

CONSTRUCCIÓN DE UN APLICATIVO GRATUITO PARA EL ESTUDIO DE LAS PROPIEDADES DE LAS ESTRUCTURAS ALGEBRAICAS



**UNIVERSIDAD PEDAGOGICA
NACIONAL**

Educadora de educadores

Trabajo realizado por:

Diego Alejandro Alonso Montenegro

Cód. 2018140002

Nicolas David Mendivelso Vanegas

Cód. 2018140046

UNIVERSIDAD PEDAGÓGICA NACIONAL

FACULTAD DE CIENCIA Y TECNOLOGÍA

DEPARTAMENTO DE MATEMÁTICAS

LICENCIATURA EN MATEMÁTICAS

BOGOTÁ D. C.

2023 – I

CONSTRUCCIÓN DE UN APLICATIVO GRATUITO PARA EL ESTUDIO DE LAS PROPIEDADES DE LAS ESTRUCTURAS ALGEBRAICAS

Trabajo realizado por:

Diego Alejandro Alonso Montenegro

Cód. 2018140002

Nicolas David Mendivelso Vanegas

Cód. 2018140046

Asesor del trabajo:

William Alfredo Jiménez Gómez

Trabajo de grado presentado como requisito parcial para
obtener el título de Licenciados en Matemáticas

UNIVERSIDAD PEDAGÓGICA NACIONAL

FACULTAD DE CIENCIA Y TECNOLOGÍA

DEPARTAMENTO DE MATEMÁTICAS

LICENCIATURA EN MATEMÁTICAS

BOGOTÁ D. C.

2023 – I

Agradecimientos

Agradecemos a Dios por darnos la oportunidad de culminar este proyecto tan importante para nuestra carrera y nuestra vida.

Agradecemos infinitamente a nuestras familias por su apoyo constante y acompañamiento durante este proceso.

A nuestro asesor William Jiménez infinitas gracias, por haber sido nuestro guía y orientador en el desarrollo de este trabajo. Por todo su tiempo, dedicación y empeño para con nosotros.

A cada uno de nuestros amigos y demás personas que nos acompañaron en el transcurso de la carrera y nos permitieron tener experiencias significativas.

Diego Alejandro Alonso Montenegro

Nicolas David Mendivelso Vanegas

TABLA DE CONTENIDO

ÍNDICE DE ILUSTRACIONES.....	6
ÍNDICE DE TABLAS	7
Introducción	8
Capítulo I. Preliminares.....	10
Planteamiento del problema	10
Objetivos.....	11
Objetivo General	12
Objetivos Específicos.....	12
Capítulo II. Marco de Referencia.....	13
La enseñanza mediada por las tecnologías	13
Búsqueda de softwares asociados al álgebra	14
Investigación sobre las limitaciones de los softwares de estructuras algebraicas	20
Historia del álgebra y de las estructuras algebraicas	21
Estructuras algebraicas.....	24
Estructuras algebraicas con una operación interna (+).	25
Estructuras algebraicas con dos operaciones internas (+, *)......	26
Softwares	26
Ventajas y desventajas de los softwares hacia la construcción del aplicativo.....	28
Capítulo III. Aplicativo SEPPEA.....	34
Generalidades de GeoGebra	34
a) Vista Algebraica:	35
b) Vista Gráfica 2D:.....	36
c) Vista CAS:	37
d) Hoja de Cálculo:	39
Primera versión del programa.....	39
a) Propiedad conmutativa para la operación +:	43
b) Propiedad asociativa para la operación +:	45
c) Propiedad del elemento neutro para la operación +:	46
d) Propiedad del elemento inverso para la operación +:.....	47
e) Propiedad distributiva de * con respecto a +.....	51

Google Sites	54
Resultados del pilotaje.....	55
Modificaciones al programa	56
Capítulo IV. Conclusiones	59
Referencias	63
Anexos.....	66
Anexo 1. Lista comandos de GeoGebra	66
Anexo 2. Respuestas del pilotaje.....	67
Anexo 3. Código de programación de HTML del aplicativo SEPPEA.....	67

