntas auxiliares

- ¿Cuáles son los beneficios específicos de la integración del pensamiento computacional en la enseñanza de las matemáticas?
- ¿Cuáles son los obstáculos comunes que los estudiantes enfrentan al aprender matemáticas a través de la programación y cómo pueden abordarse?

1.2 Revisión de la literatura

Con el propósito de recopilar la documentación necesaria para la presente investigación se han seleccionado documentos de diversas bases de datos académicas, tales como: Google Scholar, Journal of Information Technology Education: Research, Math Education Programming, ResearchGate, SciELO, Scopus, Springer, entre otras. Esto pues, con el fin de vincular documentación actualizada y lo suficientemente rigurosa para justificar los intereses presentes en la investigación. La documentación buscada se realizó tanto en inglés como en español.

En primera instancia, la selección de los documentos se centró en criterios de selección relacionados a elementos característicos de los referentes, en este caso, la primera condición de selección consideraba documentos cuyo título o resumen tuvieran relación con la Educación Matemática y el uso de Python en la misma; como segunda condición, se revisaron las fechas de los documentos con el fin de que fuesen no mayores a 15 años de antigüedad, cabe aclarar que, este segundo filtro fue exclusivo de los documentos referentes al papel de Python dentro del aula de matemáticas, y no considera a

los documentos propios a la pedagogía y didáctica, pues estos se consideraron más adelante.

En la medida en que la búsqueda de documentos avanzaba, prontamente empezó a presentar dificultades y limitaciones con los resultados, puesto que, la escasez de artículos y trabajos de grado que hablaran con la especificidad requerida para la investigación fue inminente. Dándonos así una idea de que las investigaciones que integran Python y su uso en la Educación Matemática parecen ser escasas. Es por esto que, surgió la necesidad de realizar una investigación más exhaustiva en donde se incluyeran documentos, artículos y tesis con relación a un concepto más amplio, pero sin dejar de ser inherente al tema de interés: la programación y el pensamiento computacional.

Así pues, la búsqueda de documentos abordó un camino más amplio y vasto donde fue posible encontrar diversas fuentes de información lo suficientemente rigurosas para la investigación. Se consideraron artículos principalmente centrados en la educación media en diversos países, donde la tecnología y el uso de la programación en las aulas de clase de matemáticas fueron los protagonistas. Con base en estos nuevos documentos obtenidos se centró la atención en clasificarlos en grupos o categorías simples, dicha clasificación no siguió ninguna referencia teórica, sino que, se consideró la coherencia entre los documentos y su relación con contenidos generales, de esta forma, fueron clasificados en dos categorías: (1) Pensamiento matemático/computacional y (2) Programación computacional y Python en Educación Matemática.

Categoría 1: sobre la relación entre el pensamiento matemático y el computacional

La categoría “Pensamiento matemático y computacional” resulta de gran importancia para nuestra investigación, dado que aborda conceptos clave sobre cómo se puede promover el desarrollo de habilidades de pensamiento matemático en los estudiantes a través del uso de la programación y el uso de herramientas computacionales en la Educación Matemática. Así mismo, esta categoría nos permite obtener elementos teóricos y prácticos para diseñar actividades que fomenten en los estudiantes capacidades como la resolución de problemas, el pensamiento crítico y el trabajo colaborativo. Algunos documentos referentes en esta categoría permiten ver la importancia de metodologías STEM en las aulas de matemáticas, centrando su atención en la interacción y el desarrollo de las competencias matemáticas y computacionales diseñadas de formas específicas (Cabra & Ramírez, 2022).

Por otro lado, al analizar los documentos pertenecientes a esta categoría podemos identificar los retos y dificultades que conlleva su implementación en las aulas de matemática, tales como, la necesidad de realizar una capacitación docente en el uso de herramientas computacionales, la dificultad de integrar la programación computacional en los currículos establecidos o la escasez de recursos tecnológicos en las instituciones educativas. Es importante resaltar que, la resolución de problemas aparece como una habilidad importante y central en los trabajos pertenecientes a esta categoría, pues conecta de buena manera ambos pensamientos en la medida en que provee de beneficios a los procesos de enseñanza y aprendizaje de las matemáticas escolares, por lo que es una idea para considerar (Costa & Rodríguez, 2017).

Tener en cuenta estas experiencias será de suma importancia para nosotros, ya que nos permitirá diseñar una propuesta sólida y realista para incorporar Python en la enseñanza de las matemáticas, previniendo posibles obstáculos y contemplando soluciones efectivas. En síntesis, esta categoría aporta fundamentos sólidos teóricos y prácticos de suma importancia para esta línea de investigación.

Categoría 2: sobre la influencia del uso de la programación en Educación Matemática

La categoría "Programación computacional y Python en Educación Matemática" es fundamental para nuestra investigación, pues representa la convergencia de dos áreas esenciales: la programación computacional y sus aplicaciones en las clases de matemáticas escolares. Este enfoque permite no sólo explorar el uso de la programación como recurso pedagógico para beneficiar el aprendizaje de las matemáticas, sino también comprender la relación entre las habilidades computacionales y matemáticas que pueden desarrollar los estudiantes, claves en la sociedad actual. Al incluir esta categoría, adoptamos una perspectiva multidisciplinaria (Pérez & Rosa, 2017) con potencial para transformar la enseñanza y el aprendizaje de las matemáticas, promoviendo simultáneamente la alfabetización digital y el pensamiento crítico en los estudiantes.

Esta categoría es especialmente relevante para nuestra investigación puesto que reúne documentos directamente relacionados con la integración de lenguajes de programación como Python en la Educación Matemática. Aunque inicialmente fue difícil hallar suficientes fuentes específicas sobre el uso de Python en las aulas de matemáticas, al ampliar la búsqueda hacia una visión más general de la programación computacional en

las matemáticas escolares logramos obtener una base documental sólida. Pues los artículos y trabajos de esta categoría aportan fundamentos, experiencias y resultados existentes sobre distintos lenguajes y plataformas de programación aplicados a la enseñanza de las matemáticas.

El uso de la programación computacional en Educación Matemática provee beneficios en los procesos de enseñanza y aprendizaje, pues permite resignificar metodologías tradicionales a través de la introducción de situaciones y recursos innovadores para los estudiantes. En este sentido, la interactividad y el compromiso de los estudiantes aumenta, al mismo tiempo que desarrolla el pensamiento computacional y les permite comprender conceptos matemáticos desde aproximaciones interesantes y novedosas (Kilhamn et al., 2021).

En particular, el lenguaje de programación a considerar para la presente investigación es Python, el cual es un lenguaje de alto nivel, interpretado y orientado a objetos, con una sintaxis dinámica que enfatiza la legibilidad del código, Python es cada vez más popular en el ámbito educativo debido a su simplicidad y aplicabilidad en diversas áreas, como la inteligencia artificial y el análisis de datos, convirtiéndolo en un lenguaje de programación muy versátil y cada vez más empleado en diversos campos del mundo digital. La creación de algoritmos se puede dar a través de diferentes compiladores, para nuestro interés, emplearemos Google Colab puesto que es un software en la nube que no requiere descargas sino únicamente acceder a través del conjunto de aplicaciones de Google, en este, los estudiantes podrán crear “cuadernos” donde se combina código fuente

junto con texto, con el fin de generar secuencias que den solución a problemas propuestos.

Python, además de tener bibliotecas matemáticas incorporadas que proporcionan funciones matemáticas para realizar cálculos complejos, con la adecuada implementación a las aulas podría promover la comprensión de las matemáticas escolares de una forma más versátil que las metodologías tradicionales, a través el desarrollo de algoritmos y potenciamiento del razonamiento lógico de los estudiantes. Por lo que su adaptación en las aulas de clase, junto con la correcta mediación del docente podría beneficiar los procesos de enseñanza y aprendizaje basados en el entendimiento de conceptos abstractos de forma dinámica a través de la visualización e interacción que permite la programación (Kado, 2022).

En síntesis, este enfoque enriquece nuestra investigación al ofrecer una perspectiva interdisciplinaria que contribuye al progreso de ambas disciplinas y al fortalecimiento de habilidades cognitivas y tecnológicas en los estudiantes. Su estudio revela nuevos enfoques y estrategias para la enseñanza de las matemáticas, fomentando la resolución de problemas, la creatividad y la aplicabilidad de conceptos matemáticos en situaciones reales (Kaufmann & Stenseth, 2020). Además, permite explorar la adaptación de distintos enfoques de la programación a diversos contextos educativos.

En este orden de ideas, la clasificación de documentos permitió obtener información relevante de forma óptima, mediante revisiones constantes de los mismos y el desarrollo de instrumentos que permitieran comparar múltiples referentes y establecer relaciones comunes entre ellos. Un instrumento empleado fue la tabla de comparaciones

por objetivos, la cual se basó en 6 trabajos de grado/maestría sobre el uso de la programación en el aula de matemáticas, cuyos títulos son:

- Resolución de problemas a través de programación en Python centrado en la Teoría de Situaciones Didácticas dirigido a estudiantes de grado noveno
- Desarrollo del pensamiento variacional y computacional a través del lenguaje Python en Educación Básica Secundaria en Colombia
- Propuesta didáctica para la enseñanza del concepto de polígono mediante el módulo tortuga de Python
- TIC en la escuela secundaria: Una propuesta de enseñanza conjunta de programación y matemática
- El lenguaje de programación como estrategia didáctica para desarrollar competencias lógico matemáticas en los estudiantes de grado octavo de la institución educativa Jorge Eliecer Gaitán Peñol - Nariño
- Influencia de la experiencia de aula “Lógica matemática y programación en SCRATCH: un acercamiento al aprendizaje basado en el juego”, en la motivación para el aprendizaje de las matemáticas, de los estudiantes de educación media de los colegios adventistas de Colombia.

Nos enfocamos en trabajos realizados en Colombia y/o Latinoamérica. La estructura de la tabla permitió explicitar los objetivos de los trabajos, sus preguntas de investigación y algunos de los resultados obtenidos. Esto con el objetivo de identificar ideas comunes y dar un horizonte más claro y conciso a nuestra investigación.

Posterior a dicha tarea se procedió a hacer una revisión de los diferentes trabajos de grado/maestría (tanto los que fueron empleados en la tabla como otros cuantos más) de forma más exhaustiva, cabe aclarar que, estos trabajos hacían énfasis en la integración de la programación en el aula de matemáticas con la finalidad de estudiar el rol de la programación computacional y su impacto. En la medida en que la revisión avanzaba, advertíamos la escasez de fundamentación en los diseños de aula que proponían los trabajos, mientras que muchos ofrecían importantes consideraciones didácticas y pedagógicas. En este sentido, un punto en común, fue la ausencia de diseños propios a la programación en el aula de matemáticas o el poco énfasis en las actividades planteadas en torno a la programación, por lo que es menester de nuestra investigación considerar fundamentación tanto teórica como práctica que permitan dar un papel protagónico a la programación dentro de la Educación Matemática.

2. Marco teórico

El marco teórico permitirá contextualizar y fundamentar el estudio dentro de las teorías y conceptos relevantes para el tema de investigación. Servirá para definir las variables clave, establecer relaciones entre conceptos y proporcionar una base conceptual sólida para interpretar los resultados obtenidos. Además, permitirá vincular el estudio con investigaciones previas, identificando coincidencias, contradicciones e incluso posibles vacíos de conocimiento. En el contexto del enfoque de investigación de diseño (Bakker & Van Eerde, 2015; DiSessa & Cobb, 2004), proponen tres enfoques de referencia, los cuales serán abordados e implementados en esta investigación. Estos enfoques comprenden un marco teórico orientador, un marco teórico para la acción y teorías específicas de aprendizaje en el ámbito en cuestión.

Según Bakker y Van Eerde (2015), los marcos teóricos orientadores brindan una visión general sobre aspectos del aprendizaje, la enseñanza y el diseño instruccional, aunque se encuentran limitados con relación a cómo implementar entornos de aprendizaje efectivos, ante esta situación, los marcos de acción surgen como una herramienta útil al enfocarse en guiar la creación de ambientes de aprendizaje y definir acciones concretas para la enseñanza (DiSessa y Cobb, 2004). Complementariamente, las teorías de aprendizaje específicas del dominio aportan una perspectiva enfocada y detallada. Así, esta investigación adopta una triple perspectiva teórica: un marco orientador, un marco para la acción y teorías de aprendizaje del campo en cuestión, los cuales se presentarán en los siguientes párrafos. Esta triangulación teórica permite obtener fundamentos amplios, así

como directrices y conocimientos precisos para la elaboración de un buen diseño instruccional.

2.1 Marco teórico orientador: el enfoque fenomenológico

Con el fin de establecer una postura epistemológica para la investigación, le hemos dado un sentido cualitativo a través del enfoque fenomenológico, propuesta ampliamente adoptada en la Educación Matemática, pues nos permite estudiar al fenómeno en cuestión desde la identificación de problemas, formulación de preguntas y aproximación cualitativa sobre los procedimientos a estudiar. El enfoque fenomenológico se instaura como un eje importante en el marco teórico de esta investigación pues matiza a la misma desde diferentes aproximaciones, tales como las descritas por Miles y Huberman (1994): interpretativa, antropológica y colaborativa - social.

De otra parte, es de nuestro interés soportar la investigación sobre teoría que permita ir más allá de un proceso de recolección y análisis de datos. Así pues, la selección del enfoque fenomenológico da certeza de nuestra disposición por obtener descripciones, explicaciones e inferencias a través de la investigación, para así fundamentar reflexiones significativas que permitan establecer antecedentes para futuras prácticas docentes en Educación Matemática. Lo relevante del enfoque fenomenológico subyace en las oportunidades que brinda a los investigadores, ya que la reflexión es central en el mismo y permite avanzar hacia un horizonte lleno de implicaciones en los contextos escolares, que promuevan la exploración de problemas, deficiencias y dificultades con el fin de progresar y buscar soluciones útiles para los actores del contexto educativo (Moschkovich y Brenner,

2000). A su vez, la contextualización es clave en el enfoque fenomenológico, ya que sobrepone la exploración y comprensión de los fenómenos en los procesos de enseñanza y aprendizaje y permite estudiar la perspectiva de los involucrados con el fin de construir significados e interpretaciones.

En este orden de ideas, pretendemos darle una visión funcional a la investigación, delimitada por los contextos y las problemáticas en la Educación Matemática desde una perspectiva cualitativa, con el fin de indagar en los fenómenos educativos y reflexionar en torno a los resultados e interpretaciones para dar cuenta de posibles soluciones y descubrimientos en el campo, con el objetivo de analizar los factores contextuales educativos y su impacto en las aulas de matemáticas.

2.2 Marco teórico para la acción: la Teoría de Mediación Semiótica (TMS)

La Teoría de Mediación Semiótica (TMS) en educación matemática representa un modelo pedagógico fundamentado en la teoría de Vygotsky acerca de la mediación de herramientas en la construcción de conocimiento y significados por parte de los estudiantes. Maracci y Mariotti (2013) subrayan en la TMS el papel crucial de los signos matemáticos y los artefactos, considerándolos como mediadores esenciales en la construcción y evolución del conocimiento matemático escolar. Esta teoría se centra en la exploración y análisis del potencial semiótico que surge al emplear artefactos en el contexto educativo, "la efectividad de usar cualquier artefacto como herramienta educativa requiere que el experto - por ejemplo, el profesor de matemáticas - sea consciente de sus potencialidades, en términos tanto de los significados matemáticos emergentes como de

los significados personales emergentes" (Mariotti, 2010, p. 170). Este potencial implica una doble relación semiótica generada por el artefacto: por un lado, se conecta con los significados personales emergentes durante su uso; por otro lado, establece una relación con los significados matemáticos evocados durante dicha utilización (Maracci & Mariotti, 2013). La mediación semiótica aborda y aprovecha este potencial de los artefactos para intervenir pedagógicamente en la construcción de conocimientos matemáticos mediante un ciclo de actividades individuales y colectivas llamado ciclo didáctico.

El ciclo didáctico es un proceso educativo que se sustenta en un ciclo continuo e iterativo de enseñanza y aprendizaje, abarcando tres categorías de actividades: en primer lugar, las "Actividades con el artefacto" en donde el profesor diseña tareas específicas para que los estudiantes utilicen el artefacto, el objetivo es que estos generen esquemas de utilización del artefacto y elaboran significados personales sobre su uso. "Explotar la utilidad del artefacto requiere que el experto orqueste² situaciones donde los estudiantes enfrenten tareas diseñadas en las que se espera que movilicen esquemas de utilización del artefacto y en consecuencia generen significados personales" (Mariotti, 2010, p. 172).

En segundo lugar, "la producción individual de signos" constituye otra fase del ciclo didáctico, donde los estudiantes deben producir registros escritos describiendo y explicando los procedimientos realizados con el artefacto, con el objetivo de lograr objetivar y explicitar los significados personales elaborados en la etapa anterior. Por último, el ciclo propone una "producción colectiva de signos", donde se realizan discusiones colectivas guiadas por el profesor, los estudiantes socializan, comparan y

² La orquestación instrumental es la organización sistemática y deliberada que el maestro ejerce sobre los artefactos con el fin de guiar a los estudiantes en un ambiente de aprendizaje.

discuten sus producciones individuales. Posteriormente, el docente facilita la evolución y enriquecimiento de los significados personales hacia comprensiones matemáticas más profundas, así pues, se llega a un acuerdo colectivo en donde se negocian nuevos significados matemáticos.