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1. GRÁFICA NÚMERO DE SOFTWARES DE ENSEÑANZA E INVESTIGACIÓN. CONSTRUCCIÓN PROPIA EN METACHART.....	17
ILUSTRACIÓN 2. SOFTWARE CAMINOS CERRADOS. TOMADO DE GALÁN Y PATIÑO (2019).....	18
ILUSTRACIÓN 3. GRÁFICO DE BARRAS SOFTWARES DM VS. PC. CONSTRUCCIÓN PROPIA EN EXCEL	19
ILUSTRACIÓN 4. VISTAS DE GEOGEBRA	34
ILUSTRACIÓN 5. VISTA PRINCIPAL DE GEOGEBRA	35
ILUSTRACIÓN 6. GEOGEBRA REPRESENTACIÓN SEGMENTO	36
ILUSTRACIÓN 7. GEOGEBRA REPRESENTACIÓN FUNCIONES.....	36
ILUSTRACIÓN 8. BARRA DE HERRAMIENTAS VISTA GRÁFICA 2D.....	37
ILUSTRACIÓN 9. BARRA DE TAREAS VISTA CAS	38
ILUSTRACIÓN 10. EJEMPLO COMANDOS VISTA CAS	38
ILUSTRACIÓN 11. BARRA DE HERRAMIENTAS HOJA DE CÁLCULO	39
ILUSTRACIÓN 12. TABLAS DE CAYLEY HOJA DE CÁLCULO.....	40
ILUSTRACIÓN 13. MATRICES ASOCIADAS A LAS TABLAS	41
ILUSTRACIÓN 14. MATRIZ M1	42
ILUSTRACIÓN 15. MATRIZ M2.....	42
ILUSTRACIÓN 16. MATRIZ M3	43
ILUSTRACIÓN 17. MATRIZ FINAL	43
ILUSTRACIÓN 18. COMANDO 1	44
ILUSTRACIÓN 19. COMANDO 3.....	44
ILUSTRACIÓN 20. COMANDO 4.....	44
ILUSTRACIÓN 21. COMANDO 1.....	45
ILUSTRACIÓN 22. COMANDO 7.....	46
ILUSTRACIÓN 23. COMANDO 8.....	46
ILUSTRACIÓN 24. COMANDO 22.....	46
ILUSTRACIÓN 25. COMANDO 27.....	47
ILUSTRACIÓN 26. COMANDO 34.....	47
ILUSTRACIÓN 27. MATRICES.....	48
ILUSTRACIÓN 28. COMANDO 42.....	48
ILUSTRACIÓN 29. COMANDO 49.....	48
ILUSTRACIÓN 30. COMANDO 52.....	49
ILUSTRACIÓN 31. COMANDO 53 Y 54.....	49
ILUSTRACIÓN 32. COMANDO 55 Y 56.....	50
ILUSTRACIÓN 33. COMANDO 57.....	50
ILUSTRACIÓN 34. COMANDO 62.....	50
ILUSTRACIÓN 35. COMANDO 116.....	51
ILUSTRACIÓN 36. COMANDO 121.....	52
ILUSTRACIÓN 37. COMANDO 129.....	52
ILUSTRACIÓN 38. COMANDO 136.....	52
ILUSTRACIÓN 39. EJEMPLO 3X3	53
ILUSTRACIÓN 40. NÚMERO DE CLASE IGUAL A 1.....	54
ILUSTRACIÓN 41. NUEVA INTERFAZ.....	56
ILUSTRACIÓN 42. VISTA GOOGLE SITES COMPUTADOR	57
ILUSTRACIÓN 43. VISTA GOOGLE SITES CELULAR	58

ÍNDICE DE TABLAS

TABLA 1. SOFTWARES ALGEBRAICOS ENSEÑANZA VS INVESTIGACIÓN	15
TABLA 2. CLASIFICACIÓN DE LAS ESTRUCTURAS SEGÚN SUS PROPIEDADES	25
TABLA 3. CLASIFICACIÓN DE ESTRUCTURA CON DOS OPERACIONES.....	26
TABLA 4. VENTAJAS Y DESVENTAJAS DE SOFTWARES DE PROGRAMACIÓN.....	32

INTRODUCCIÓN

El desarrollo de este documento constituye un requisito para obtener el título de Licenciados en Matemáticas, como también representa una parte fundamental de nuestra formación académica y el producto de las experiencias, conocimientos y habilidades adquiridas a lo largo de estos años en la Universidad Pedagógica Nacional. En ese sentido, resulta pertinente que, como futuros educadores matemáticos, nos involucremos en el desarrollo de propuestas para la promoción de la Educación Matemática, siendo un claro ejemplo de participación en la comunidad matemática, esta propuesta que está orientada hacia el estudio de las propiedades de las estructuras algebraicas.

Es importante mencionar que referirnos a este objeto matemático conlleva el estudio de conceptos, procesos, evolución histórica y otros aspectos que se dejarán evidenciados a lo largo del documento con la intención de concretar una propuesta de enseñanza y aprendizaje apoyada en las TIC sobre el estudio de las operaciones internas involucradas en las estructuras algebraicas, las propiedades inmersas allí y la clasificación según el tipo de estructura acorde a las mismas.

En primer lugar, se alude de manera clara en el primer capítulo lo referente al planteamiento del problema y cómo se consolidó una idea para el trabajo de grado. Así mismo, se presentan los objetivos (general y específicos) que establecen el propósito y los alcances que se pretenden con la realización de esta propuesta relacionada a la construcción de un aplicativo gratuito para el estudio de las propiedades de las estructuras algebraicas.

En el segundo capítulo, que desarrolla todo lo relacionado al marco teórico, se hace una revisión de los conceptos fundamentales en torno al trabajo, como lo son álgebra, tecnología, estructuras algebraicas y software. Así, se aproximan dichos conceptos desde una mirada histórica frente a su avance, como también desde las Matemáticas, tomando como referencia algunas definiciones para acotar la propuesta de trabajo a los intereses establecidos.

Por otra parte, el capítulo tres aborda principalmente la explicación de la construcción de nuestro aplicativo llamado *Software Educativo Para las Propiedades de las Estructuras Algebraicas* (SEPPEA). Previo a eso, se hace una contextualización sobre el software escogido para tal fin, el cual es GeoGebra y sus herramientas más importantes para la creación del programa. Además, se presentan los resultados de un pilotaje realizado en un espacio académico de la licenciatura, esclareciendo algunas modificaciones realizadas al programa según los resultados obtenidos en esta prueba piloto.

Finalmente, el capítulo cuatro presenta las conclusiones de nuestro trabajo de grado, dejando claro el cumplimiento de los objetivos especificados, su relevancia en nuestra formación matemática y otros comentarios acerca de lo que nos dejó la realización de este trabajo, la manera de desarrollarlo y la proyección que se tenía con el mismo.

CAPÍTULO I. PRELIMINARES

Planteamiento del problema

En el trabajo vivenciado en los cursos *Sistemas Numéricos* que hace parte del plan de estudio de la LM en el segundo semestre del pensum académico de la cuarta versión, cursado en el semestre 2018 – II y el espacio electivo *Construcción de Estructuras Algebraicas*, se percibió la importancia de la construcción y verificación de conjuntos numéricos que cumplen ciertas características y propiedades, las cuales determinan una estructura algebraica, como lo pueden ser los números reales (\mathbb{R}).

Al momento de estudiar el conjunto y de demostrar las propiedades que cumple, los cálculos implicados en ello pueden llegar a ser tediosos o extensos. Por ejemplo, una forma de demostrar la propiedad asociativa de la suma en una estructura algebraica de 5 elementos consiste en verificar todas las posibles combinaciones de ternas de los elementos del conjunto, es decir, que se pueden presentar 5^3 posibilidades que deben satisfacer la igualdad de la propiedad asociativa $a + (b + c) = (a + b) + c$.

Además, si durante el proceso de enseñanza y aprendizaje estos cálculos se realizan a “lápiz y papel”, pueden generar mayores dificultades en el proceso de comprensión y consolidación de las propiedades, puesto que se centra la atención mayoritariamente en el desarrollo del conocimiento procedimental, que según Arcavi (2018) este componente está asociado al “saber cómo” del estudiante, que ayuda a la consolidación del conocimiento conceptual, el cual desde nuestra perspectiva como futuros educadores matemáticos se puede enriquecer con los ambientes computacionales, siendo una vía para entrelazar dichos componentes en la praxis del docente. De este primer acercamiento, surge la idea de simplificar este problema haciendo uso de diferentes herramientas tecnológicas dado que hemos evidenciado la utilidad y el potencial que tienen éstas

para sintetizar este tipo de cálculos. Debido a esto, el estudiante se centraría más en reconocer los diferentes tipos de propiedades y de estructuras que se pueden determinar acorde a las propiedades halladas.

Al conocer que en el libro *Actividades matemáticas para el desarrollo de procesos lógicos: Representar estructuras algebraicas finitas y enumerables* de Luque, Jiménez y Ángel (2009) y algunos otros trabajos como, por ejemplo, *Software para la construcción de operaciones distributivas con respecto a otras dadas* de Molina, Luque y Ángel (2005), habían construido un aplicativo para hacer los cálculos respectivos de las propiedades, identificamos que estos tenían algunas limitantes. En el caso del primer aplicativo mencionado es necesario contar con el libro en el que está el CD para instalarlo. Mientras que el otro programa, es de acceso libre para los estudiantes de la Licenciatura en Matemáticas de la Universidad Pedagógica Nacional, aunque no se ha podido tener acceso al mismo. De manera general, los programas presentan incompatibilidad con algunas versiones de los computadores y no se encuentran en alguna plataforma o página en línea. Por lo mencionado hasta ahora, surge la siguiente pregunta de interés:

¿Cómo sería posible diseñar un software de acceso libre y en línea para hacer los cálculos en relación con las propiedades de las estructuras algebraicas finitas con una y dos operaciones, implementando y determinando su impacto con estudiantes que estén finalizando el curso de *Sistemas Numéricos* o *Construcción de Estructuras Algebraicas*?

Objetivos

Desde lo analizado y expuesto hasta el momento, es pertinente establecer diferentes objetivos que satisfagan las necesidades planteadas para el desarrollo del trabajo o producto, de tal modo que se pueda llevar a cabo este de la mejor manera posible.

Objetivo General

Desarrollar un aplicativo en el cual se promueva el trabajo de las estructuras algebraicas de manera dinámica e interactiva con el fin de analizar sus propiedades, el tipo de estructura que se determina, las operaciones y las demostraciones que se ven inmersas en la validación de los procesos.

Objetivos Específicos

- Realizar una consulta en la cual se indaguen los autores que han trabajado con las estructuras algebraicas, la tecnología y los conceptos más relevantes de estas.
- Estudiar los diferentes softwares, analizando aspectos como la accesibilidad, el lenguaje de programación y la portabilidad, con el fin de seleccionar el más adecuado para el desarrollo del producto propuesto.
- Construir una primera versión del aplicativo, de acuerdo con el software escogido para ello y adecuarlo a la aplicación en línea gratuita *Google Sites*.
- Aplicar el recurso diseñado en estudiantes que estén cursando *Sistemas Numéricos* o *Construcción de Estructuras Algebraicas*.
- Complementar el aplicativo con algunas mejoras y ajustes en función a la implementación realizada.

De esta manera, atendiendo a los objetivos anteriores, se abordará el álgebra desde la tecnología para tener un campo de visión más amplio frente a los softwares desarrollados acerca de las estructuras algebraicas y la evolución matemática que ha tenido el concepto a lo largo de la historia. De esta manera, esto nos permitirá realizar un análisis de los softwares alrededor de las estructuras algebraicas, con el fin de detallar las limitantes presentes y dar solución a estas a través de nuestro aplicativo.

CAPÍTULO II. MARCO DE REFERENCIA

La enseñanza mediada por las tecnologías

Desde hace unos años el impacto de la tecnología en los diferentes entornos o contextos cotidianos cada vez es más amplio. Así mismo, la educación se rige fundamentalmente por lo que se impone en la sociedad, y es de allí en donde surge la relación entre la tecnología y la educación, haciéndose constantemente más cercana. Por consiguiente, los docentes, los directivos, la comunidad y los entes estatales educativos son los encargados de propiciar la integración mancomunada entre el currículo educativo, el PEI, las planeaciones, las mallas curriculares y las actividades desarrolladas por los maestros con el uso de las TIC. A causa de ello “no es de extrañar que acaparen gran parte de la investigación en este último ámbito” (Barroso, Cejas y Navío, 2016, p. 106).