En la TMS la mediación del profesor es considerada indispensable, dado que es el principal encargado de promover y gestionar el ciclo didáctico, así como de facilitar el aprendizaje matemático de los estudiantes por medio de diversas actividades y estrategias, el docente tiene la labor de diseñar o seleccionar un artefacto mediador que posea un potencial semiótico relacionado con las necesidades y objetivos de aprendizaje que se quieren llevar a cabo, este artefacto debe contar con actividades que promuevan en los estudiantes esquemas de uso del artefacto, así como la construcción de significados matemáticos personales sobre los conceptos matemáticos involucrados, posterior a esto, el docente debe realizar discusiones colectivas con sus estudiantes con el objetivo de negociar los significados matemáticos personales contruidos, enriqueciendo y evolucionando los a significados matemáticos formales esperados. En síntesis, el rol del docente es fundamental en la TMS debido a que, es el encargado de gestionar y promover la internalización y apropiación de los conocimientos y significados matemáticos por parte de los estudiantes a través de la mediación semiótica del artefacto.

Mariotti y Bartolini Bussi (2008) enfatizan la relevancia de la mediación semiótica al decir que "ofrece una perspectiva innovadora para estudiar los procesos de enseñanza y aprendizaje matemático, proporcionando tanto descripciones detalladas de interacciones en el aula como herramientas conceptuales para el análisis e interpretación de los procesos de

construcción de conocimiento matemático. En síntesis, la TMS es un modelo pedagógico que aprovecha el potencial de los artefactos para mediar los procesos de construcción del conocimiento matemático por parte de los estudiantes. Se basa en un ciclo de actividades individuales y colectivas donde los estudiantes desarrollan significados personales al usar artefactos, los explicitan en registros escritos y luego los negocian colectivamente hacia significados matemáticos formales. En este proceso, la mediación del docente es indispensable para gestionar el ciclo didáctico, seleccionar el artefacto mediador adecuado, promover la evolución de significados y propiciar la interiorización del conocimiento matemático por parte de los estudiantes.

2.3 Marco de dominio específico: sobre el pensamiento lógico matemático, el pensamiento computacional y la programación en Python

Para la construcción del marco de dominio específico consideramos nociones conceptuales que abordan los objetivos de la investigación con gran detalle, así pues, surgen algunos referentes en este marco:

- Pensamiento lógico matemático
- Pensamiento computacional
- Programación computacional

Estos referentes servirán como horizonte para la investigación y permitirán abordar conceptos específicos necesarios para su desarrollo:

2.3.1 Pensamiento lógico matemático

El pensamiento lógico matemático se refiere a las habilidades que el estudiante desarrolla y le permiten resolver operaciones, analizar situaciones, pensar de forma crítica y reflexiva y comprender el mundo que los rodea a través del razonamiento lógico y los conceptos matemáticos. Este conjunto de habilidades es fundamental en la Educación Matemática, pues busca promover el desarrollo de la capacidad de razonamiento lógico de los estudiantes a través de estrategias didácticas y metodologías que permitan a los estudiantes construir el conocimiento de formas significativas.

Es pues necesario dejar claro que el pensamiento lógico no es parte del pensamiento matemático, sino que el pensamiento lógico apoya y perfecciona el pensamiento matemático, y con éste –en cualquiera de sus tipos– se puede y se debe desarrollar también el pensamiento lógico. (MEN, 2006)

En este sentido, la construcción del conocimiento matemático se beneficia del pensamiento lógico-matemático en la medida en que potencia habilidades propias que se relacionan con el actuar de las matemáticas. Piaget (1970) afirmaba que el pensamiento lógico es necesario para la culminación de las estructuras matemáticas, sin embargo, no es suficiente, pues las acciones asociadas al pensamiento lógico matemático se constituyen a través de procesos cognitivos que movilizan el saber matemático en los estudiantes. En este orden de ideas, desde la teoría Piagetiana y su relación con la Educación Matemática,

Corredor (2011) propone cuatro procesos cognitivos implícitos en el pensamiento lógico matemático:

1. **Representación:** capacidad de representar objetos y situaciones matemáticas mediante símbolos, gráficos, diagramas, entre otros.
2. **Abstracción:** capacidad de identificar patrones y generalizar a partir de situaciones específicas.
3. **Creatividad:** capacidad de generar nuevas ideas y soluciones a problemas matemáticos.
4. **Demostración matemática:** capacidad de justificar y demostrar la validez de un resultado matemático.

Es menester de la Educación Matemática proveer a los estudiantes de los recursos conceptuales suficientes para su íntegro desarrollo, es por esto que los agentes educativos reflexionen en torno a las metodologías empleadas para enseñar matemáticas, considerando alternativas modernas y relacionadas al devenir de la sociedad digital. Por tanto, es de nuestro principal interés mostrar la manera en que la programación moviliza los cuatro procesos cognitivos y asimismo beneficia el pensamiento matemático de los estudiantes. El Ministerio de Educación Nacional (2006) establece que si bien los Lineamientos Curriculares subdividen al pensamiento matemático en cinco tipos, el pensamiento lógico se encarga de potencializar y refinar a cada uno de los cinco pensamientos de forma transversal.

Así pues, la presente investigación pretende dar cuenta de la relación entre el pensamiento lógico matemático y el computacional, a través del uso de la programación en Python con el fin de caracterizar el impacto de la programación sobre los procesos cognitivos que moviliza el pensamiento lógico matemático.

2.3.2 Pensamiento computacional

Según Jeannette Wing (2006), una reconocida científica de la computación y promotora del pensamiento computacional en las aulas de clase a nivel internacional, el pensamiento computacional es una habilidad fundamental para toda persona del común, pues este implica resolver problemas, diseñar sistemas y comprender el comportamiento humano aprovechando los conceptos fundamentales de la informática.

Wing propuso un conjunto de habilidades implícitas en el pensamiento computacional y las define de la siguiente manera:

1. **Abstracción:** la habilidad para abstraer propiedades y particularidades a la vez que centrarse en los aspectos clave de un problema.
2. **Algoritmia:** la capacidad de desarrollar un conjunto de pasos secuenciales, ordenados y lógicos para resolver un problema determinado.
3. **Descomposición:** la habilidad de descomponer un problema en problemas más pequeños o partes más intrínsecas para lograr una solución.
4. **Reconocimiento de patrones:** la capacidad de identificar patrones y tendencias en la información o el problema a desarrollar.

Tal como se ha presentado antes en esta investigación, las habilidades digitales requeridas en el mundo actual no solo influyen al campo de la Educación Matemática, sino que directamente alteran las necesidades curriculares respecto a la importancia del pensamiento crítico y analítico de los estudiantes. La caracterización del pensamiento computacional permite entrever su relación con el pensamiento matemático, y aunque esta idea será abordada de forma transversal al desarrollo de la investigación, es pertinente aclarar que tanto el pensamiento computacional como el matemático poseen habilidades y competencias en común, las cuales convergen y dan paso a un sinnúmero de oportunidades para estudiar su relación y promover nuevas investigaciones a la misma.

“Wing también lo definió como un conjunto de habilidades y destrezas (“herramientas mentales”), habituales en los profesionales de las ciencias de la computación, pero que todos los seres humanos deberían poseer y utilizar para “resolver problemas”, “diseñar sistemas” y, sorprendentemente, “comprender el comportamiento humano”. Por tanto, todo lo relacionado a la programación, robótica y la tecnología debería formar parte de la educación de todo ser humano” (Adell Segura et al., 2019).

En este sentido, la importancia de la inclusión del pensamiento computacional en la educación se hace cada vez más notoria, pues se refiere a un conjunto de habilidades cognitivas fundamentales inspiradas en el campo de la informática y destinadas a abordar problemas complejos y tomar decisiones informadas, lo cual es clave en los procesos de enseñanza y aprendizaje de las matemáticas. Es por esto que, a partir de la información aquí presentada, abordaremos la investigación considerando en todo momento la

importancia del pensamiento computacional y su influencia en el desarrollo del pensamiento matemático a través de uno de sus elementos más característicos: la programación computacional.

2.3.3 Programación computacional

La programación se instaure como una de las formas óptimas para el desarrollo del pensamiento computacional en los estudiantes, pues funge como un conjunto de habilidades procedimentales que permiten dar solución a un problema específico. A través de la programación, los estudiantes pueden mejorar sus habilidades de pensamiento lógico, resolución de problemas, creatividad y adaptación a la sociedad actual y futura.

“La adopción de la programación, especialmente en el plan de estudios de matemáticas, se basa en el argumento de que la programación puede apoyar la resolución de problemas y el pensamiento lógico, ambas habilidades importantes para las matemáticas, así como el objetivo general de evitar una nueva materia” (Bocconi et al., 2018; Education (CECE), 2017).

En este sentido, la programación provee a la Educación Matemática de un contexto práctico y aplicado para los conceptos abstractos de las matemáticas, ofreciendo a los estudiantes una amalgama de oportunidades para implementar conocimiento matemático en situaciones que involucren resolución de problemas, desarrollo del pensamiento lógico y algorítmico y visualización de conceptos matemáticos a través de distintas representaciones.

Según Papert (1980), programar no solo permite al alumno comunicarse con el ordenador, sino también explorar las dinámicas que rigen sus propios pensamientos, asentar el razonamiento lógico a través de la sistematización y aumentar su capacidad de autocrítica por medio de la corrección de sus errores. A través de la retroalimentación y la reflexión constante, es posible advertir de concepciones erróneas de los estudiantes y trabajar sobre ellas en futuras prácticas educativas.

Así pues, es un reto en la edad moderna el digitalizar los procesos educativos, por tanto, es pertinente considerar a la programación como una de las habilidades relevantes dentro del proceso de alfabetización digital. Tal como lo afirma Llorens (2015), la programación es algo que parece estar fuera del alcance de las personas del común, sin embargo, no se trata de convertir a los alumnos en expertos en informática, sino más bien de ofrecerles la oportunidad de desenvolverse mejor en un mundo que se encuentra en constante cambio y evolución.

Python como lenguaje de programación para la enseñanza de las matemáticas

Python es un lenguaje de programación de alto nivel, interpretado, dinámico y multiplataforma. Tiene sus orígenes a finales de los años ochenta a manos del programador alemán Guido van Rossum. Este lenguaje de programación es conocido y popular en el campo de la tecnología debido a su legibilidad y sintaxis simple, facilidad de uso y versatilidad, permitiendo el desarrollo de aplicaciones simples hasta programas con altos niveles de complejidad, esta versatilidad lo convierte en un candidato ideal para enseñar matemáticas y desarrollar aplicaciones en diversos campos.

Python es el lenguaje de programación escogido para la presente investigación por las siguientes razones:

1. **Facilidad de aprendizaje:** Python es fácil de aprender y utilizar, lo que lo hace accesible para estudiantes de todas las edades y niveles de habilidad.
2. **Versatilidad:** Python es un lenguaje de programación versátil que se puede utilizar para resolver una amplia variedad de problemas matemáticos, desde la aritmética hasta el cálculo diferencial e integral.
3. **Recursos educativos:** Python cuenta con una gran cantidad de recursos educativos, como cursos en línea, tutoriales y videos de YouTube, que pueden ayudar a los estudiantes a aprender matemáticas de manera interactiva.
4. **Uso en la industria y la investigación:** Python es un lenguaje de programación utilizado en la industria y la investigación, lo que significa que los estudiantes pueden aplicar lo que aprenden en la escuela en situaciones del mundo real.

Para el análisis de esta investigación se abordarán algunos conceptos propios del lenguaje de programación Python los cuales son:

- **Operadores de decisión o estructuras de control de flujo:** Los operadores de decisión en Python son estructuras de control que permiten ejecutar diferentes bloques de código basándose en condiciones específicas. Estos operadores ayudan a tomar decisiones dentro de un programa, controlando el flujo de ejecución según

las condiciones evaluadas como verdaderas o falsas. A continuación, se describen los principales operadores de decisión en Python:

- **If:** el operador *if* ejecuta un bloque de código si la condición especificada es verdadera.
- **Elif:** el operador *elif*, siempre es utilizado después de un operador *if* y se ejecuta cuando el bloque de código *if* es falso.
- **Else:** el operador *else* se ejecuta cuando los operadores *if* y *elif* son falsos.
- **Operadores de iteración o bucles:** los operadores de iteración en Python permiten ejecutar un bloque de código repetidamente mientras una condición es verdadera o a través de una secuencia de elementos. Estos operadores son fundamentales para realizar tareas repetitivas y para procesar colecciones de datos. Los principales operadores de iteración en Python son:
 - **While:** el bucle *while* se utiliza para ejecutar un bloque de código repetidamente mientras una condición específica sea verdadera. Si la condición nunca se evalúa como falsa, el bucle se ejecutará indefinidamente, lo que puede causar un bucle infinito.
 - **For:** el bucle *for* se utiliza para iterar sobre una secuencia (como una lista, tupla, diccionario, conjunto o cadena) y ejecutar un bloque de código para cada elemento de la secuencia.

3. Marco metodológico

El marco metodológico de la presente investigación supone un eje central puesto que enmarca las prácticas pedagógicas inmersas en el proyecto propuesto, así pues, serán detalladas las corrientes teóricas asociadas a los intereses prácticos de la investigación y su relación con el marco teórico.

3.1 Sobre la estructura metodológica

Este capítulo se centra en describir los elementos que caracterizan la metodología propuesta para la investigación, para la cual se precisará su respectiva naturaleza, así como, su enfoque y aproximación, se detallarán los recursos (información y análisis de datos), la estrategia cualitativa (estudio de caso) y los recursos (captura de información) en donde se desliga: el escenario, los informantes, los eventos y por último los procesos que se llevarán a cabo en la investigación. A continuación, se mostrará la ficha técnica que sintetiza y evidencia el panorama general de la metodología a implementar:

Tabla 2. *Ficha panorámica metodológica*

Ficha panorámica: metodología		
• Naturaleza y Enfoque		IBD - Fenomenológico
• Aproximación		Interpretativa
• Recursos (Información y análisis de datos)		Diseño secuencial cualitativo
• Estrategia Cualitativa		Estudio de caso
• Recursos Captura información	Escenario	Salon de informatica, colegio privado
	Informantes	7 estudiantes de noveno a undécimo grado
	Eventos	Implementación de recursos Google Colab y Python

	Recursos	Actividades que sirvieron como instrumento de análisis mediadas por Google Colab mediante el uso del lenguaje de programación Python.
• Análisis local de los datos		Descripción de la actividad
		Desarrollo de la actividad
		Objetivo
		Análisis de la información
		Análisis retrospectivo

Nota. Tomado y adaptado de: Bonelo (2023)

3.2 Naturaleza de la investigación

Para propósito de esta metodología, la investigación basada en diseño (IBD) (Cobb et al., 2017; De Benito Crosetti & Salinas Ibáñez, 2016) se impone como principal referente, ya que beneficia los objetivos propuestos al inicio de esta, caracterizándose por ser de naturaleza participativa y colaborativa, siempre desde un enfoque cualitativo. La IBD permite estudiar el ejercicio educativo con intenciones reflexivas y de mejora, pues se preocupa por dar respuesta a problemas concretos en contextos reales, con intenciones de estudiar la naturaleza de los procesos educativos para resignificar conceptos desde teorías del aprendizaje. Tal como lo afirman Wang y Hannafin (2005), se puede resumir a la IBD bajo 5 características fundamentales: pragmática, fundamentada, interactiva, interactiva y flexible, integrada, contextual. Así pues, la IBD nos ofrece algunos principios:

1. **Iteratividad**, respecto al proceso de desarrollo y participación y constante de los involucrados, pues en la medida en que se ejecuta ofrece la posibilidad de revisión, reformulación y retroalimentación necesaria.

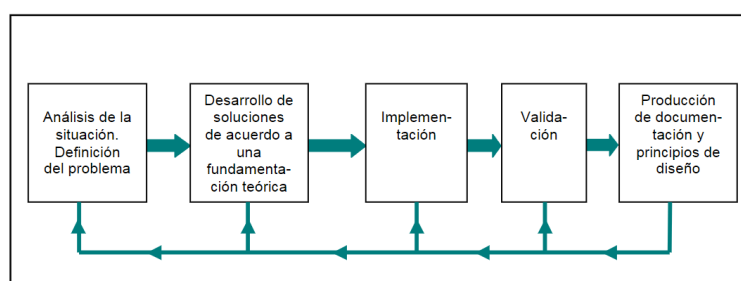
2. **Reflexividad**, pues su enfoque es práctico y fija su interés en situaciones de aula no preconcebidas, así, da paso a problemas relevantes del campo educativo desde una perspectiva práctica.
3. **Participatividad**, ya que la colaboración entre los agentes educativos es clave para el proceso reflexivo de la misma, todos hacen parte de un equipo, no es cuestión exclusiva del investigador o del docente.

La implementación en esta investigación se enmarca en la IBD pues la reflexión es una idea central y transversal durante el desarrollo del proyecto, ya que la perspectiva pragmática de la IBD nos permite analizar las interacciones de aula en un contexto real con vistas hacia la mejora de las prácticas educativas dentro del aula de matemáticas. Es de especial importancia resaltar que no es de nuestro interés ejecutar una investigación cuantitativa con enfoque hacia la generalización de los resultados, sino más bien una investigación centrada en la práctica, con intenciones de caracterizar el rol de la programación computacional y su papel en el aula de matemática. Por ende, el énfasis central es la solución de problemas y la construcción del conocimiento dirigido hacia el diseño, desarrollo y evaluación del proceso educativo, que permitan desplegar nuevos principios y enfoques para investigaciones futuras con una base referencial sólida.

Para la investigación presente, uno de los modelos (Fig. 1) más representativos de su proceso es el propuesto por Reeves (2000; 2006), pues da cuenta del desarrollo de la investigación desde el análisis de la situación a partir de una perspectiva cualitativa para posterior a ello definir el problema. Así pues, se encamina la investigación hacia posibles

soluciones basadas en fundamentación teórica pertinente para abordar la problemática, considerando aspectos de diversas índoles en Educación Matemática, en consecuencia, se dará paso a la implementación, validación y finalmente a la recolección de información seguida de una reflexión y rediseño considerando los resultados recolectados.

Figura 1. *Proceso de la investigación de desarrollo*



Nota. Tomado y adaptado de: Reeves (2000, en De Benito, 2006).

3.2.1 Enfoque: fenomenológico

La IBD en Educación Matemática se beneficia de enfoques de origen cualitativo no predictivo, es decir que, la interpretación de los resultados será fundamental para el análisis de los fenómenos educativos estudiados. Por tanto, el enfoque fenomenológico permitirá dar una mirada cualitativa a través de la descripción, interpretación, explicación y cuestionamiento constante. Así como lo describe Camargo (2019), una investigación con este enfoque pretende ir más allá de solo dar cuenta de la problemática y describirla, pues concibe el aula de clases de matemáticas como un entorno dinámico y cambiante. Por consecuencia, es de nuestro profundo interés explorar oportunidades que permitan trazar un horizonte hacia el progreso y dilucidar soluciones útiles a los agentes educativos según el contexto en el que se encuentren.

3.2.2 Aproximación: interpretativa

La aproximación adoptada para esta investigación es la aproximación interpretativa, la cual según González y Gómez (2018), se centra en comprender los significados matemáticos de los estudiantes subyacentes a sus acciones, interacciones y discursos, tiene como objetivo interpretar los diferentes significados sin ser juzgados, es decir, sin evaluarlos como correctos o incorrectos, al contrario, se busca comprender cómo los estudiantes entienden e interpretan los conceptos e ideas matemáticas. En la aproximación interpretativa los estudiantes son considerados como conjuntos de signos, en donde sus acciones, interacciones y discurso son vistos como expresiones simbólicas que revelan las comprensiones y experiencias del individuo.

La aproximación interpretativa cuenta con dos enfoques que pueden ser utilizados en la interpretación de los significados de los estudiantes, el primero de estos es la aproximación interpretativa fenomenológica, la cual se basa en comprender y conocer el fenómeno en su totalidad, el docente es el encargado de acercarse al estudiante para entender e interpretar las concepciones y significados desde la perspectiva de este, en esta aproximación, no se busca describir los significados de manera superficial, al contrario se busca captar a profundidad toda su esencia. Por otro lado, el segundo enfoque es la aproximación interpretativa interaccionista, la cual, al contrario de la anterior, no busca abordar el fenómeno en su totalidad, sino que se enfoca en seleccionar y analizar ciertas interacciones y acciones que destacan para el objetivo de la investigación, este enfoque se encarga de analizar grupos de acciones que pueden ayudar a revelar patrones o significados particulares de los estudiantes del fenómeno en estudio.

3.2.3 Recursos (información y análisis de datos)

Con el fin de registrar información para su posterior análisis cualitativo, hemos escogido el procedimiento de diseño secuencial cualitativo, pues en pro de recolectar información es pertinente disponer de procesos iniciales que permitan identificar situaciones centrales en la investigación y posterior a ello efectuar el análisis con base en los resultados empleando procesos relacionados a la fundamentación teórica establecida. Debido a la naturaleza de nuestra investigación y objeto de estudio, la programación computacional supone una práctica emergente en las aulas de matemáticas, por lo que la tarea exploratoria a través de procesos cualitativos será esencial en el análisis de los efectos de su implementación en el contexto escolar. Así pues, los objetivos establecidos nos permitirán generar hipótesis que pueden ser probadas mediante procedimientos de análisis cualitativo (Camargo, 2019).

A su vez, este tipo de análisis nos permite investigar situaciones de aula de forma empírica, combinando procedimientos tanto cualitativos como cuantitativos (Kelle y Boichholtz, 2015). En este orden de ideas, la planificación y diseño tendrán fundamentación en recursos que permitan retroalimentación instantánea y recolección de la información en entornos virtuales que ofrezcan la posibilidad de implementar situaciones previstas desde un enfoque investigativo cualitativo secuencial.

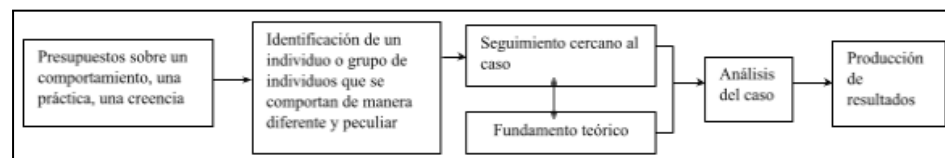
3.2.4 Estrategia cualitativa: estudio de caso

El estudio de caso se centra en contar y describir en detalle una situación real o una historia de un caso específico que es interesante o llamativo para analizar desde el punto

de vista investigativo, es por esto que esta investigación abordará como enfoque el estudio de caso, partiendo del hecho de que nuestro interés está en abordar la aproximación interpretativa fenomenológica mencionada anteriormente, en donde estos dos enfoques se relacionan y en conjunto permitirán comprender a mayor profundidad la singularidad y complejidad de un caso específico, en lugar de buscar patrones generales.

Para la realización y ejecución de la estrategia “estudio de caso”, se adoptó por la estructura propuesta por Camargo (2019) la cual es:

Figura 2. Plan de ejecución de la estrategia “estudio de caso”



3.3 Recursos: captura información

3.3.1 Escenario ¿Dónde?

El desarrollo de la presente investigación tomó como escenario las instalaciones de un establecimiento educativo de carácter privado ubicado en el sector oeste de la ciudad de Cali. Específicamente, la implementación del diseño propuesto tuvo lugar en la sala de informática de primaria y se contó con la participación de 7 estudiantes matriculados en dicha institución.

3.3.2 Informantes ¿A quién?

En esta investigación cualitativa se ha seleccionado una muestra de siete estudiantes con un buen nivel académico en matemáticas para explorar el uso del lenguaje de programación Python. Esta decisión se fundamenta en varios aspectos clave de la metodología cualitativa. El objetivo principal de esta investigación es obtener una comprensión profunda y detallada de las experiencias y conocimientos de estos estudiantes destacados al utilizar Python en el contexto matemático. La metodología cualitativa permite explorar en profundidad los matices y particularidades de sus experiencias.

Se optó por una muestra intencional de estudiantes con alto rendimiento en matemáticas porque poseen características necesarias para proporcionar información relevante sobre el uso de Python en matemáticas. Dado el interés por caracterizar en un enfoque poco explorado en educación matemática este tipo de muestreo es adecuado cuando se busca una comprensión detallada de un fenómeno específico.

En la investigación cualitativa, el tamaño de la muestra se determina en función de la saturación teórica, es decir, el punto en el cual no emergen nuevos datos significativos. A menudo, este punto se alcanza con un número reducido de participantes cuando la muestra es homogénea y específica.

La investigación cualitativa valora la calidad y profundidad de los datos obtenidos. Con una muestra pequeña, es posible realizar un análisis exhaustivo y detallado de las experiencias de los participantes, lo que enriquece los hallazgos del estudio. Además han

demostrado que muestras pequeñas pueden ser efectivas para explorar fenómenos complejos y específicos. Estos estudios respaldan la viabilidad y la justificación de utilizar un número reducido de participantes cuando se busca un análisis detallado y profundo.

De otra manera y desde una perspectiva estadística, el enfoque cualitativo y la selección de una muestra pequeña no buscan la generalización estadística, sino la comprensión profunda de un fenómeno específico. Sin embargo, en investigaciones cualitativas, el objetivo no es necesariamente obtener resultados que sean representativos de una población más amplia, sino proporcionar una visión detallada y matizada de las experiencias de un grupo particular. La validez de los hallazgos cualitativos radica en la profundidad de la información obtenida, así como en la pertinencia y relevancia de los datos en relación con la pregunta de investigación.

3.3.3 Eventos ¿Cuándo?

A continuación, se presenta el cronograma de actividades con sus respectivos nombres que detallan las etapas de la investigación:

La implementación y recolección de datos se llevó a cabo en 4 sesiones de clase. Tres de estas sesiones se realizaron en modalidad presencial, mientras que la cuarta sesión se llevó a cabo de manera virtual con el objetivo de ayudar a los estudiantes a resolver cualquier duda que pudieran tener en la elaboración de la actividad 2.

Tabla 3. Cronograma de actividades de la investigación

[illegible]

3.3.4 Procesos ¿haciendo qué?

Durante las actividades, se llevaron a cabo varios momentos de intervención con los estudiantes. Inicialmente, los docentes ofrecieron una orientación introductoria a Python, guiando a los estudiantes desde los conceptos fundamentales de programación. En esta primera intervención, se abordaron aspectos como la definición de variables y la introducción a funciones básicas del lenguaje, como *input* y *print*. Además, se exploraron las operaciones básicas y el uso de operadores de decisión en Python. Posteriormente, los estudiantes fueron guiados para llevar a cabo la actividad A0, diseñada para que pudieran interactuar con la herramienta y poner en práctica los conocimientos recién adquiridos. Finalmente, durante el proceso de exploración por parte de los estudiantes, los docentes intervinieron con preguntas específicas para cada ejercicio, fomentando así un aprendizaje más personalizado y guiado.

Otro momento clave de intervención, se llevó a cabo durante la actividad dos, aquí los docentes retomaron algunos conceptos previamente abordados en la sesión anterior y profundizaron en temas como los operadores de iteración, ilustrando su uso con ejemplos concretos. Posteriormente, facilitaron a los estudiantes el desarrollo de la primera actividad, ofreciendo orientación y resolviendo cualquier pregunta o inquietud que pudiera surgir. Al concluir la actividad, los docentes brindaron un resumen de los aspectos más relevantes y proporcionaron recomendaciones para la próxima sesión.

Finalmente, en la última sesión, se llevó a cabo una reunión adicional donde los docentes proporcionaron a los estudiantes las instrucciones para llevar a cabo la actividad

propuesta, A2. Durante esta actividad, se optó por una participación mínima por parte de los docentes, ya que se había diseñado para que los estudiantes trabajarán de manera autónoma. No obstante, algunos expresaron ciertas dudas que fueron abordadas con la ayuda de los docentes.

3.3.5 Recursos: Actividades

Se diseñaron y elaboraron tres actividades, denominadas A0, A1 y A2, que sirven como instrumentos para el análisis. Los detalles completos de cada actividad se encuentran en los anexos correspondientes. Cada una de estas actividades está estructurada en tres secciones principales:

- **Operaciones aritméticas con Python:** esta sección introduce a los estudiantes a las operaciones matemáticas básicas utilizando Python, permitiéndoles aplicar y reforzar sus conocimientos de aritmética a través de la programación.
- **Operadores de decisión:** en esta parte, los estudiantes aprenden a utilizar estructuras de control de flujo como *if*, *else* y *elif* para tomar decisiones basadas en condiciones específicas, desarrollando su capacidad para implementar lógica condicional en sus programas.
- **Operadores de iteración:** la última sección se enfoca en el uso de bucles *for* y *while* para realizar operaciones repetitivas, enseñando a los estudiantes cómo manejar tareas repetitivas y procesar colecciones de datos de manera eficiente.

Estas actividades están diseñadas para evaluar y analizar el desarrollo de habilidades en programación, pensamiento lógico-matemático y pensamiento computacional de los estudiantes a lo largo del curso.

3.3.6 Sobre la estructura del análisis

Para la redacción del análisis de cada hoja de trabajo se consideraron varios aspectos clave: la descripción de la actividad, el desarrollo de la actividad, el objetivo, las condiciones de aplicación y el análisis de la información.

- **Descripción de la actividad:** esta sección proporciona una visión general de la actividad, destacando la intención y el interés específico al que está dirigida.
- **Desarrollo de la actividad:** en esta sección se ofrece un detallado contexto en el que se desarrolló la actividad, incluyendo información sobre el escenario, los participantes y el proceso de implementación.
- **Objetivo:** esta parte especifica el propósito central de la actividad.
- **El análisis de la información:** el proceso de análisis de datos consistió en la descripción, comparación y conexión con teorías emergentes, así como la identificación de patrones en los datos recopilados. Esta fase tiene como objetivo convertir la información bruta en datos con significado. Dado que este estudio se sitúa dentro de la Investigación Basada en el Diseño (IBD), el análisis abarca características de análisis local y retrospectivo, que son comunes en las

investigaciones cualitativas que siguen este enfoque, similar a lo que se realiza en los experimentos de enseñanza.

- **La fase del análisis retrospectivo:** en esta etapa, a partir del análisis local de cada sesión, se identifican regularidades y variaciones según lo propuesto. Se detalla lo que se ha logrado y lo que está fuera de lo previsto. Una característica distintiva del análisis retrospectivo es que las variables de análisis son cualitativas, lo que permite un examen más detallado de la investigación, integrando continuamente teoría y práctica.

4. Análisis de datos

4.1 Sobre el análisis

En este capítulo se presentan los análisis obtenidos de tres actividades realizadas en Google Colaboratory utilizando el lenguaje de programación Python, de acuerdo con los objetivos establecidos al comienzo de la investigación. Este análisis permitió evidenciar las diferentes heurísticas, potencialidades y dificultades experimentadas por las estudiantes a medida que avanzaban y desarrollaban las actividades propuestas. Además, se utilizó una encuesta para la recolección de datos, la cual permitió identificar la percepción de las estudiantes con relación a la programación y en particular con el uso de Python en la educación matemática, la encuesta se realizó al comienzo del estudio de caso, antes de que las estudiantes tuvieran algún acercamiento con las actividades y la programación, y se

llevó a cabo una nueva encuesta después de completar las actividades. Esto permitió analizar el cambio en la percepción de las estudiantes con relación a la programación y su impacto en la educación matemática. Las actividades presentadas en esta investigación, así como la encuesta para la recolección de datos y las operaciones cognitivas contaron con la mediación y participación de los docentes, quienes colaboraron y trabajaron en la construcción del conocimiento de los estudiantes.

4.2 Encuesta previa a las actividades

Tomando en consideración el grado de escolaridad de los estudiantes y el contexto en el cual nos encontramos, se decidió crear una encuesta previa sobre nociones y creencias de los estudiantes en torno a la programación computacional. Dicha encuesta fue realizada con el objetivo de discernir el conocimiento general de los estudiantes sobre la programación y sus expectativas en cuanto al desarrollo de la investigación. No obstante, la encuesta fue realizada tras iniciar la primera sesión y habiendo hecho la introducción a la programación desde una perspectiva conceptual y con ejemplos de aplicación en la vida real, es decir que, sin un conocimiento teórico ni procedimental, los estudiantes debían contestar tomando en cuenta la breve introducción presentada sobre nociones básicas de la programación y sus usos. Esto fue tomado en cuenta para la estructuración de las sesiones implementadas y el diseño de las actividades, pues uno de los ejes teóricos centrales es la IBD (Cobb et al., 2017; De Benito Crosetti & Salinas Ibáñez, 2016), la cual nos permitió dar un horizonte al diseño tomando en consideración las respuestas de esta encuesta. Así pues, las respuestas obtenidas se presentan en las siguientes tablas:

Tabla 4. Respuestas de la pregunta 1 de la encuesta introductoria

¿Qué entiendes por "programación computacional" y cómo crees que se relaciona con las matemáticas?	
Estudiante	Respuesta
E1	La programación es la matemática actual. Ya que hace posible que tengamos todas las aplicaciones y páginas en la actualidad
E2	Lo que entiendo de la programación es que se utiliza para darle instrucciones a una máquina y que ésta las ejecute. Yo creo que estos programas utilizan lenguaje matemático
E3	La programación computacional es configurar algo para que siga ciertas órdenes. Teniendo en cuenta que las órdenes que se mandan son por algoritmos lo podemos asociar con las matemáticas ya que para resolver un problema matemático necesitamos ciertos algoritmos para solucionarlo.
E4	Entiendo que la programación computacional es darle instrucciones a un computador para que cumpla diversas funciones y creo se requiere del uso del lenguaje matemático para programar
E5	Es la forma de comunicarse y darle órdenes a un computador y se relaciona con las matemáticas en la forma de que la programación busca solucionar un problema
E6	Es como darle instrucciones a una computadora para que realice ciertas tareas, enseñándole cosas específicas siguiendo pasos detallados y se relaciona con las matemáticas ya que para poder lograrlo y poder programar se utilizan muchos conceptos matemáticos
E7	Programar es como dar órdenes para lograr un objetivo

Las respuestas de la pregunta 1 permiten ver la diversidad de perspectivas sobre la programación y cómo los estudiantes la relacionan con las matemáticas. En general, se reconocen patrones en las respuestas, pues apuntan hacia la concepción de que la programación consiste en dar instrucciones u órdenes para lograr un objetivo específico, ya sea mediante “configuraciones” o mediante la comunicación con una computadora.