En medio de este auge a la incorporación de las nuevas tecnologías, los docentes son los responsables de diseñar e implementar estas en el salón de clases. Sin embargo, se observan casos en los cuales los maestros, según Gamboa (2007), rechazan el uso de las calculadoras y los computadores porque creen que su uso inhibirá otras habilidades (p. 8). Por lo anterior, la Educación Matemática debe fortalecer la implementación de nuevas estrategias y herramientas metodológicas por medio de las TIC, que favorezcan el desarrollo de procedimientos poco trabajados en la enseñanza tradicional de las matemáticas, como lo puede ser el proceso de razonamiento y justificación, ya que en la enseñanza tradicional se tiende a priorizar la mecanización de los procesos, sin indagar realmente porque funciona o se da cierto concepto.

De allí se hace necesario que el maestro adquiera un perfil transferencial, flexible y polivalente, adecuándose a la diversidad y los continuos cambios que se dan en la sociedad a través de la tecnología. Esto se da cuando el rol del docente evoluciona en la medida que deja de lado el

autoritarismo con el alumno, y se preocupa más por establecer buenos puentes de comunicación y relación con ellos, apoyándose de los recursos tecnológicos para cumplir eso.

De acuerdo con Cejas, Navío y Barroso (2016) es más bien un facilitador del aprendizaje, que acentúa la importancia de la evaluación continua, diseña escenarios mediados de aprendizaje, y se convierte en un asesor y orientador de la formación del alumno, quien adquiere mayor protagonismo y responsabilidad en su propio estudio (p.108). También cabe resaltar que, al generar esos puentes de comunicación con el estudiante, se percatará el docente de dificultades y obstáculos en el aprendizaje de ellos, lo cual será ideal para promover el uso de otras herramientas y técnicas de evaluación apoyadas por las TIC.

Búsqueda de softwares asociados al álgebra

Siguiendo la idea del apartado anterior, se hará una investigación en diferentes áreas del álgebra en general, buscando aplicativos o softwares desarrollados dentro de estos campos temáticos, enfocados en la educación matemática o en las matemáticas. Esto con el fin de conocer qué aplicativos se han desarrollado hasta el día de hoy, y específicamente, de las estructuras algebraicas, lo que contribuye a nutrir la construcción de nuestro programa por medio de las falencias y cualidades encontradas.

De esta manera, es importante delimitar la zona de investigación en el álgebra, por lo que se considerarán 17 categorías generales extraídas de algunos libros consultados referentes a los contenidos del álgebra, teniendo en cuenta un determinado rango de tiempo el cual comprende desde el año 2010 hasta el año 2021.

Al contar con las categorías se decidió utilizar palabras claves para la búsqueda, tales como aplicativo, software, programa, tecnología y el nombre de cada categoría. Además, frente al rango de tiempo no se hizo restricción alguna, aunque se observó que el desarrollo se encuentra enfocado desde el siglo XXI.

A continuación, se muestra la información obtenida en una tabla para cada categoría, después de la búsqueda realizada en diferentes bases de datos las cuales fueron *Google Académico*, *Mendeley*, repositorios de universidades y también en navegadores como *Google Chrome*, *Google Play* y *App Store*. En su mayoría, los aplicativos encontrados están en español.

Categoría	Enseñanza	Investigación
Geometría Algebraica	1	1
Teoría Axiomática de Conjuntos	3	0
Teoría de Categorías	1	2
Álgebra Diferencial	1	0
Campos y Anillos	1	0
Grupos	2	1
Álgebra Homológica	0	0
Retículos	0	7
Álgebra de Lie	0	0
Álgebra Lineal	4	3
Teoría de Matrices	2	0
Álgebra Abstracta	3	1
Polinomios	10	3
Teoría de la Representación	0	0
Álgebra de Boole	5	1
Raíces	3	0
Teoría de Grafos	4	8

Tabla 1. Softwares algebraicos Enseñanza vs. Investigación

Para realizar un análisis de los datos anteriores, se presentará un diagrama de barras en el cual se representa la cantidad de softwares encontrados con respecto a cada una de las categorías mencionadas. Además, se muestra una clasificación de los diferentes softwares acorde al enfoque mencionado previamente (enseñanza vs. investigación). Previo a esto, es importante aclarar que concepto o definición asociaremos a enseñanza e investigación, puesto que se dotan de diferentes características que hacen que tengas particularidades entre una y otra definición y algunas diferencias entre dichos conceptos.

En primer lugar, consideraremos la enseñanza como un medio a través del cual se forma al estudiante en competencias y procesos, los cuales responden a las necesidades de la vida, el trabajo y el mundo, el cual se puede apoyar de recursos, estrategias y ambientes computacionales. Por otra parte, la investigación conforme a Ávila (2016) se concibe como una metodología que tiene como objetivo ampliar el conocimiento del área en estudio, aunque los fines de la investigación pueden variar de acuerdo con los intereses que establezca el investigador. Dicho esto, también es importante resaltar ciertas distinciones que se mencionan a continuación:

- La enseñanza se enfatiza en crear conocimientos de manera crítica, así como de fomentar capacidades y valores, mientras que, la investigación descubre conocimientos.
- El proceso de enseñanza mediado por el docente necesita de los conocimientos propios del educador y los conocimientos que posea el estudiante. Por otra parte, el investigador tiene como base para su ejercicio los conocimientos previos que se adecúen a sus intereses.
- Para un docente la comunicación es esencial en el proceso de enseñanza, caso contrario a un investigador que no es una habilidad imprescindible, pero sí necesaria.