Además, parece haber una idea general sobre la relevancia del uso del lenguaje matemático y las matemáticas en la programación, pues varias respuestas hablan sobre algoritmos y el uso de las matemáticas en la actualidad.

Se identifica una relación entre la programación y las matemáticas en las respuestas de los estudiantes. No obstante, es descrita de forma superficial, así como también se reconoce la necesidad de conceptos matemáticos, como algoritmos, en el proceso de programación. Es decir que los estudiantes tienen una comprensión básica pero fundamental sobre la inclusión de la programación en las matemáticas, aunque podría profundizar en ejemplos concretos para mejorar esta comprensión, pues si bien reconocen la importancia del lenguaje matemático en la programación, podrían beneficiarse de ejemplos concretos sobre aplicaciones de las matemáticas y programación en contextos reales. La precisión de la relación podría lograrse a través de actividades que vinculen de manera explícita los conceptos matemáticos con la programación computacional.

Tabla 5. *Respuestas de la pregunta 2 de la encuesta introductoria*

¿Has tenido alguna experiencia previa en programación? Si es así, ¿qué lenguajes has utilizado y qué actividades has realizado?	
Estudiante	Respuesta
E1	Si, en una universidad utilizamos Python para un taller
E2	Mi papá me ha enseñado Python y HTML
E3	No
E4	Sí, creo que era Python
E5	No

E6	Si he tenido experiencia previa con Python, pero no me acuerdo de que fue lo que hicimos
E7	Me gusta el tema, pero no he experimentado con ningún programa

La pregunta 2, más que ser objeto de análisis se convierte en una guía para el diseño, pues nos permitió reconocer la experiencia que poseían los estudiantes en el campo de la programación, en este sentido, cuatro de las siete estudiantes tenían familiaridad con la programación, en específico Python, lo cual facilitó la tarea pedagógica durante las sesiones, mientras que las tres restantes no tenían conocimiento, pero si disposición hacia el aprendizaje de este nuevo conocimiento.

Tabla 6. Respuestas de la pregunta 3 de la encuesta introductoria

¿Consideras que la programación puede ser un instrumento útil para comprender y aplicar conceptos matemáticos? Por favor, explica brevemente tu respuesta.	
Estudiante	Respuesta
E1	Si, ya que permite construir cosas por medio de elementos matemáticos
E2	Yo creo que sí ya que programando podemos solucionar problemas asignándoles a una máquina, haciendo el proceso más eficiente.
E3	Si, porque nos permite analizar de una forma más breve los conceptos y es mucho más sencillo aplicar las fórmulas
E4	Sí, creo que se puede utilizar la programación para resolver problemas matemáticos
E5	Si, ya que esté se maneja con algoritmos, así como los problemas matemáticos y además la programación y las matemáticas buscan llegar a lo mismo, darle solución a un problema
E6	Si creo que es útil para las matemáticas ya que a través de la programación puedes ver cómo se aplican conceptos matemáticos en problemáticas de la vida real

E7	Sí me parece que es una forma la cual llega a muchas respuestas y soluciones, ayuda mucho
----	---

Las respuestas de los estudiantes sugieren un consenso general sobre la utilidad de la programación en la comprensión y aplicación de conceptos matemáticos, pues reconocen que hay elementos matemáticos presentes en la programación y en la construcción de algoritmos y soluciones a problemas. Se evidencian respuestas en torno a la eficiencia de la programación para la solución de problemas, aunque no hay relaciones explicitadas sobre los objetos matemáticos aquí presentes, sin embargo, las respuestas resaltan la similitud de los objetivos entre la programación y las matemáticas, mostrando cómo la resolución de problemas es una habilidad en común para ambas disciplinas.

Además, queda clara la idea sobre cómo la aplicación de fórmulas se puede dar a través de la programación, lo que implica una percepción sobre cómo la programación puede facilitar el quehacer matemático. Estas respuestas fueron un insumo relevante para encaminar el diseño de las actividades posteriores, considerando la habilidad de resolución de problemas y razonamiento lógico como ejes centrales que movilizaron la aplicación de conceptos matemáticos en los problemas específicos de programación.

Tabla 7. *Respuestas de la pregunta 4 de la encuesta introductoria*

¿Qué áreas específicas de las matemáticas crees que podrían beneficiarse más de la incorporación de la programación en Python?	
Estudiante	Respuesta
E1	Los arquitectos, ingenieros y programadores en general
E2	Estadística

E3	La solución de problemas
E4	El álgebra
E5	A la solución y comprensión de problemas
E6	Creo que áreas como la geometría y el álgebra se podrían beneficiar con la incorporación de la programación
E7	La estadística y el cálculo

Las respuestas revelan variedad de ramas de las matemáticas sobre las cuales los estudiantes consideran que la programación puede tener un efecto positivo. Se destacan áreas específicas como la estadística, el álgebra, la geometría y el cálculo, además, es claro que hay una idea presente sobre la habilidad de resolución de problemas. Estas respuestas evidencian no solo las creencias de los estudiantes sino también sus intereses, pues, aunque unas respuestas son más específicas que otras, todas están de acuerdo en que áreas muy reconocidas de las matemáticas podrían ser beneficiadas por la programación.

Tabla 8. Respuestas de la pregunta 5 de la encuesta introductoria

¿Qué expectativas tienes sobre cómo la enseñanza de la programación en Python podría enriquecer tu experiencia de aprendizaje en matemáticas?	
Estudiante	Respuesta
E1	Nos va a ayudar mucho, ya que las carreras tienen materias donde se requiere programar, lo cual nos acercará a tener un mejor rendimiento en la universidad y la vida
E2	Espero poder aprender a programar diferentes programas con diferentes funciones y a la vez encontrar la relación que tiene la programación con las matemáticas y poder implementar esto cuando esté estudiando
E3	Aprender sobre la programación ya que sirve para el futuro y es importante por la manera en que la población ha evolucionado

E4	Creo que me puede ayudar a encontrar soluciones a problemas matemáticos que sin esta herramienta pueden ser más complicados
E5	Pues pienso que no solo me puede servir a mejorar mis habilidades matemáticas, sino que también me enseñará a programar y a usar diferentes herramientas por medio de las matemáticas
E6	Podría hacer que las matemáticas sean más interesantes y accesibles para mí. Podría ayudarme a ver cómo se pueden aplicar los conceptos matemáticos en situaciones reales y prácticas
E7	El tema de la programación me interesa y tengo expectativas de comenzar a conocer más sobre el tema

Las respuestas permiten ver las distintas expectativas que tenían los estudiantes al comienzo de la investigación, es posible destacar ideas tales como la utilidad práctica y aplicativa de la programación para la resolución de problemas matemáticos, la aplicación de conceptos matemáticos en contextos reales gracias a la programación, el interés y disposición en aprender programación como una habilidad transversal y complementaria para la vida y el futuro, así como la percepción de la programación que vuelve a las matemáticas más accesibles y relevantes para todos. En general, las percepciones de los estudiantes sugieren una idea general sobre cómo pueden fortalecer sus habilidades matemáticas mediante la programación puesto que consideran que les proporcionará habilidad de aplicación y una perspectiva distinta para abordar situaciones matemáticas.

Todas las respuestas presentes en la encuesta introductoria fungen como elementos sustanciales para el diseño de las actividades que fueron implementadas de forma posterior, pues las consideraciones de los estudiantes permitieron entrever algunas ideas generales sobre lo que consideraban que sería la programación y su papel en el aula de

matemáticas. Se destaca la presencia de la resolución de problemas como elemento central en las respuestas, en la medida en que la mayoría de los estudiantes consideraban desde el principio que la programación tenía un gran impacto en dicha habilidad, valiéndose de la aplicación de las matemáticas en contextos reales y la resolución de problemas matemáticos.

4.3 Análisis actividad 0

4.3.1 Descripción de la actividad

La primera actividad es llamada en esta investigación como “Actividad 0 o A0” debido a que se presentó como una orientación inicial a Python, diseñada para enseñar a los estudiantes a programar desde cero. En esta se abordan preguntas como: ¿Qué es la programación y para qué sirve? Además, se introdujeron los cuadernos de Google Colab y se proporcionó un espacio de exploración donde los estudiantes pudieron observar e identificar las celdas de código y los bloques de texto. Esta actividad se dividió en dos momentos, inicialmente, el objetivo era impartir conocimientos básicos de Python, comenzando desde la definición de una variable, pasando por las operaciones aritméticas y los operadores de comparación, hasta llegar a los condicionales y operadores lógicos. Este primer momento de la clase se centró en proporcionar una base sólida a los estudiantes en los conceptos de programación que se querían abordar para la investigación, asegurando que los estudiantes comprendieran conceptos básicos antes de avanzar a nociones más complejas.

Para el segundo momento de la actividad, se ofreció un espacio de exploración, donde los estudiantes pudieron poner en práctica los conocimientos adquiridos, permitiendo que experimentaran con el código en un entorno práctico, fomentando la experimentación y la aplicación de conceptos teóricos en situaciones reales. Logrando así, un aprendizaje desde la comprensión teórica hasta la aplicación práctica del conocimiento.

4.3.2 Desarrollo de la actividad

Como recurso para la captura de datos de la actividad se contó como escenario la sala de sistemas de primaria del colegio. Con relación a los informantes, se contó con la participación de 7 estudiantes de niveles de escolaridad entre grado noveno y undécimo. Para el desarrollo de la actividad se entregó la hoja guía de trabajo en Google Colab (ver anexo 1) de forma virtual y cada estudiante contó a su disposición con un computador que tenía conexión a internet y le permitió ingresar a la plataforma Colaboratory de la suite de Google.

4.3.3 Objetivo

Introducir los conceptos básicos de programación en Python y proporcionar a los estudiantes un espacio práctico en Google Colab para aplicar y consolidar estos conocimientos.

4.3.4 Análisis local A0

La naturaleza de la actividad inicial es de carácter exploratorio y de imitación, es decir que, los estudiantes exploraron la plataforma sobre la cual trabajaron posteriormente en las actividades 1 y 2 al mismo tiempo que seguían las instrucciones y ejemplos de los investigadores encargados de enseñar los conceptos básicos de la programación en Python. Esta actividad sirvió como elemento para relacionar algunas capacidades matemáticas fundamentales con procesos matemáticos asociados a la labor realizada por los estudiantes.

En particular, la representación aparece en primer lugar como una capacidad matemática fundamental en los estudiantes puesto que el entorno de programación les permitió llevar a cabo distintas formas de representar objetos matemáticos tales como las variables, pues en el lenguaje matemático se encuentran acostumbrados a el uso de variables como valores numéricos desconocidos, mientras que en el campo de la programación el concepto de variable es mucho más amplio y abarca desde valores numéricos hasta valores textuales y lógicos. Presentamos a continuación representaciones hechas por E1, E3 y E4 bajo la consigna de definir variables de distintos tipos en Python:

Figura 3. Trabajo realizado en A0 por E1, E3, E4

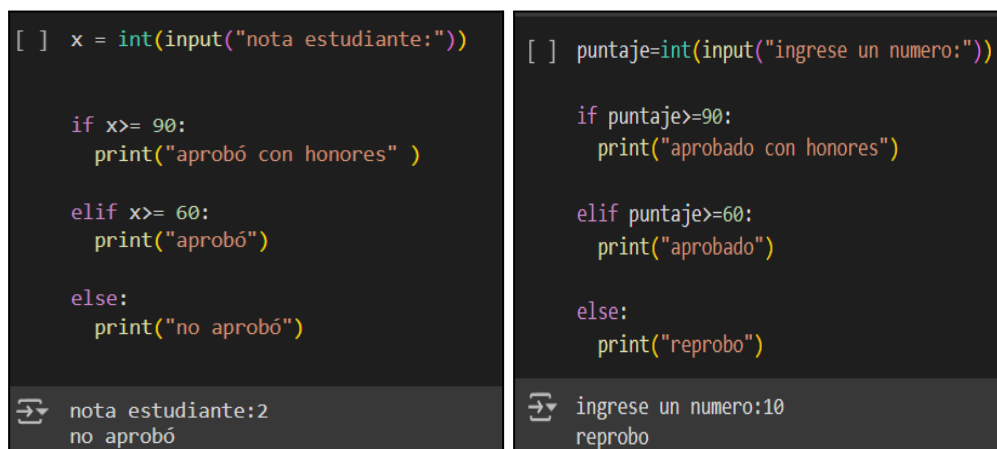
<pre>[] edad = 15 print(edad)</pre> <pre>⇒ 15</pre>	<pre>[] Nombre = "Valeria" Nombre</pre> <pre>⇒ 'Valeria'</pre>	<pre>[] edad = 15 print(edad)</pre> <pre>⇒ 15</pre>
<pre>[] nombre= "lisa" nombre</pre> <pre>⇒ 'lisa'</pre>	<pre>[] Altura= 1.63 Altura</pre> <pre>⇒ 1.63</pre>	<pre>[] nombre = "Isabella" nombre</pre> <pre>⇒ 'Isabella'</pre>
<pre>[] altura = 1.55 altura</pre> <pre>⇒ 1.55</pre>	<pre>[] input("Ingresa tu edad")</pre> <pre>⇒ Ingresa tu edad 16 ' 16'</pre>	<pre>▶ altura = 1.54 altura</pre> <pre>⇒ 1.54</pre>

Aquí podemos observar cómo los estudiantes experimentan desde el principio representaciones distintas para el concepto de variable, salen de lo rutinario y exploran una noción mucho más completa de lo que una variable puede ser en el campo de la computación. Es clave observar que la actividad guía estaba intencionada con el fin de proveer a los estudiantes con percepciones que complementen su formación matemática a través de la programación, es decir que, considerando a Mariotti (2010), la actividad se ha diseñado bajo la consigna de hallar una relación entre el pensamiento matemático y el computacional, pues lo que se pretende es que los estudiantes conozcan el lenguaje de programación y eventualmente adopten sus dinámicas y particularidades con el fin de integrarlas en la resolución de problemas lógicos.

Es de nuestro interés presentar respuestas de la actividad guía debido a que nos ofrece un panorama inicial sobre el estado de los estudiantes y la manera en que se acoplan al entorno y lenguaje de programación Python, esto es importante porque permite establecer un inicio para posteriormente analizar el progreso y la manera en que el

pensamiento computacional se integra con el pensamiento matemático. Posterior a ello se realizaron algunas actividades en torno al uso de operadores de decisión como *if*, *elif* y *else*, donde es posible ajustar el comportamiento de un algoritmo según algunas condiciones dadas, considerando cambios en los parámetros que intervengan en el problema a resolver, lo que ayuda a que los estudiantes identifiquen y optimicen sus procesos de resolución, en relación con el pensamiento matemático les permite trabajar habilidades como identificar, analizar e implementar soluciones diversas con el fin de encontrar la combinación de pasos óptima (Acevedo, 2018). Se adjuntan los algoritmos propuestos por E1 y E4 para solucionar la situación problema “Haga un programa que le pida al usuario la nota de un estudiante, si la nota es 90 o superior, imprima un mensaje donde se indique que aprobó con honores, si la nota es menor que 90 pero mayor o igual que 60, imprima un mensaje que indique que aprobó, de lo contrario imprima un mensaje que indique que no aprobó”:

Figura 4. Trabajo realizado en A0 por E1, E4



```
[ ] x = int(input("nota estudiante:"))

if x>= 90:
    print("aprobó con honores" )

elif x>= 60:
    print("aprobó")

else:
    print("no aprobó")

[ ] x = 2
nota estudiante:2
no aprobó
```

```
[ ] puntaje=int(input("ingrese un numero:"))

if puntaje>=90:
    print("aprobado con honores")

elif puntaje>=60:
    print("aprobado")

else:
    print("reprobo")

[ ] puntaje=10
ingrese un numero:10
reprobo
```

Aunque ambos estudiantes completaron de manera satisfactoria la tarea propuesta, las diferencias entre los algoritmos subyacen en la manera de definir las variables, mientras que E1 representa el puntaje a ingresar como “x”, E4 lo hace como “puntaje”, si bien estas diferencias parecen sencillas a primera vista, con el transcurso de las actividades permiten entrever las formas de apropiación del lenguaje Python para los estudiantes, posteriormente veremos cómo fue el progreso de los estudiantes durante las actividades propuestas y cómo cada uno se fue apropiando poco a poco de la sintaxis y el orden necesario para construir los algoritmos, eje central de la investigación pues permitió seguir el ritmo a la conexión entre el pensamiento matemático y computacional y cómo se iban complementando entre sí.