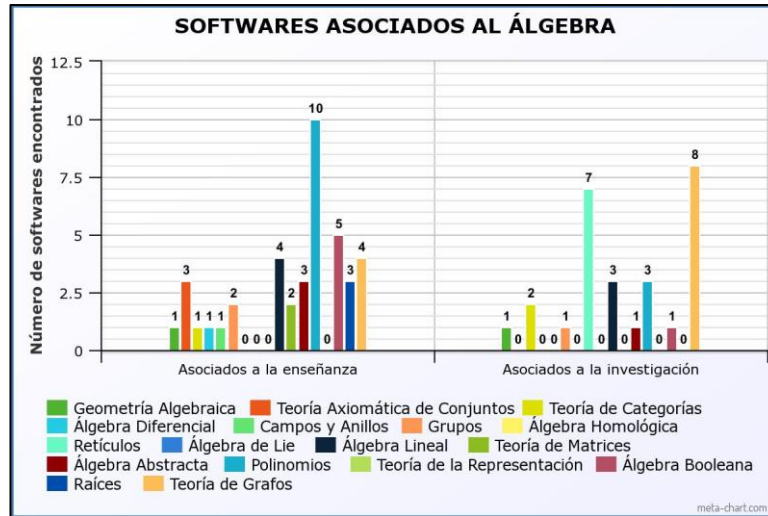


Ilustración 1. Gráfica número de softwares de Enseñanza e Investigación. Construcción propia en MetaChart

Observando los datos en el cuadro y el diagrama de barras, se perciben que hay ciertas categorías en las que predomina el uso y desarrollo de softwares para la enseñanza del álgebra como lo son polinomios, teoría de grafos, retículos, entre otros. Mientras que se tienen categorías en las cuales no se tiene un gran o ningún desarrollo y adaptación de la tecnología para la construcción de softwares.

Por ejemplo, podemos evidenciar que en la categoría de retículos los softwares encontrados están dirigidos hacia la investigación, puesto que, en el software denominado *Caminos Cerrados* desarrollado por Galán y Patiño (2019), este se enfoca a la representación y visualización de algunos conceptos matemáticos como lo son la función de homotopía lineal, caminos cerrados en \mathbb{R}^2 (epicicloide, deltoide, curvas de Lissajous, entre otros) y homomorfismos inducidos.

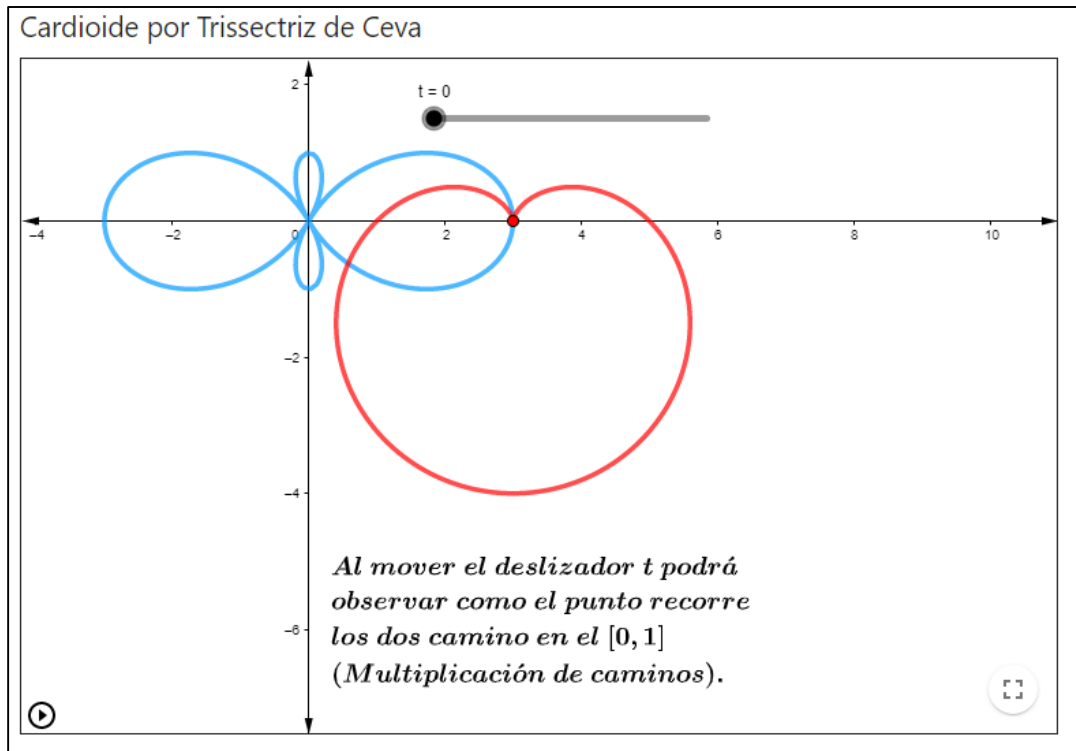


Ilustración 2. Software caminos cerrados. Tomado de Galán y Patiño (2019)

Además, si revisamos cada uno de los dispositivos en los cuales se construyeron los softwares de cada categoría encontramos que ninguno de estos se desarrolló para poder ser utilizados simultáneamente en varios dispositivos (computador, tabletas y celulares). Así mismo, se evidenció un bajo desarrollo de softwares para dispositivos móviles concentrando la gran mayoría en computadoras como se puede constatar en la Ilustración 3 debido a que estos brindan características con las que los dispositivos móviles no cuenta en ciertos casos, como lo son la capacidad para ejecutar distintas aplicaciones, la memoria con la que cuentan para la ejecutar programas, el acceso a plataformas que no están adaptadas para dispositivos móviles, entre otras. Así, estas características en muchos casos son necesarias para poder procesar y ejecutar los algoritmos que se encuentran detrás de los softwares matemáticos construidos. A partir de esto surge un interés particular en la construcción del aplicativo, el cual consiste en lograr que este se

pueda ejecutar tanto en dispositivos móviles como computadores, ya sea a partir de que el programa en el que se construya el aplicativo cuenta con esta característica o debamos tomar una herramienta externa al programa la cual nos facilita la adaptabilidad que deseamos para el aplicativo.