4.4 Análisis actividad 1

4.4.1 Descripción de la actividad

La actividad #1 (A1) consistió en un taller diseñado con el objetivo de afianzar los conocimientos adquiridos por los estudiantes en el uso de Python. La actividad se estructuró en tres secciones de acuerdo con los temas abordados en la primera clase: operaciones aritméticas, operadores de decisión y operadores de iteración. Cada una con un conjunto específico de preguntas y ejercicios para evaluar distintas habilidades.

En la primera sección se presentaron cuatro preguntas enfocadas en las operaciones básicas de Python: *suma*, *resta*, *multiplicación* y *división entera*. Estas preguntas fueron

diseñadas para asegurar que los estudiantes comprendieran y pudieran aplicar correctamente estas operaciones fundamentales.

Figura 5. A1 - Sección 1

Operaciones básicas con Python

1. Calcula la división entera de 45 entre 6
2. Calcula el residuo de la división entera de 45 entre 6
3. Realiza la siguiente operación en Python, donde $\div$ indica la división entera:
$$10 + 20 \div 7 - 2$$
4. Realiza la siguiente operación en Python:
$$9 - \frac{25 + 5 - 2}{7 * 4}$$
$$\frac{\quad}{2^3}$$

La segunda sección incluyó dos preguntas que requerían el uso de operadores de decisión, como *if*, *elif* y *else*. Estas preguntas tenían como objetivo evaluar la capacidad de los estudiantes para implementar estructuras de control de flujo, una habilidad crucial en la programación.

Figura 6. A1 - Sección 2

Operadores de decisión (if, elif, else)

1. Haz que un usuario introduzca un número real y evalúa si dicho número es positivo, negativo o cero. Devuelve por pantalla el resultado en cada caso.
2. Haz que un usuario introduzca dos números enteros positivos. Comprueba si el primer número introducido por el usuario es mayor, menor o igual que el segundo número introducido por el usuario. Devuelve por pantalla el resultado en cada caso.

PISTA: Asegúrate de hacer uso de la función `int()`.

Finalmente, la tercera sección consistió en cuatro preguntas relacionadas con los operadores de iteración, también conocidos como ciclos *for* y *while*. Estas preguntas fueron diseñadas para evaluar la comprensión y la habilidad de los estudiantes para trabajar con bucles, lo cual es esencial para la realización de tareas repetitivas y la automatización de procesos.

Figura 7. A1 - Sección 3

Operadores de iteración (for, while)

1. Programar un bucle que cuente atrás del 5 al 1 y al final imprimir el mensaje ¡despegue!
2. Mostrar la tabla de multiplicar del número 6.
3. Hallar la suma de los números múltiplos de 5 menores a 50
Ejemplo: $5 + 10 + 15 + \dots + 45$
4. Del ejercicio anterior, solicitar el número mayor hasta donde llegue la suma y el número base para los múltiplos a calcular, es decir que el usuario debe ingresar dos números, uno que determine los múltiplos y otro para el número más grande.

La actividad estaba planificada para ser desarrollada de manera autónoma por los estudiantes, con la mediación y el apoyo de los docentes disponibles en caso de ser necesario. La intención era que los estudiantes pudieran enfrentar y resolver los problemas de manera independiente, recurriendo a la ayuda docente solo cuando se encontraran con dificultades significativas.

4.4.2 Objetivo

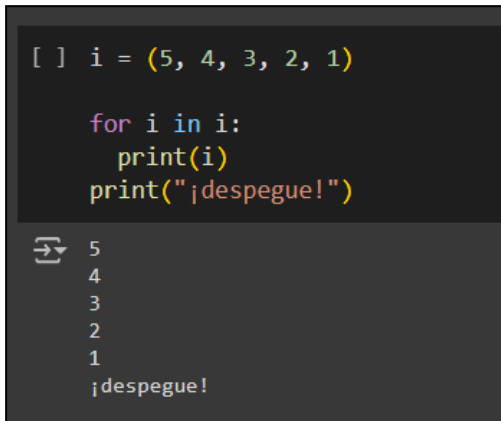
Afianzar y evaluar los conocimientos adquiridos por los estudiantes en el uso de Python, específicamente en las operaciones aritméticas, operadores de decisión y operadores de iteración.

4.4.3 Análisis local A1

Esta actividad A1 funge con el propósito de evaluar los conocimientos adquiridos por los estudiantes con el uso de la programación. A través de esta actividad, se pudo evidenciar cómo los estudiantes lograron responder y abordar las preguntas de manera muy acertada, utilizando diversas heurísticas, estas estrategias y métodos heurísticos empleados serán detallados y analizados en este capítulo.

A continuación, tomaremos en consideración el caso del estudiante E4 que se encuentra desarrollando el primer punto de los operadores de iteración, que veremos en la siguiente figura:

Figura 8. Respuesta E4. A1 - Sección 3. Pregunta 1



```
[ ] i = (5, 4, 3, 2, 1)

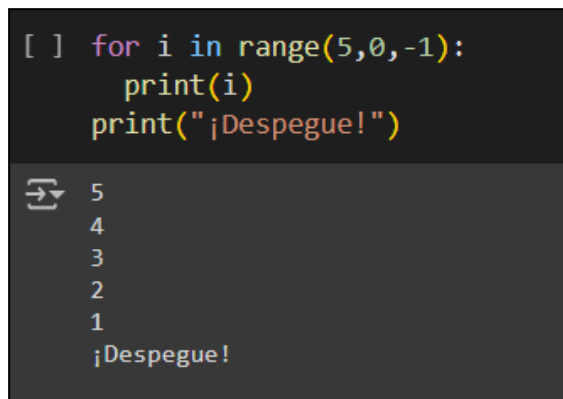
for i in i:
    print(i)
    print("¡despegue!")
```

↔ 5
4
3
2
1
¡despegue!

Como se puede observar, E4 muestra una comprensión básica sobre cómo funcionan las iteraciones con Python, dado que inicialmente crea una tupla con los elementos que desea iterar y posteriormente, utiliza el bucle *for* para iterar sobre estos. En cuanto a su pensamiento algorítmico, se evidencia como los elementos de la tupla se encuentran en orden decreciente lo que permite afirmar que la estudiante comprende que el bucle debe contar hacia atrás, identificando un entendimiento en la progresión decreciente y su implementación. No obstante, podemos observar cómo E4 no utiliza la función *range* para generar la secuencia decreciente la cual es más común a utilizar en estos casos, lo que evidencia que no se encuentra completamente familiarizada con esta.

Por otro lado, podemos observar cómo E2 si hace uso de la función *range* para resolver este punto de la actividad, como vemos a continuación:

Figura 9. Respuesta E2. A1 - Sección 3. Pregunta 1



```
[ ] for i in range(5,0,-1):  
    print(i)  
    print("¡Despegue!")
```

5
4
3
2
1
¡Despegue!

El estudiante E2 genera un código bien estructurado haciendo un uso adecuado del bucle *for*, asimismo, demuestra un buen conocimiento al usar la función *range* con los parámetros adecuados para formar una secuencia numérica decreciente de forma simple y eficiente y no optando por escribir uno a uno los elementos de la tupla.

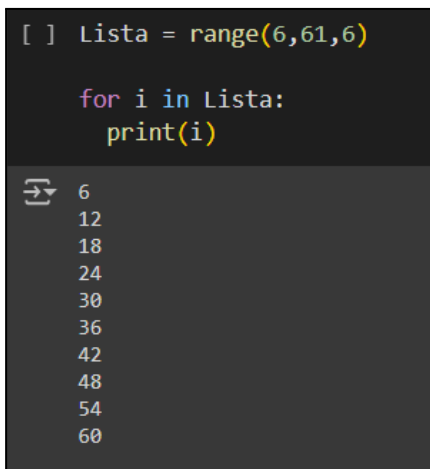
De esta manera podemos ver como las acciones de las estudiantes reflejan lo propuesto por Bartolini Bussi & Mariotti (2002) en la TMS, debido a que reflejan el papel de los artefactos y signos en el proceso de mediación semiótica. En este caso, Python y su sintaxis actuaron como artefactos que mediaron la comprensión y solución del problema. Dado que las estudiantes al interactuar con el artefacto (Python) lograron interpretar y producir signos matemáticos (códigos) que les permitieron generar una secuencia decreciente de números, las estudiantes a través de este proceso están internalizando y representando conceptualmente la idea matemática de una secuencia numérica de orden descendiente

En este orden de ideas, podemos decir que ambas estudiantes logran resolver la actividad al problema de bucle de una manera adecuada, usando diferentes estrategias y

heurísticas. No obstante, las respuestas de los dos estudiantes ilustran diferentes niveles de comprensión y habilidades en el uso de la programación como herramienta para el aprendizaje de conceptos matemáticos, siendo E2 quien tiene un dominio más avanzado del uso de Python, especialmente en la utilización de la función *range* para generar una secuencia decreciente.

Ahora bien, analizaremos el segundo punto del taller de esta parte de iteradores de decisión, el cual pide que los estudiantes logren mostrar la tabla de multiplicar del número 6, a continuación, examinaremos las respuestas dadas por E4 y E1.

Figura 10. Respuesta E4. A1 - Sección 3. Pregunta 2



```
[ ] Lista = range(6,61,6)

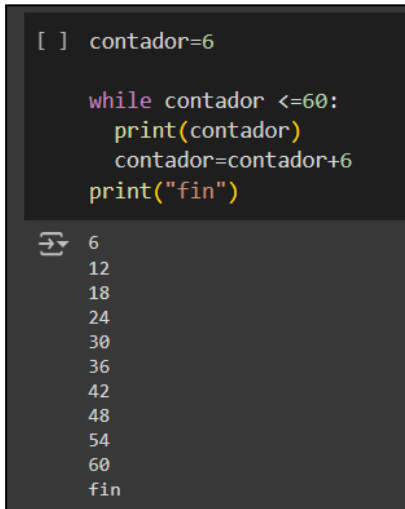
for i in Lista:
    print(i)
```

6
12
18
24
30
36
42
48
54
60

Esta fue la respuesta dada por E4, el estudiante utiliza correctamente la función *range* para generar la tabla del 6. Los parámetros (6, 61, 6) indican que la secuencia comienza en 6, termina en 60 (ya que 61 no está incluido) y avanza en incrementos de 6. Así mismo, utiliza el bucle *for* para iterar a través de los elementos generados por *range* siendo este un método adecuado y eficiente para iterar sobre una secuencia predefinida. Este ejercicio permite evidenciar cómo, con la ayuda y la mediación del docente, el

estudiante logra un dominio más claro sobre la función *range*, superando el obstáculo presente en el ejercicio anterior. Ahora bien, analicemos la respuesta dada por E1:

Figura 11. Respuesta E1. A1 - Sección 3. Pregunta 2



```
[ ] contador=6

while contador <=60:
    print(contador)
    contador=contador+6
print("fin")
```

6
12
18
24
30
36
42
48
54
60
fin

Por otro lado, E1 presenta un método de solución distinto a E4, en donde hace uso del bucle *while* para generar la tabla del 6, esto demuestra que entiende el funcionamiento del bucle, el cual se encarga de iterar a partir de una condición dada. En este caso, la condición *while contador <= 60* asegura que el bucle continuará ejecutándose mientras el contador sea menor o igual a 60, teniendo un buen control del bucle y evitando crear un bucle infinito.

Así pues, ambas estudiantes demuestran una comprensión sólida de bucles y secuencias utilizando el lenguaje de programación Python, aunque ambas utilizan diferentes funciones y enfoques logran llegar a una respuesta acertada, el estudiante E4 evidencia una buena comprensión de las secuencias numéricas y cómo generarlas utilizando *range*. Esto indica que puede aplicar conceptos matemáticos a problemas de

programación de manera efectiva. Por otro lado, el estudiante E2 tiene una comprensión de la lógica de bucles, específicamente el uso de *while* para iterar hasta que se cumpla una condición específica, lo cual refleja un buen entendimiento del control de flujo en programación. Este ejercicio nos permite destacar como la programación permite llegar a una misma respuesta con el uso de diversos comandos y funciones de Python, así mismo, se muestra como un recurso apto y capaz para enseñar y reforzar conceptos matemáticos.

4.5 Análisis actividad 2

4.5.1 Descripción de la actividad

La actividad #2 (A2) consistió en la realización de un taller que presentaba un nivel de complejidad superior al de la actividad anterior. Este taller fue diseñado con el objetivo de identificar y analizar los conocimientos adquiridos por los estudiantes en el área de programación a lo largo del proceso de investigación. A través de una serie de ejercicios y desafíos avanzados, se buscó no solo verificar el dominio de conceptos fundamentales, sino también la capacidad de aplicar dichos conocimientos en situaciones prácticas y en la resolución de problemas. Este taller al igual que el anterior se estructuró en tres secciones, las cuales son: operaciones aritméticas, operadores de decisión y operadores de iteración. Cada una con un conjunto específico de preguntas y ejercicios para evaluar distintas habilidades.

En la primera sección se presentaron dos preguntas relacionadas con las operaciones básicas de Python: suma, resta, multiplicación y división. Adicionalmente,

esta sección del taller introduce a los estudiantes a la creación de una aplicación sencilla pero funcional, la cual consiste en una calculadora básica. Estos ejercicios propuestos tienen como objetivo reforzar en los estudiantes el uso de la entrada y salida de datos en Python, así como el uso de funciones básicas para realizar operaciones matemáticas.

Figura 12. A2 - Sección 1

Operaciones básicas con Python

1. Realiza la siguiente operación en Python:

$$\frac{2 + 2^3 - 2(2 - 2^5) + 2^2 * 4 + 2}{2(2 * 2 - 2^4) + 2^2 * 2 - 2}$$

2. Crea una calculadora básica en Python, la cual le pida al usuario ingresar dos números, luego escoger entre cuatro operaciones posibles (suma, resta, multiplicación y división) y después resolver la operación escogida usando los dos números que ingresó, a la final se debe mostrar el resultado

Pista: la variable numérica debe ser de tipo float()

En cuanto a la segunda sección, esta se compone de dos preguntas en las que los estudiantes deben utilizar estructuras de control de flujo como *if*, *elif* y *else* para tomar decisiones basadas en condiciones específicas. Los ejercicios incluyen verificar si una división es exacta y determinar si un número es múltiplo de otro, en donde los estudiantes deben abarcar conceptos matemáticos como residuo y cociente de un número. El objetivo de esta sección es reforzar en los estudiantes el uso de estas estructuras, además de abordar conceptos matemáticos fundamentales que a menudo se pasan por alto en la educación tradicional.

Figura 13. A2 - Sección 2

Operadores de decisión

1. Haz que un usuario introduzca dos números enteros positivos. Suponiendo que el primer número introducido por el usuario es mayor o igual al segundo número introducido por el usuario, comprueba que la división del primer número entre el segundo número es exacto.
En caso de que la división sea exacta, devuelve el cociente por pantalla e indica que la división en efecto es exacta. En caso de la división no ser exacta, devuelve el cociente y el resto por pantalla e indica que la división entre los dos números no es exacta
2. Haz que un usuario introduzca dos números enteros positivos. Comprueba si el mayor es múltiplo del menor. Devuelve por pantalla el resultado en cada caso.

Finalmente, la tercera sección del taller se constituye por dos preguntas relacionadas con los operadores de iteración, conocidos también como bucles *for* y *while*. Estos ejercicios fueron diseñados con el objetivo de reforzar en los estudiantes el uso de operadores de iteración para repetir acciones basadas en condiciones, reforzando su comprensión de generalización, así como automatización de tareas repetitivas.

Figura 14. A2 - Sección 3

Operadores de iteración

1. Haz que el usuario introduzca números enteros por teclado. Mientras el usuario no introduzca el 0, muestra si el número introducido es par o impar.
2. Haz que el usuario introduzca su edad y el año actual. Imprime todos los años que han pasado desde su año de nacimiento hasta el año actual (ambos incluidos).

Este segundo taller fue diseñado con el propósito de ser realizado de manera autónoma por los estudiantes, sin necesidad de apoyo o mediación directa de los docentes. Esto permitió evaluar de manera efectiva los conocimientos adquiridos por los estudiantes

a lo largo de la investigación. Sin embargo, los docentes dieron seguimiento al progreso y el avance de cada estudiante, estando atentos a cualquier dificultad que pudieran enfrentar, proporcionándoles asistencia cuando era necesario.

4.5.2 Objetivo

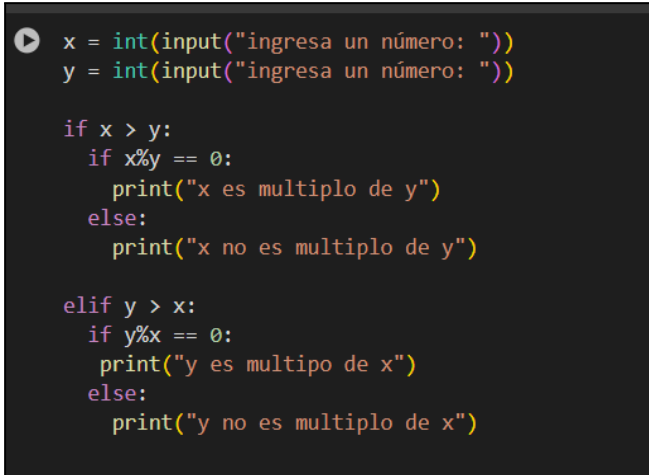
Consolidar los conocimientos adquiridos por los estudiantes mediante la realización de una serie de ejercicios que abordan conceptos fundamentales de la sintaxis, estructuras de control de flujo, operadores matemáticos y de iteración.