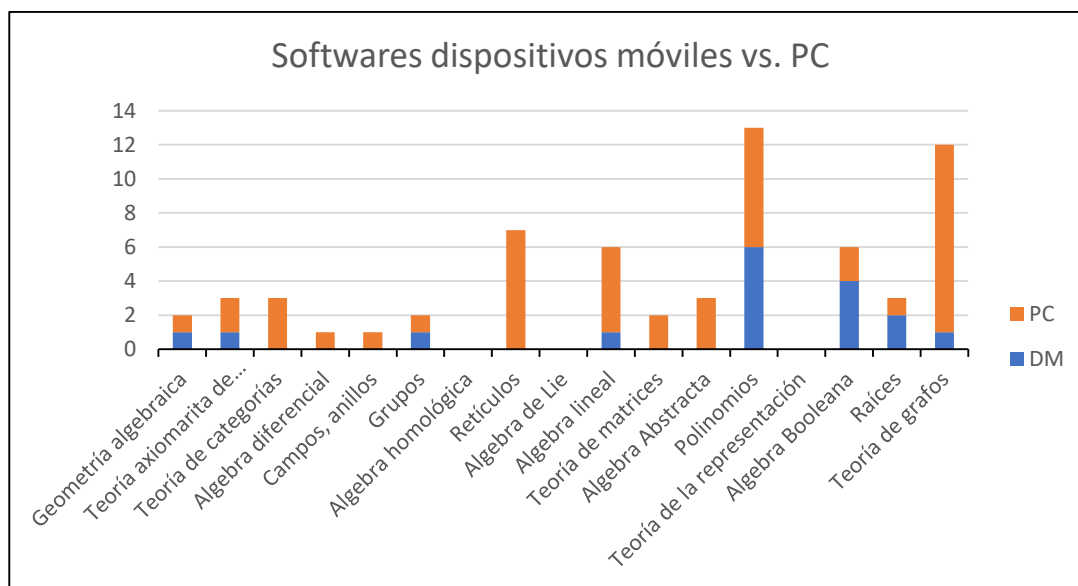


Ilustración 3. Gráfico de barras softwares DM vs. PC. Construcción propia en Excel

De esta manera, al evidenciar la escasez de softwares en algunas categorías, en especial el álgebra abstracta y particularmente de las estructuras algebraicas, surge un mayor interés por aproximar esta temática a las aulas a través de las TIC, consiguiendo que el programa sea un apoyo para el docente en el proceso de enseñanza y aprendizaje de las matemáticas, y al mismo tiempo, para los estudiantes como una herramienta que les ayude en su proceso de investigación a favor de la construcción del conocimiento matemático sobre el estudio de las propiedades de las estructuras algebraicas.

Esto contribuye a dar un motivo más para el desarrollo de la idea principal del trabajo, la cual es construir un aplicativo sobre las propiedades de las estructuras algebraicas. Además, esto nutriría el campo temático en relación con los softwares que se tienen para el estudio y enseñanza de las estructuras algebraicas, debido a que se puede concluir por medio del gráfico que, dentro de

este, se cuenta con muy poco desarrollo. Por esto mismo, el paso a seguir es desarrollar una investigación en torno a la relación existente entre la tecnología y las estructuras algebraicas.

Investigación sobre las limitaciones de los softwares de estructuras algebraicas

A raíz de lo concluido hasta el momento y nuestro interés inmediato por las estructuras algebraicas, es necesario recopilar una serie de información concerniente a los estudios o trabajos más relevantes para nosotros, desarrollados en torno a la enseñanza de las estructuras algebraicas mediante el uso de la tecnología.

Así es como, al haber realizado una investigación se encontró primero que en el contenido del libro *Actividades matemáticas para el desarrollo de procesos lógicos: Representar estructuras algebraicas finitas y enumerables* de la Universidad Pedagógica Nacional, se había construido un aplicativo para hacer los cálculos respectivos a las propiedades, identificando que este tenía algunas limitantes, tales como las que se mencionan en el primer apartado.

A su vez, en el trabajo realizado en la maestría de matemática aplicada de la Universidad Central de Cuba, encontramos que Martínez (2010) quiere abordar la enseñanza de las estructuras algebraicas mediante un sistema inteligente, y para ello hace uso de un aplicativo llamado HESEI. Esta herramienta le permite al docente evaluar un contenido por medio de preguntas que se realizan al estudiante y que a través de HESEI, se tiene la oportunidad y la posibilidad de desarrollar un proceso de aprendizaje más elaborado, puesto que el docente controla, a través de dicho sistema, el tiempo y la secuenciación del aprendizaje para cada estudiante.

Sin embargo, para hacer uso de esta herramienta es necesario contar con unos requisitos mínimos que son: Procesador Intel 486/100 MHz o superior, 16 Mb de memoria RAM, sistema operativo Windows 95 o superior y Monitor SVGA. Además, HESEI está programado de tal manera que las propiedades de las estructuras algebraicas no se comprueban y es el maestro, más no el usuario (estudiante), quien las define.

Teniendo en cuenta las restricciones encontradas en los softwares anteriormente analizados, estas serán importantes a la hora de la construcción del aplicativo, de modo que esté presente una estructura más robusta, completa y pertinente para lo que se pretende trabajar con el aplicativo. Además, observamos que estas restricciones están asociadas a las herramientas usadas para la elaboración del software, por lo que más adelante se mostrarán las ventajas y desventajas que conlleva trabajar o no con determinados programas. No obstante, primero se abordará el estudio del álgebra a través de los tiempos, siendo allí de donde surgen las estructuras algebraicas, haciendo un estudio matemático sobre dicho concepto, dado que en el aplicativo se tienen en cuenta definiciones asociadas a las propiedades y la clasificación de las estructuras algebraicas, siendo importante este recorrido histórico para establecer qué definiciones tomaremos como referencia para el desarrollo del programa.

Historia del álgebra y de las estructuras algebraicas

Sabemos que el desarrollo matemático es un proceso continuo de construcción, puesto que cada avance se soporta en saberes previos como hace referencia Newton con su frase “si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes”, mostrándonos que el conocimiento matemático no es una aparición espontánea, sino una estructura en donde cada uno de los pilares representan los aportes de muchos matemáticos. Esto quiere decir que para desarrollar nuestro trabajo tendremos como soporte los programas encontrados y el conocimiento matemático relacionado a las estructuras algebraicas necesario para la construcción del programa.

Con base en esto, el álgebra surge como una generalización y extensión de la aritmética, por esto se puede decir que los inicios del álgebra datan de los números y las operaciones con estos, como se reflejaba en las civilizaciones hindú, árabe, egipcia, babilónica y griega. Para las primeras civilizaciones mencionadas el sentido del álgebra estaba asociado a problemas de reparticiones, transacciones, delimitación de terrenos, entre otros. Mientras que, en la civilización griega, se hizo

énfasis en la parte geométrica, puesto que cualquier problema debía ser demostrado a partir de áreas y volúmenes. Esto llevó a que no se tuvieran en cuenta los números negativos e imaginarios como soluciones, ya que no tienen representación geométrica. Además, la relación entre la geometría y el álgebra que daban los griegos causó que el álgebra no se viera como una rama independiente de la matemática. Sin embargo, no fue sino hasta el siglo III D.C. que muchos matemáticos ubican los principios del álgebra desligada de la geometría, con los aportes publicados por Diofanto en su obra *Aritmética*, en la cual él introduce un símbolo aritmético muy elemental para representar la incógnita en los problemas, la abreviación de potencias y además Diofanto no centraba la atención en la representación geométrica.

Después de esto, los aportes matemáticos relacionados al álgebra se centran en civilizaciones como la hindú, la árabe, la mesopotámica, entre otras. Lo más significativo que realizaron estas civilizaciones, fue utilizar el álgebra para la resolución de ecuaciones, hasta ese entonces, ecuaciones de primer y segundo grado. Esto causa que el estudio alrededor del álgebra desde el siglo VII hasta el siglo XVIII se centre en la búsqueda de fórmulas algebraicas que den soluciones a ecuaciones de grado n .

No fue hasta el siglo XVII que los matemáticos Tartaglia, Cardano y Del Ferro encontraron la fórmula para las ecuaciones de grado 3, siendo este el motivo para que en el siguiente siglo los matemáticos centrarán sus trabajos en encontrar las fórmulas para ecuaciones de grado superior. A pesar de esto, se obstaculiza el estudio para las ecuaciones de grado superior y no fue hasta el año 1825 que el matemático Niels Abel probó que no existe una fórmula general para resolver polinomios con grado igual o mayor a 5. Por otra parte, esto llevó a que el álgebra trascendiera más allá del análisis de ecuaciones, en aspectos como sistemas matemáticos abstractos, el álgebra de Lie y Boole y la teoría de grupos.

La demostración realizada por Abel da un cierre al álgebra tradicional asociada a las ecuaciones, pero abre la puerta al álgebra moderna o abstracta, gracias a que Galois da un agregado a la demostración que conforme a Pérez (2018) él “determinó qué propiedades deben tener los coeficientes de cualquier ecuación y de cualquier grado para poder tener una fórmula que permita obtener una solución general” (p. 42). Lo importante de su avance es que Galois se percató de que los coeficientes de las ecuaciones auxiliares utilizados en las fórmulas de solución de las ecuaciones de grado igual o mayor a 5 no eran invariantes bajo subgrupos del grupo de permutaciones de las raíces. Lo que quiere decir, que el grupo puede o no poseer los subgrupos requeridos para poder encontrar la fórmula algebraica de la ecuación, así que se involucró el concepto de grupo y subgrupo relacionado a las raíces en la demostración y lo cual conlleva a que se considere a Galois como el padre del álgebra abstracta.

A partir de la construcción y estudio del concepto de grupo que introduce Galois y de la ley de composición, nace el álgebra abstracta que se encarga del estudio de las estructuras algebraicas. Esto a partir de que los grupos planteados por Galois se pueden analizar sin necesidad de conocer las raíces, lo que llevó a encontrar las características que posee cualquier grupo, surgiendo de lo anterior la primera definición de grupo: Sea G un conjunto no vacío y una operación binaria definida en G , las propiedades o axiomas del grupo son:

- i. Propiedad clausurativa: $\forall a, b \in G, a * b \in G$
- ii. Propiedad asociativa: $\forall a, b, c \in G, a * (b * c) = (a * b) * c$
- iii. Propiedad modulativa: \exists un $e \in G$ tal que $a * e = a$
- iv. Propiedad invertiva: $\forall a \in G, \exists x \in G$ tal que $x * a = e$
- v. Propiedad conmutativa o abeliana: $\forall a, b \in G, a * b = b * a$

Así pues, dirigiendo el estudio de grupo hacia algo más general, se llega al concepto de estructura algebraica, el cual se define inicialmente como un conjunto no vacío con una colección de elementos, operaciones y relaciones. Más adelante, hacia la época de los Bourbaki se desarrolla una definición más formal que según Pérez (2018) enuncia lo siguiente:

Se entiende por estructura a un conjunto de elementos cuya naturaleza no está especificada.