4.5.3 Análisis local A2

Esta segunda actividad, se centra en la comprensión inherente a los operadores de iteración, los controles de flujo y las operaciones aritméticas en Python. Este enfoque permitió revelar de manera más detallada los conocimientos adquiridos por los estudiantes en este ámbito. Para este análisis, se evaluaron las respuestas proporcionadas por los estudiantes designados como E1, E4 y E6. El análisis de sus respuestas nos permitió apreciar con mayor claridad cómo cada estudiante aplicó los conceptos aprendidos, así como, identificar sus áreas de fortaleza y aquellas que requerían mayor atención.

A continuación, analizaremos las respuestas dadas por E4 y E1 con relación al siguiente ejercicio propuesto: “Haz que un usuario introduzca dos números enteros positivos. Comprueba si el mayor es múltiplo del menor. Devuelve por pantalla el resultado en cada caso.”

Figura 15. Respuesta E4. A2 - Sección 2. Pregunta 2



```
x = int(input("ingresa un número: "))
y = int(input("ingresa un número: "))

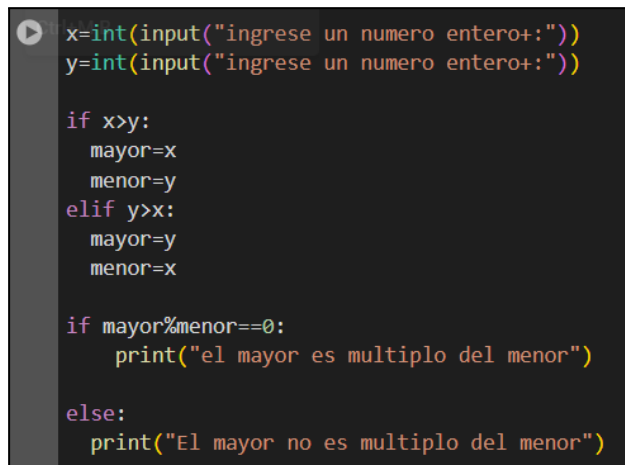
if x > y:
    if x%y == 0:
        print("x es multiplo de y")
    else:
        print("x no es multiplo de y")

elif y > x:
    if y%x == 0:
        print("y es multiplo de x")
    else:
        print("y no es multiplo de x")
```

La respuesta del estudiante E4 nos permite evidenciar su pensamiento matemático, ya que al verificar si un número es múltiplo de otro, utiliza la operación de módulo % para determinar si el residuo de la división entre x y y es cero, lo cual es una aplicación directa de la definición matemática de múltiplos. Incluso, al utilizar las estructuras de control de flujo (*if*, *elif*, *else*) para comparar los dos números ingresados por el usuario y al hacer uso de las condiciones $if\ x \% y == 0$: y $if\ y \% x == 0$: refleja un razonamiento lógico para cubrir todos los casos posibles cuando x es mayor que y , y viceversa. Esta estructura demuestra un pensamiento lógico-matemático al prever y gestionar todas las combinaciones posibles de entrada.

Ahora bien, se puede notar que el estudiante E4 abordó correctamente la pregunta propuesta. Analicemos la respuesta dada por E1 de este mismo punto.

Figura 16. Respuesta E1. A2 - Sección 2. Pregunta 2



```
x=int(input("ingrese un numero entero+:"))
y=int(input("ingrese un numero entero+:"))

if x>y:
    mayor=x
    menor=y
elif y>x:
    mayor=y
    menor=x

if mayor%menor==0:
    print("el mayor es multiplo del menor")
else:
    print("El mayor no es multiplo del menor")
```

Con respecto al estudiante E1, muestra habilidades del pensamiento computacional al diseñar un algoritmo claro y eficiente para resolver el problema planteado, así mismo, al igual que E4 demuestra comprensión del concepto matemático al seguir un enfoque similar a este, utilizando la operación de módulo % para determinar si el residuo de la división es cero, lo cual indica que un número es múltiplo del otro.

Así pues, aunque ambos estudiantes abordaron el problema utilizando diferentes heurísticas y metodologías, sus respuestas demuestran una comprensión sólida de los conceptos matemáticos y la capacidad de aplicarlos utilizando estructuras de control de flujo en Python. Ambos manejan un enfoque claro y estructurado que refleja un pensamiento lógico-matemático y computacional sólido, lo que les permite a ambos abordar el problema de manera efectiva y eficiente.

Se puede observar que las respuestas dadas por los estudiantes cumplen con los criterios del pensamiento lógico-matemático propuestos por Corredor (2011). Dado que

ambos demuestran habilidades de representación al poder traducir el problema matemático en un algoritmo ejecutable. Además, lograron desarrollar la capacidad de abstracción ya que ambos identificaron un patrón general, el cual consiste en que, si el residuo de la división entre el número mayor y el número menor es cero, entonces el número mayor es múltiplo del número menor, generalizando la solución a cualquier par de números enteros positivos.

Así mismo, ambos estudiantes cumplen con el criterio de creatividad, ya que diseñaron algoritmos eficientes y efectivos para abordar el problema de formas distintas. Ahora bien, en cuanto a la demostración matemática, aunque no realizan una demostración matemática formal, sus soluciones incluyen una justificación implícita y es que, si el residuo de la división entre el número mayor y el número menor es cero, entonces el número mayor es múltiplo del número menor. Esta justificación se basa en principios matemáticos fundamentales sobre múltiplos y divisibilidad.

El segundo punto para analizar se centra en la sección 3 con los estudiantes E1 y E6, estos deben dar respuesta a la siguiente pregunta: “Haz que el usuario introduzca números enteros por teclado. Mientras el usuario no introduzca el 0, muestra si el número introducido es par o impar”. Para esto, el estudiante E1 aborda el problema de la siguiente manera:

Figura 17. Respuesta El. A2 - Sección 3. Pregunta 1

```
[ ] x=int(input("ingrese un numero entero:"))  
  
while x!= 0:  
  
    if x%2==0:  
        print("el numero es par")  
  
    else:  
        print("el numero es impar")  
  
    x=int(input("ingrese un numero entero:"))
```

El estudiante da respuesta a la pregunta de una forma muy acertada demostrando una sólida comprensión del concepto de números pares e impares, siendo consciente de que, si a número x lo divide entre 2 y obtiene como residuo al cero, implica que el número x es par. Así mismo, cuando este residuo es diferente de cero es porque el número es impar. Por otro lado, utiliza un bucle *while* para repetir la solicitud de ingreso de números hasta que el usuario ingrese el número cero, lo que refleja un buen razonamiento lógico-matemático por parte del estudiante para manejar la entrada del usuario de manera eficiente.

El estudiante exhibe un sólido pensamiento computacional al emplear un bucle *while* para iterar el proceso de entrada de datos hasta que se cumpla la condición de ingresar el número cero. Esta habilidad refleja una comprensión destacada del flujo de control en la programación. Este enfoque, como señala Acevedo (2018), está estrechamente vinculado al desarrollo del pensamiento algorítmico y el razonamiento numérico. Al solicitar la recopilación de datos a través del teclado en un ciclo repetitivo, el

estudiante no solo demuestra la repetición de un proceso varias veces, sino que también ilustra cómo dicho proceso puede generalizarse y ejecutarse, independientemente de las entradas o salidas específicas. Esto resalta la importancia del razonamiento numérico en el desarrollo del pensamiento algorítmico. Ahora bien, analicemos la respuesta dada por el estudiante E6 a esta misma pregunta:

Figura 18. Respuesta E6. A2 - Sección 3. Pregunta 1

```
[ ] x = int(input("Ingresa un número "))
    if x%2==0:
        print(x,"es par")
    else:
        print(x,"es impar")
```

Al igual que la estudiante E1, la estudiante E6 demuestra un claro entendimiento del concepto matemático de números pares e impares y la capacidad de aplicarlos mediante una estructura de control de flujo en Python. Sin embargo, aunque el código de E6 es más conciso, no resuelve completamente el problema. E6 utiliza una única instancia de entrada para verificar la paridad o imparidad de un solo número, omitiendo la condición inicial de emplear un bucle *while* para repetir la solicitud de ingreso de números hasta que se ingrese un cero, condición que E1 si cumple.

En conclusión, ambas respuestas dadas por los estudiantes demuestran una comprensión sólida de los conceptos matemáticos, lógico-matemáticos y computacionales necesarios para abordar el problema de determinar si un número es par o impar. No obstante, la respuesta de la estudiante E1 muestra un enfoque más estructurado y repetitivo

a hacer uso del bucle *while* para proporcionar múltiples entradas de usuario, mientras que la respuesta de la estudiante E6 no cumple con la condición propuesta la pregunta, dado que la aborda de una manera más directa al procesar una sola entrada de usuario.

4.6 Encuesta final

La encuesta final se presenta como una oportunidad para reflexionar sobre el impacto de la investigación, pues recopila respuestas de los estudiantes una vez culminado su proceso a través de las distintas actividades analizadas. El propósito de esta encuesta consiste en obtener información sobre las percepciones de los estudiantes en cuanto a su progreso y crecimiento en el campo de la programación y su uso en el aula de matemáticas. Además, funge como insumo para los investigadores en aras de mejorar los diseños e implementaciones futuras en este campo. Esta encuesta permite evaluar de forma íntegra el impacto de las sesiones de acuerdo con los objetivos planteados inicialmente y proporcionar consideraciones sobre el desarrollo de las habilidades de razonamiento lógico matemático y del pensamiento computacional en los estudiantes.

Tabla 9. *Respuestas de la pregunta 1 de la encuesta final*

¿Cómo describirías tu experiencia general aprendiendo programación en Python en relación con las matemáticas durante este breve curso?	
Estudiante	Respuesta
E1	Me pareció muy interesante poder entender la programación básica
E2	Me pareció una experiencia muy positiva e interesante. Aprender a programar me parece una herramienta muy importante
E3	Muy buena, aprendí mucho sobre cómo utilizar ese tipo de programación

E4	Me pareció una experiencia interesante que me permitió aprender sobre una herramienta para darle solución a distintos problemas matemáticos
E5	Me sentí muy satisfecha
E6	Describiría esta experiencia como interesante, fue divertido aprender todo alrededor de Python, los códigos y actividades
E7	Muy bien, una gran experiencia

Las respuestas de los estudiantes reflejan una experiencia general positiva y enriquecedora en su proceso de aprendizaje de la programación en Python con un enfoque hacia las matemáticas, si bien no hay grandes descripciones de su vivencia, podemos destacar algunos elementos importantes aquí presentes. E2 hace una precisión relevante sobre cómo aprender a programar parece ser una “herramienta muy importante”, esto es clave en la investigación puesto que desde las consideraciones pedagógicas planteadas deseábamos que el lenguaje de programación Python se convirtiera en un instrumento para el estudiante, si bien en una primera instancia los “códigos” escritos en Google Colab parecían algo aislado, durante el transcurso de las sesiones los estudiantes se desarrollaron mejor dentro de la plataforma y su capacidad de emplear un tipo de operador u otro aumentó sustancialmente. Bajo las palabras de Moreno (2002), apreciamos un esbozo de una reorganización conceptual en la medida en los estudiantes pasaron de la escritura de códigos por repetición y de complejidad sencilla hacia el uso de operadores de programación de forma más orgánica dentro del entorno, logrando dar solución a problemas estructuralmente más complejos. E4 comparte la idea de “herramienta” con E2,

además de advertir sobre la utilidad en la resolución de problemas, habilidad que estuvo presente durante toda la investigación.

Tabla 10. *Respuestas de la pregunta 2 de la encuesta final*

¿Qué conceptos matemáticos has logrado comprender mejor gracias a la incorporación de la programación en Python? Proporciona algunos ejemplos específicos.	
Estudiante	Respuesta
E1	Entendí los parámetros de las operaciones básicas, por ejemplo, cómo identificar realmente qué es un número par, que es un múltiplo, etc
E2	Las variables
E3	Cómo generar una respuesta en la aplicación de Google Colab
E4	Comprendí mucho más el concepto del residuo en una división, ya que antes no le daba tanta importancia, pero ahora sé que este puede ayudar cómo un factor que determina cuando una división es exacta, algo que antes no tenía en cuenta
E5	He logrado aprender a realizar operaciones básicas programando
E6	Variables
E7	Variables

Se destacan conceptos como variables, operaciones básicas y residuos en divisiones. La interacción con la programación les ha permitido entender de forma explícita conceptos matemáticos que parecen ser inconexos de sus clases de matemáticas actuales o de uso poco frecuente. Por ejemplo, tanto E1 como E4 mencionan el concepto de división y las formas de identificar si un número es par o impar, el concepto de múltiplo y cómo el residuo ayuda a solucionar problemas que involucren dichos objetos

matemáticos. Tomando como objeto matemático el residuo, es posible afirmar que es un concepto fundamental dejado de lado en las clases de matemáticas debido a su falta de aplicaciones en contextos diversos, además, cuando se enseña la división, el énfasis recae sobre el cociente como resultado, y el residuo pasa a segundo plano, sin embargo, en el campo de la programación donde la máquina entiende únicamente algoritmos, es completamente central entender cómo el residuo puede ayudarnos a establecer relaciones entre números y solucionar problemas que involucren este tipo de conceptos.

De otra parte, se evidencia una idea común sobre las variables, pues múltiples estudiantes consideran que es un concepto que han logrado afianzar gracias al uso de Python, esto es debido a que en programación el uso de variables es rutinario y elemental para la construcción de algoritmos, pues las variables representan espacios en el sistema, comprender y manipular variables permite clasificar y depurar información de una manera más ordenada y sencilla (Acevedo, 2018), promoviendo el desarrollo del pensamiento lógico de los estudiantes en la medida en que deben organizar y estructurar los datos para utilizarlos en algoritmos que den solución a un problema.

Tabla 11. *Respuestas de la pregunta 3 de la encuesta final*

¿Cómo ha influido el aprendizaje de Python en tu forma de abordar y resolver problemas de lógica y programación? Proporciona algún ejemplo si es posible.	
Estudiante	Respuesta
E1	Python hizo que en la vida cotidiana tuviera herramientas para solucionar problemas. Aparte puedo utilizar esos conceptos básicos en materias como sistemas, como cuando trabajamos con Excel
E2	Yo creo que me ha servido para tener un orden conciso al intentar

	solucionar problemas
E3	Me ayudó a entender un poco más cómo funciona la programación
E4	El aprendizaje de Python me permitió comprender que este tipo de problemas se pueden resolver a través de algoritmos que simplifican los pasos a realizar para llegar a una solución ante diferentes problemas
E5	Es mucho más fácil ya que las herramientas que Python proporciona son bastante útiles
E6	Python me parece otra forma de ver temas o abordar matemáticas, en base de leer la actividad, ayuda a comprender y llevarlo a un lenguaje matemático
E7	Que me ayudó mucho con mi organización

Las respuestas de los estudiantes muestran cómo el aprendizaje de Python ha influenciado su habilidad para abordar problemas mediante la descomposición en pasos más pequeños y la aplicación de algoritmos. Los estudiantes expresan haber adquirido habilidades para desglosar problemas complejos en tareas más manejables, lo que les permite abordarlos de manera más estructurada y eficiente. Además, mencionan que la forma en que Python les permite ordenar y solucionar los problemas les ayuda a comprender mejor las maneras de abordar las situaciones, y esto es clave ya que los algoritmos computacionales deben tener una estructura definida para funcionar. A través de este proceso, los estudiantes no solo mejoran su comprensión de los algoritmos, sino que también desarrollan habilidades para enfrentar problemas de manera más sistemática y metódica. Además, no limitan el uso de la programación al ámbito computacional, sino que también reconocen su utilidad en diversos contextos, desde la vida cotidiana mediante el orden y la organización y hasta la comprensión del lenguaje matemático.

Tabla 12. Respuestas de la pregunta 4 de la encuesta final

¿Qué desafíos encontraste al aprender programación en Python y cómo los superaste? ¿Hay algo que el breve curso podría haber hecho de manera diferente para ayudarte más?	
Estudiante	Respuesta
E1	La manera en la que explicaron fue muy clara y en si el programa es muy fácil de entender. Lo único que me parece complejo es el pequeño margen de error que tiene Python, aunque con experiencia se soluciona
E2	De pronto se me complicó el acordarme del orden que debe tener el código, pero poco a poco me fui acostumbrando
E3	Tuve dificultades con el internet, pero solucioné las dudas que tenía
E4	Un desafío que tuve fue que lo que escribía no podía tener ningún error y por eso algunos ejercicios no me quedaban bien en el primer intento. Sin embargo, en el transcurso de las clases intenté ser más precavida con cada paso y tener en cuenta los detalles en cada ejercicio
E5	A veces me confundía al no saber cómo decirle al computador lo que quería que hiciera, pero gracias a la explicación de los profes fue mucho más fácil
E6	Podría ser el orden o la forma correcta de escribir los códigos, lo supere con ayuda de los chicos
E7	Aprenderse cuales signos usar, o al momento de interpretar del texto y aplicarlo en el programa, con apoyo de ellos, y mis compañeras también pude superarlo

Es común para los estudiantes encontrar algunos desafíos en torno a la sintaxis correcta, el orden del código y, en general, a los errores que aparecen al ejecutar un algoritmo que no funciona de forma esperada. Además, el correcto orden y estructuración de los algoritmos en programación vuelve al proceso más complejo. Tal como lo mencionan E2, E4 y E6, podemos observar un común acuerdo en cuanto al orden de los

“códigos”, pues se percataron de la importancia que tiene este, ya que si un algoritmo no está bien estructurado, no se podrá dar solución al problema planteado, y esto no es un elemento aislado, pues en programación, a diferencia del mundo real, se está “hablando” con una máquina, la cual responde a instrucciones específicas, así que si un algoritmo no está bien escrito por el mínimo error, el programa no se ejecutará correctamente. Así como lo hemos narrado durante los análisis previos, el progreso de los estudiantes refleja sus esfuerzos por aprender, por adaptarse a un entorno nuevo y desafiante pero lleno de oportunidades por explorar.

Tabla 13. Respuestas de la pregunta 5 de la encuesta final

¿Te sientes más preparado para abordar problemas de razonamiento lógico matemático y/o para continuar aprendiendo programación después de este curso? Explica brevemente por qué sí o por qué no.	
Estudiante	Respuesta
E1	Python me pareció como la vida real, hay un problema, y se tiene que seguir unos lineamientos para resolver ese problema. Lo cual aprendí de programación, ya que tuve que intentar varias veces hasta que me saliera el código, con paciencia; como en matemáticas
E2	Yo digo que, me siento más organizada para afrontar este tipo de problemas y quede con mucha motivación de aprender más programación
E3	Si, gracias a el curso que tomé entiendo más este tipo de programación
E4	Sí, ya que antes tenía una idea de lo que era programación, pero no recordaba mucho. Ahora siento que tengo bases para continuar aprendiendo sobre programación y para resolver así sea problemas sencillos con esta herramienta
E5	Si, ya que creo que las bases ya las tengo y ya el resto es saber cómo formular las cosas

E6	Me siento preparada
E7	Si, con la ayuda de este curso me sentí con más conocimiento sobre este tema, me gustó y me gustará ver más sobre esto

Es pertinente destacar la respuesta de E1, pues compara la experiencia de aprender Python con la resolución de problemas en la vida real, conectando la necesidad de seguir lineamientos específicos y la constancia. Esto nos permite ver que el estudiante ve a la programación como un proceso iterativo, similar al abordaje de problemas matemáticos, donde las requieren atención a los detalles y correcto uso del conocimiento. Además, los estudiantes expresan haber adquirido conocimientos sólidos que les permitirán seguir su camino de aprendizaje en el campo de la programación, respuestas que reflejan una experiencia enriquecedora y estimulante proporcionada por la investigación, reforzada por la motivación de seguir explorando y ampliando su conocimiento en este ámbito.

En síntesis, a través de las respuestas analizadas tanto en la encuesta previa como en la final se destaca el impacto positivo del curso de programación en Python en los estudiantes. Desde el fortalecimiento de las habilidades de resolución de problemas y razonamiento lógico hasta el aumento de la confianza y la motivación para seguir aprendiendo, los estudiantes expresan opiniones favorables sobre su experiencia de aprendizaje. Es posible reconocer un progreso en la comprensión de la programación y el razonamiento lógico matemático a través de sus respuestas, pues en la encuesta final hay conceptos mucho más concretos presentes en las mismas, así como una mayor preparación para abordar desafíos futuros en este campo. Destacamos la relevancia de un proceso de enseñanza efectivo, con un ambiente de colaboración entre compañeros y la práctica activa

en el proceso de aprendizaje de la programación, pues gracias a las condiciones favorables y objetivos de la investigación los estudiantes se encuentran motivados para continuar explorando y desarrollando habilidades en este campo de estudio.

4.7 Análisis retrospectivo

En este apartado se presentarán las conexiones entre los análisis locales previamente mencionados, las encuestas realizadas y la manera en que estas conexiones convergen hacia el marco teórico empleado en la investigación. Las descripciones realizadas en los análisis junto con los propios desarrollos de las actividades permitirán vincular elementos teóricos y conceptuales con la investigación. Tal como lo menciona Camargo (2019), el enfoque fenomenológico de esta investigación nos permite dar cuenta de los resultados obtenidos en aras de dilucidar oportunidades hacia el futuro educativo, pues desde el punto de vista pedagógico, la educación matemática ha sufrido cambios inherentes a la sociedad actual, y esto se ha visto reflejado en los procesos educativos.

4.7.1 Descripciones sobre los análisis realizados

La encuesta inicial y la actividad 0 se establecieron como una introducción hacia la programación en Python y sirvieron de elemento para construir las dos siguientes actividades, es pertinente aclarar que las intenciones educativas de la actividad 0 estuvieron enmarcadas en el contexto pedagógico de la TMS (Mariotti y Bartolini Bussi, 2008), pues desde el inicio las actividades fueron concebidas como un recorrido

progresivo para los estudiantes, los cuales debían adoptar las dinámicas del lenguaje de programación con el fin de apropiarse de él.

En este sentido, la actividad 1 surge como cierre a la fase de inicial de acople y establece un inicio hacia la reorganización conceptual, pues como lo sugiere Armella (2002), el proceso de amplificación está presente cuando los estudiantes programan algoritmos para dar solución a un problema planteado, las situaciones dan pie hacia la integración del pensamiento computacional en el currículo de matemáticas y promueven el razonamiento lógico de los estudiantes, ya que estos empiezan a “reorganizar” los usos que le dan al lenguaje Python, desde interacciones inicialmente leves hasta usos más fluidos y con cada vez más atención a la lógica y no a la sintaxis, es decir que, en la medida en que avanzaban por las actividades los estudiantes atendían mucho más al razonamiento lógico detrás de los problemas y se iban apropiando del lenguaje de programación.

Así pues, la conclusión del proceso de reorganización conceptual, así como la transformación de una “herramienta” hacia un instrumento se da en la actividad 2, en la cual los estudiantes poseen una mejor comprensión de la sintaxis del lenguaje Python y se centran netamente en el planteamiento lógico y estructurado de los algoritmos que les permitan dar solución a los problemas.

4.7.2 El rol de la TMS y el vínculo entre el pensamiento matemático y computacional

Las relaciones entre el pensamiento matemático y computacional se pueden apreciar desde las propias concepciones teóricas, pues elementos como la abstracción y la creatividad son comunes en ambos ámbitos, en nuestro caso, la correspondencia entre habilidades se hizo presente durante el desarrollo de las actividades, ya que los estudiantes constantemente se valían del uso de operadores del lenguaje Python para dar solución a problemas mediante la lógica y el razonamiento matemático.

El pensamiento lógico matemático es transversal a los demás pensamientos (MEN, 2006), por ende, es necesario que haya un elemento conector entre estos, y es aquí donde juega un papel central la TMS, debido a que esta funge como eje articulador de los dos pensamientos (matemático y computacional), pues permite trazar el progreso de conversión del lenguaje Python como un artefacto hasta ser un instrumento para los estudiantes, y esto lo consigue gracias a las relaciones entre los procesos cognitivos de ambos pensamientos. Así como Corredor (2011) ha propuesto cuatro habilidades centrales en el pensamiento lógico matemático: *representación, abstracción, creatividad, demostración matemática*. De otra parte, tenemos que Wing (2002) también propone cuatro habilidades presentes en el pensamiento computacional: *abstracción, algoritmia, descomposición, reconocimiento de patrones*. Si ahondamos un poco en estos fundamentos propuestos por ambos, es fácil reconocer algunos elementos comunes tales como la abstracción y la demostración matemática como una forma de algoritmia, en este sentido,

los elementos comunes se establecieron como oportunidades para el diseño que nos permitieran guiar a los estudiantes durante el proceso de instrumentalización y reorganización conceptual.

En consecuencia, la TMS consigue movilizar aspectos de un pensamiento hacia otro a través de la mediación del docente, la cual juega un papel indispensable en la investigación, ya que sin un correcto diseño de actividades no habría sido posible tal articulación. Asimismo, González y Gómez (2018) proponen una investigación de índole interpretativa, que es lo que hemos vislumbrado a través de los análisis, empleando respuestas de los estudiantes y conectando sus razonamientos con la teoría que soporta la investigación.

4.7.3 La resolución de problemas y el pensamiento lógico matemático a través de Python

La programación computacional está amarrada a la idea de pensamiento lógico de forma inherente, esto es debido a que la construcción de algoritmos computacionales exige una estructuración correcta y el uso de habilidades tales como la descomposición del problema en partes más pequeñas, la identificación de variables que intervienen en el problema, el análisis de las variables, el reconocimiento de patrones y abstracciones, así como la optimización de la cantidad de líneas que tendrá el algoritmo para dar solución al problema.

La incorporación de Python como herramienta dentro del currículo y su posterior transformación hacia instrumento debe tener un propósito claro, debido a que el rol como recurso pedagógico atiende al mismo y subyace al marco teórico planteado. El enfoque adoptado en la investigación nos ha permitido indagar en las respuestas de los estudiantes de forma cualitativa y considerar las construcciones que han hecho como un reflejo de su pensamiento matemático y la forma en que este se ha entrelazado con el pensamiento computacional. Es importante garantizar un diseño coherente y caracterizado por intereses concisos, en nuestro caso, la progresión y dificultad de los temas se vieron reflejados en las respuestas de los estudiantes, debido a que se evidencia un avance en la medida en que se apropiaban del recurso.

Durante las sesiones los estudiantes se enfrentaron a desafíos de programación en los cuales deben descomponer el problema en pasos más pequeños, identificar patrones y relaciones, y aplicar algoritmos y lógica matemática para llegar a una solución. En conjunto, los objetos matemáticos como residuo, paridad de un número y relaciones de orden propiciaron un entorno adecuado para potenciar el pensamiento matemático de los estudiantes, en la medida en que este proceso refleja directamente el pensamiento lógico y computacional, ya que implica la aplicación de principios fundamentales como la abstracción, la deducción y la inferencia (Bocconi et al., 2018).

5. Conclusiones y consideraciones finales

En este apartado concluye la investigación, destacando hallazgos importantes con relación a los objetivos de esta, presentando resultados e implicaciones pedagógicas subyacentes a los análisis realizados. Además, bajo la consideración de un enfoque fenomenológico según Camargo (2019), se comparten recomendaciones y reflexiones pedagógicas para futuras investigaciones que permitan trabajar sobre el campo de la programación y su uso en el aula de matemáticas.

5.1 Objetivo específico 1

- *Identificar las habilidades del pensamiento lógico matemático que se desarrollan a través de la programación computacional.*

El desarrollo de las habilidades del pensamiento lógico matemático a través de la programación está presente especialmente en la actividad 1 y 2. Por ejemplo, al analizar a E4 (figura 15) durante la actividad 2 es posible ver cómo el pensamiento matemático del estudiante se ha visto beneficiado por los operadores de decisión en Python, puesto que al entender la sintaxis el estudiante efectúa algoritmia, encontrando una forma de resolver el problema planteado a través de la secuencia correcta de pasos (Wing, 2006). Además, el uso de conceptos matemáticos como división entera y residuo vislumbran el razonamiento numérico del estudiante y cómo éste se potencia a lo largo de las actividades, pues es E4 quien afirma en la respuesta 2 de la encuesta final (Tabla 10) que comprendió mejor el concepto de residuo, pues no suele usarlo de forma regular en sus clases de matemáticas.

Durante el análisis local A2, se hace una comparación entre las respuestas de la Sección 2 - Pregunta 2 de E1 y E4 (Figura 16), situación que nos permite ver cómo efectúan un proceso de abstracción y descomposición del problema a través del uso de algoritmos computacionales, en este sentido, hay una conexión importante entre la forma en que los estudiantes abordan los problemas y cómo los caminos que toman muestran sus comprensiones sobre el lenguaje Python.

A su vez, en el análisis local A1 es posible ver cómo la creatividad es desarrollada gracias a las distintas representaciones que hacen los estudiantes, ya que una consigna clara desde el inicio de las sesiones era la posibilidad de llegar a una misma respuesta desde múltiples ángulos, y tal como lo menciona el MEN (2006), esta aproximación incide en todos los demás tipos de pensamiento matemático, surgiendo una transversalidad no sólo para el pensamiento lógico matemático, sino también para el uso de la programación como recurso para el desarrollo de todos los pensamientos.

5.2 Objetivo específico 2

- *Analizar las percepciones de los estudiantes al aprender programación en el currículo de matemáticas en la Educación Secundaria en Colombia.*

La encuesta inicial y final permitieron caracterizar de manera amplia este objetivo, pues la recopilación de respuestas reveló un interés de parte de los estudiantes por aprender programación, teniendo motivaciones y expectativas sobre el uso de la misma en distintos campos de la vida. En este sentido, compartimos algunas percepciones generales

sobre la programación y Python extraídas de los análisis realizados a las respuestas de las encuestas:

- La programación es una herramienta esencial en el mundo actual, de igual forma la comparaban como una serie de instrucciones que recibe una máquina para dar solución a un problema mediante algoritmos.
- Es útil para construir y aplicar conceptos matemáticos.
- Ayuda a la solución de problemas de manera eficiente y les facilita la aplicación de conceptos matemáticos en situaciones de la vida real.
- Simplifica la descomposición de problemas complejos en pasos manejables y más “pequeños”.
- Motiva a aprender matemáticas en la medida en que ofrece una alternativa para aplicar las matemáticas mientras que abordan conceptos abstractos.
- Ayuda a entender la lógica matemática de forma más organizada y permite tomar un enfoque sistemático al momento de resolver problemas.

La integración de la programación en Python en el aprendizaje de matemáticas ha sido percibida de manera muy positiva por los estudiantes, quienes destacaron su utilidad en la comprensión y aplicación de conceptos matemáticos. Los estudiantes identificaron que áreas como la estadística, el álgebra y la geometría se benefician significativamente de la programación. Con ayuda de la investigación, los estudiantes lograron una mejor comprensión de operaciones básicas, variables y conceptos como el residuo en una división, y encontraron que la programación les ayudó a descomponer problemas complejos en pasos más simples. Si bien enfrentaron desafíos sobre el orden y la sintaxis

del lenguaje Python, estos fueron superados con práctica, colaboración con sus pares y mediación por parte de los investigadores. En general, las respuestas de la encuesta permiten entrever la motivación de los estudiantes para seguir aprendiendo programación y aplicar estos conocimientos en la resolución de problemas lógicos y matemáticos, revelando así el valor de la programación como un recurso pedagógico funcional en el aula de matemáticas.

5.3 Objetivo específico 3

- *Estudiar el vínculo entre el pensamiento computacional y el pensamiento matemático y su relación con la Educación Matemática.*

Tal como se presenta en Orjuela & Rodríguez (2022), algunos elementos del pensamiento computacional movilizan al pensamiento matemático, y si bien estas son consideraciones teóricas, las actividades implementadas permitieron ver la relación entre dichos elementos de forma práctica, debido a que la resolución de las mismas tiene las siguientes implicaciones:

- **Descomposición de problemas complejos en partes manejables:** proceso inherente a la descomposición de problemas matemáticos en la medida en que se usa un análisis paso a paso para llegar a la solución.
- **Reconocimiento de patrones y regularidades:** la identificación de patrones es una habilidad compartida puesto que utiliza el desarrollo de algoritmos y el análisis de estos para la consideración de patrones

numéricos, geométricos o algebraicos, lo cual evoca en la formulación de conjeturas y razonamientos lógicos por parte de los estudiantes.

- **Abstracción:** en computación, la creación de modelos y algoritmos ayuda a tener una perspectiva lógica de las situaciones a resolver, así pues, en matemáticas la abstracción es una habilidad central pues permite comprender y manipular los conceptos de las distintas ramas.
- **Algoritmos y procedimientos:** el pensamiento computacional tiene una estrecha relación con el pensamiento variacional en la medida en que los algoritmos computacionales fungen como un paso a paso similar a la resolución de una ecuación en álgebra, donde el entendimiento del concepto de variable es central para lograr una solución.

En la presente investigación la incorporación de la programación en el currículo de matemáticas permitió proveer a la enseñanza de nuevas herramientas (que se encaminaron hacia ser instrumentos) y métodos para explorar conceptos matemáticos. Esto no solo desarrolla habilidades transversales como la lógica y la resolución sistemática de problemas, sino que también incide de forma directa en la motivación y el interés de los estudiantes al mostrar aplicaciones prácticas y reales de los conceptos matemáticos. Además, enseñar programación prepara a los estudiantes mejor para el futuro, dado que es una habilidad cada vez más valorada en el mundo laboral.

5.4 Consideraciones finales

5.4.1 A la investigación

El constante crecimiento del campo de la Educación Matemática supone un ritmo acelerado y cambiante en los intereses de esta. Esta investigación pretende establecer un punto de inicio para futuras investigaciones que deseen encaminarse en este ámbito, así como establecer una manera de introducir a la programación en el currículo de matemáticas colombiano. A grandes rasgos hemos establecido relaciones entre la TMS, el pensamiento computacional y el pensamiento lógico matemático, así como elementos conceptuales puntuales del uso de Python y el progreso de los estudiantes durante las distintas etapas de la investigación.

Se evidencia que en términos generales hay mucho camino por recorrer, ya que los procesos experimentales empleados no hacen otra cosa más que abrir múltiples oportunidades para investigaciones en esta línea, y queda claro que, al menos en un sentido amplio, la programación puede instaurarse como un recurso pedagógico que moviliza al saber matemático dentro del aula de matemáticas, y que apoya a procesos como la resolución de problemas y el razonamiento lógico (Bocconi et al., 2018; Education (CECE), 2017).

5.4.2 A las prácticas pedagógicas

Tras la culminación de las sesiones de implementación y los análisis aquí presentados, salen a la luz algunas reflexiones sobre la importancia de considerar a la programación en el currículo de matemáticas y sobre cómo puede lograrse una correcta implementación en la educación matemática. El uso de proyectos interdisciplinarios, el uso de herramientas y plataformas educativas, y la formación docente. Enfatizar en proyectos pedagógicos que involucren a la programación como recurso dentro del aula de las matemáticas, capacitar a los docentes en el uso de estas herramientas, y evaluar tanto el proceso de programación como el entendimiento de los conceptos matemáticos, son pasos cruciales para transformar la manera en que los estudiantes aprenden y aplican las matemáticas.

Estas consideraciones apuntan hacia el enriquecimiento del currículo, además de fomentar un aprendizaje activo, colaborativo y prepara a los estudiantes con habilidades esenciales para su futuro. Para los docentes que deseen tomar este camino, hay un sinfín de posibilidades por explorar en este vasto campo, pues aún se encuentra en sus inicios y cuenta con muchas oportunidades pedagógicas para beneficiar y diversificar los procesos de enseñanza y aprendizaje de las matemáticas.

Referencias

- Acevedo Mera, N. A. (2018). *Desarrollo del pensamiento computacional mediante Scratch en estudiantes de educación media del municipio de Pamplona*. [Pregrado, Universidad de Pamplona].
<http://repositoriodspace.unipamplona.edu.co/jspui/handle/20.500.12744/5565>
- Bikner-Ahsbabs, A., Knipping, C., & Presmeg, N. (Eds.). (2015). *Approaches to Qualitative Research in Mathematics Education: Examples of Methodology and Methods*. Springer Netherlands. <https://doi.org/10.1007/978-94-017-9181-6>
- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). *THE NORDIC APPROACH TO INTRODUCING COMPUTATIONAL THINKING AND PROGRAMMING IN COMPULSORY EDUCATION*. <https://doi.org/10.17471/54007>
- Bonelo Ayala, Y. (2023). *Operaciones visuales y estructuras de control asociadas a la construcción de conjeturas geométricas mediante el Uso de Ambientes de Geometría Dinámica. Un Estudio de Caso de Educadores Matemáticos en Formación*. [Doctoral]. Universidad del Valle.
- Camargo Uribe, L. (2019). *Estrategias cualitativas de investigación en Educación Matemática*. 1–8.
<https://conferencia.ciaem-redumate.org/index.php/xvciaem/xv/paper/viewFile/1061/568>
- Cobb, P., Jackson, K., & Dunlap, C. (2017). Conducting design studies to investigate and support mathematics students' and teachers' learning. *Compendium for research in mathematics education*, 208–233.
- Costa, E. J. F., Campos, L. M. R. S., & Dario Serey Guerrero, D. (2017). Computational thinking in mathematics education: A joint approach to encourage problem-solving ability. *2017 IEEE Frontiers in Education Conference (FIE)*, 1–8.
<https://doi.org/10.1109/FIE.2017.8190655>
- Cristancho Contreras, L. A. (2019). *Uso de software educativo en la enseñanza—Aprendizaje del álgebra escolar*. <http://hdl.handle.net/11349/14876>
- Crosetti, B. de B., & Ibáñez, J. M. S. (2016). La Investigación Basada en Diseño en Tecnología Educativa. *RiiTE Revista Interuniversitaria de Investigación en Tecnología Educativa*. <https://doi.org/10.6018/riite2016/260631>
- diSessa, A. A., & Cobb, P. (2004). Ontological Innovation and the Role of Theory in Design Experiments. *Journal of the Learning Sciences*, 13(1), 77–103.
https://doi.org/10.1207/s15327809jls1301_4

- Farfan Duma, W. E. (2014). *El desarrollo del pensamiento lógico y su incidencia en el proceso de enseñanza- aprendizaje en el área de matemática, de los niños del tercer año de básica la escuela "AGUSTÍN IGLESIAS", de la provincia del Azuay, cantón Sigsig, parroquia Ludo*. [bachelorThesis].
<https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/7937>
- Gómez, P., & González, M. J. (2018). Análisis cognitivo. En P. Gómez (Ed.), *Formación de profesores de matemáticas y práctica de aula: Conceptos y técnicas curriculares* (pp. 113–196). Universidad de los Andes.
<https://funes.uniandes.edu.co/funes-documentos/analisis-cognitivo/>
- Granados Romero, J., López Fernández, R., Avello Martínez, R., Luna Álvarez, D., Luna Álvarez, E., & Luna Álvarez, W. (2014). Las tecnologías de la información y las comunicaciones, las del aprendizaje y del conocimiento y las tecnologías para el empoderamiento y la participación como instrumentos de apoyo al docente de la universidad del siglo XXI. *MediSur*, 12(1), 289-294.
<https://www.redalyc.org/pdf/1800/180032233017.pdf>
- Kado, K. (2022). Teaching and learning the fundamental of calculus through Python-based coding. *International Journal of Didactical Studies*, 3(1), 15006.
<https://doi.org/10.33902/IJODS.202215006>
- Kaufmann, O. T., & Stenseth, B. (2020). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 52(7), 1029–1048. <https://doi.org/10.1080/0020739X.2020.1736349>
- Kelle, U., & Buchholtz, N. (2015). The Combination of Qualitative and Quantitative Research Methods in Mathematics Education: A “Mixed Methods” Study on the Development of the Professional Knowledge of Teachers. En A. Bikner-Ahsbahr, C. Knipping, & N. Presmeg (Eds.), *Approaches to Qualitative Research in Mathematics Education: Examples of Methodology and Methods* (pp. 321–361). Springer Netherlands. https://doi.org/10.1007/978-94-017-9181-6_12
- Kilhamn, C., Bråting, K., & Rolandsson, L. (2021). *Argumentos de los docentes para incluir la programación en la educación matemática*. 169–176.
- Llorens Largo, F. (2015). *Dicen por ahí... que la nueva alfabetización pasa por la programación*. <http://rua.ua.es/dspace/handle/10045/49092>
- Maracci, M., & Mariotti, M. A. (2013). Semiotic Mediation within an AT Frame. *International Journal for Technology in Mathematics Education*, 20(1), 21–26.
<https://www.learntechlib.org/p/167926/>

- Mariotti, M. A. (2002). La influencia de los avances tecnológicos en el aprendizaje de las matemáticas en los estudiantes. En *Manual de investigación internacional en educación matemática* (1a ed., p. 30). Routledge. <https://doi.org/10.4324/9781410602541>
- Mariotti, M. A. (2010). Proofs, Semiotics and Artefacts of Information Technologies. En G. Hanna, H. N. Jahnke, & H. Pulte (Eds.), *Explanation and Proof in Mathematics: Philosophical and Educational Perspectives* (pp. 169–188). Springer US. https://doi.org/10.1007/978-1-4419-0576-5_12
- Mariotti, M. A., & Bartolini Bussi, M. G. (2008). Semiotic mediation in the mathematics classroom: Artifacts and signs after a Vygotskian perspective. En *Handbook of International Research in Mathematics Education* (2a ed., p. 746). Routledge.
- Miles, M. B., & Huberman, A. M. (1994). *Análisis de datos cualitativos: Un libro de consulta ampliado*. Thousand Oaks, CA: Sage.
- Ministerio de Educación Nacional. (2006). Estándares Básicos de Competencias en Matemáticas. En *Estándares Básicos de Competencias en Lenguaje, Matemáticas, ciencias y Ciudadanas*. (pp. 46–95). Escribe y Edita. https://www.mineducacion.gov.co/1621/articles-340021_recurso_1.pdf
- Moreno Armella, L. (2002). Primer Seminario Nacional de Formación de Docentes en el Uso de la Tecnología: Instrumentos matemáticos computacionales. En *INCORPORACIÓN DE NUEVAS TECNOLOGÍAS AL CURRÍCULO DE MATEMÁTICAS DE LA EDUCACIÓN MEDIA DE COLOMBIA*. Ministerio de Educación Nacional.
- Moschkovich, J. N., & Brenner, M. E. (2000). Integrating a Naturalistic Paradigm Into Research on Mathematics and Science Cognition and Learning. En *Handbook of Research Design in Mathematics and Science Education* (1a ed., p. 30). Routledge. <https://www.taylorfrancis.com/chapters/edit/10.4324/9781410602725-21/integrating-naturalistic-paradigm-research-mathematics-science-cognition-learning-judit-moschkovich-mary-brenner>
- Orjuela Díaz, M., & Rodríguez Vanegas, E. A. (2022). *Desarrollo del pensamiento variacional y computacional a través del lenguaje Python en Educación Básica Secundaria en Colombia* [Pregrado, Universidad Abierta y a Distancia]. <http://repository.unad.edu.co/handle/10596/47745>
- Padilla Terán, V. M. (2022). *Influencia del uso de las TICS para mejorar el aprendizaje de las destrezas de relaciones lógico matemáticas en niños y niñas de 3 a 5 años de edad durante el año lectivo 2020-2021* [Maestría, Universidad Tecnica del Norte]. <https://repositorio.utn.edu.ec/handle/123456789/12096>

- Páez, M. L. C., & Gamboa, S. A. R. (2022). Desarrollo del pensamiento computacional y las competencias matemáticas en análisis y solución de problemas: Una experiencia de aprendizaje con Scratch en la plataforma Moodle. *Revista Educación*, 46(1), 1–30. <https://www.redalyc.org/journal/440/44068165016/html/>
- Papert, S. (1980). *Children, Computers, and Powerful Ideas*. Basic Books.
- Perez, T., & da Rosa, S. (2017). *MATEMÁTICA Y PROGRAMACIÓN: UNA EXPERIENCIA INTERDISCIPLINARIA E INTERINSTITUCIONAL*. 212–220. <https://funes.uniandes.edu.co/wp-content/uploads/tainacan-items/32454/1169856/Perez2017Matem25C325A1tica.pdf>
- Piaget, J. (1972). *Psicología y epistemología*. Emecé.
- Pico, O. A. G., Guadalupe, J. I. A., Pulloquina, R. H. M., Moreano, J. A. C., & Salazar, J. R. (2018). Una aproximación a la aplicación de las TICS en la didáctica de la matemática. *Revista Boletín Redipe*, 6(12), Article 12. <https://revista.redipe.org/index.php/1/article/view/412>
- Porras, M. C. de. (2011). Instrumentos cognitivos en el pensamiento matemático. *Praxis & Saber*, 2(4), Article 4. <https://doi.org/10.19053/22160159.1125>
- Reeves, T. (2000). *Enhancing the Worth of Instructional Technology Research through “Design Experiments” and Other Development Research Strategies*. <https://www.semanticscholar.org/paper/Enhancing-the-Worth-of-Instructional-Technology-and-Reeves/c25f2726efbd87f20bc80772bd16ed16ba28111e>
- REEVES, T. (2006). Investigación de diseño desde una perspectiva tecnológica. En *Investigación en diseño educativo* (1a ed., p. 15). Routledge. <https://doi.org/10.4324/9780203088364>
- Reich, R. (1991). *Preparing Students for Tomorrow’s Economic World*. Educational Reform. Making Sense of It All.
- Salazar, J. V. F., & Fernández, V. N. (2017). Tecnologías y visualización en Educación Matemática: Desarrollo y desafíos actuales. *REMATEC*, 12(24), Article 24. <https://www.rematec.net.br/index.php/rematec/article/view/258>
- Segura, J. A., Nebot, M. Á. L., Mon, F. E., & Novella, M. G. V. (2019). El debate sobre el pensamiento computacional en educación. *RIED-Revista Iberoamericana de Educación a Distancia*, 22(1), Article 1. <https://doi.org/10.5944/ried.22.1.22303>
- Tapia Cortés, C., Navarro Rangel, Y., Serna Tuya, A. S. de la, Tapia Cortés, C., Navarro Rangel, Y., & Serna Tuya, A. S. de la. (2017). El uso de las TIC en las prácticas

académicas de los profesores de la Benemérita Universidad Autónoma de Puebla.
Revista electrónica de investigación educativa, 19(3), 115–125.

<https://doi.org/10.24320/redie.2017.19.3.1270>

The Committee on European Computing Education (CECE). (2017). *Informatics Education in Europe: Are We All In The Same Boat?* [Technical Report]. Association for Computing Machinery.

Uribe, C. E. V. (2018). Reformas de los currículos escolares en matemáticas en las Américas: El caso colombiano. *Cuadernos de Investigación y Formación en Educación Matemática*, 223–229. <https://revistas.ucr.ac.cr/index.php/cifem/article/view/34377>

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4), 5–23. <https://doi.org/10.1007/BF02504682>

Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3), 33–35.
<https://doi.org/10.1145/1118178.1118215>

Anexos

Anexo 1

Actividad #1 Python

Operaciones básicas con Python

1. Calcula la división entera de 45 entre 6
2. Calcula el residuo de la división entera de 45 entre 6
3. Realiza la siguiente operación en Python, donde $\div$ indica la división entera: $10 + 20 \div 7 - 2$
4. Realiza la siguiente operación en Python:

$$\frac{9 - \frac{25 + 5 - 2}{7 * 4}}{2^3}$$

Operadores de decisión (if, elif, else)

1. Haz que un usuario introduzca un número real y evalúa si dicho número es positivo, negativo o cero. Devuelve por pantalla el resultado en cada caso.
2. Haz que un usuario introduzca dos números enteros positivos. Comprueba si el primer número introducido por el usuario es mayor, menor o igual que el segundo número introducido por el usuario. Devuelve por pantalla el resultado en cada caso.

Pista: asegúrate de hacer uso de la función `int( )`.

Operadores de iteración (for, while)

1. Programar un bucle que cuente atrás del 5 al 1 y al final imprimir el mensaje ¡despegue!
2. Mostrar la tabla de multiplicar del número 6.
3. Hallar la suma de los números múltiplos de 5 menores a 50

Ejemplo: $5 + 10 + 15 \dots + 45$

4. Del ejercicio anterior, solicitar el número mayor hasta donde llegue la suma y el número base para los múltiplos a calcular, es decir que el usuario debe ingresar dos números, uno que determine los múltiplos y otro para el número más grande.

Anexo 2

Actividad #2 Python

Operaciones básicas con Python

1. Realiza la siguiente operación en Python:

$$\frac{2 + 2^3 - 2(2 - 2^5) + 2^2 * 4 + 2}{2(2 * 2 - 2^4) + 2^2 * 2 - 2}$$

2. Crea una calculadora básica en Python, la cual le pida al usuario ingresar dos números, luego escoger entre cuatro operaciones posibles (suma, resta, multiplicación y división) y después resolver la operación escogida usando los dos números que ingresó, a la final se debe mostrar el resultado

Pista: la variable numérica debe ser de tipo float()

Operadores de decisión (if, elif, else)

1. Haz que un usuario introduzca dos números enteros positivos. Suponiendo que el primer número introducido por el usuario es mayor o igual al segundo número introducido por el usuario, comprueba que la división del primer número entre el segundo número es exacto. En caso de que la división sea exacta, devuelve el cociente por pantalla e indica que la división en efecto es exacta. En caso de la división no ser exacta, devuelve el cociente y el resto por pantalla e indica que la división entre los dos números no es exacta
2. Haz que un usuario introduzca dos números enteros positivos. Comprueba si el mayor es múltiplo del menor. Devuelve por pantalla el resultado en cada caso.

Operadores de iteración

1. Haz que el usuario introduzca números enteros por teclado. Mientras el usuario no introduzca el 0, muestra si el número introducido es par o impar.
2. Haz que el usuario introduzca su edad y el año actual. Imprime todos los años que han pasado desde su año de nacimiento hasta el año actual (ambos incluidos).

Anexo 3

Enlace de la encuesta inicial realizada a los estudiantes:

<https://forms.gle/BbP3oh4uoLvBJSfj9>

Anexo 4

Enlace de la encuesta final realizada a los estudiantes:

<https://forms.gle/AKievFDvHZKZAHqH7>

Anexo 5

Hoja guía actividad A0:

https://colab.research.google.com/drive/1WeERrNVzvLgj9920z4Gt8up7XT6_f1_d?hl=es#scrollTo=HjKOXXcnX92N&uniqifier=2

Anexo 6

Enlace a carpeta de Drive con las hojas de trabajo de los estudiantes y respuestas de las encuestas:

https://drive.google.com/drive/folders/1knXZJWLFV06U7ErU1B-ULq6KxH_X4GE4?usp=sharing

Anexo 7

Enlace de las diapositivas utilizadas para la primera sesión:

<https://docs.google.com/presentation/d/1c91nHkTMXZyh4-lL0hGYyx9WlpEXv9XXLT5lu3RGBIY/edit>