Para definir una estructura se da, en primer lugar, una serie de relaciones en las que han de intervenir estos elementos. En segundo lugar, se postulan sobre estas relaciones ciertas condiciones (axiomas de la estructura) que deben ser satisfechas (p. 51).

Un concepto primordial para su desarrollo fue la ley de la composición, la cual fue trabajada por los matemáticos de la escuela inglesa, aunque llevándose el reconocimiento de este concepto por parte del grupo Bourbaki. De manera sencilla, se entiende la ley de composición como una función que toma dos elementos de dos conjuntos dados y los asigna a un tercer elemento, que hace parte de otro conjunto. Así, si ese tercer elemento se encuentra en el mismo conjunto de partida se le denominará ley de composición u operación interna, de lo contrario, se llamará ley de composición u operación externa.

Estructuras algebraicas

Después de realizar el repaso histórico acotando las definiciones de las propiedades especificadas en la sección anterior, es necesario ampliar el conocimiento matemático involucrado directamente en el estudio de las estructuras algebraicas, los cuales ayudarán a la construcción del aplicativo.

De acuerdo con Laudecina (2009) “la estructura algebraica es una clasificación que depende de las propiedades que las operaciones cumplen en un conjunto” (p. 5). De esta definición, nos podemos percatar que el concepto de grupo es un caso específico de las estructuras, las cuales se definen por operaciones internas o externas de dichas estructuras. Sin embargo, para el desarrollo

de este aplicativo se hará énfasis en las operaciones internas o ley de composición interna, que según Fraleigh (1988) es “una regla que asigna a cada par ordenado de elementos de un conjunto A , algún elemento del conjunto” (p. 11). Esta operación puede cumplir algunas de las siguientes propiedades:

- **Asociativa:** $\forall a, b, c \in A$ tal que $(a * b) * c = a * (b * c)$.
- **Conmutativa:** $\forall a, b \in A$ tal que $a * b = b * a$.
- **Elemento neutro:** $\exists e \in A$ tal que $a * e = e * a = a, \forall a \in A$.
- **Inverso:** $\forall a \in A \exists a' \in A$ tal que $a * a' = a' * a = e$.
- **Distributiva de $*$ a la derecha e izquierda con respecto a $+$:** $\forall a, b, c \in A$ tal que $a * (b + c) = (a * b) + (a * c)$ y $(b + c) * a = (b * a) + (c * a)$.
- **Distributiva de $+$ a la derecha e izquierda con respecto a $*$:** $\forall a, b, c \in A$ tal que $a + (b * c) = (a + b) * (a + c)$ y $(b + c) * a = (b + a) * (c + a)$.

Estructuras algebraicas con una operación interna (+).

Se trata de un conjunto A con una operación $+$, que al contar con una sola operación este tipo de estructuras no cuentan con ninguna propiedad distributiva y solamente pueden contar con las cuatro primeras propiedades mencionadas previamente. Por lo tanto, se les puede denominar de las siguientes maneras de acuerdo con las propiedades que cumpla la operación:

Nombre	Asociativa	Elemento Neutro	Inverso	Conmutativa
Semigrupo	✘			
Monoide	✘	✘		
Grupo	✘	✘	✘	
Grupo conmutativo o abeliano	✘	✘	✘	✘

Tabla 2. Clasificación de las estructuras según sus propiedades

Estructuras algebraicas con dos operaciones internas (+, *).

Se trata de un conjunto A con las operaciones + y *, el cual se le podrá denominar de las siguientes maneras de acuerdo con las propiedades de las operaciones.

Nombre	Propiedades
Semianillo	Si * y + es semigrupo y + es distributiva con respecto a *
Semianillo conmutativo	Si * es un monoide conmutativo, + es un semigrupo y + es distributiva respecto a *
Anillo	Si * es un grupo abeliano, + es semigrupo y distributiva con respecto a *
Cuerpo o campo	Si * y + son un grupo abeliano y + es distributiva con respecto a *

Tabla 3. Clasificación de estructura con dos operaciones

Softwares

Teniendo como base los elementos matemáticos necesarios para la construcción del programa, ahondaremos en el aspecto tecnológico, empezando por la definición general de software y continuando cuando a este se le da un enfoque educativo. De acuerdo con lo mencionado por el Centro de Investigaciones y Soluciones Empresariales y Tecnológicas Ciset (s. f.) el software:

Da significado al soporte lógico de un sistema informático, es decir, es la parte no física que hace referencia a un programa o conjunto de programas de cómputo que incluye datos, reglas e instrucciones para poder comunicarse con el ordenador y que hacen posible su funcionamiento.

Partiendo de esta definición, precisamos que el concepto de software educativo es un conjunto de programas con sus diferentes características y añadiendo que este nace para facilitar el proceso de enseñanza y aprendizaje. Conforme a Gros (2014) nos muestra que algunos de los softwares se centran en la transmisión de un determinado contenido y otros en la adquisición de habilidades o estrategias. Sin embargo, hay que recalcar que no todo software que se implementa en el aula por los docentes tiene inicialmente una connotación educativa, tales como los programas encargados de entender ciertos lenguajes de programación (Pascal, P – Seint, Raptor, entre otros), pero que el docente los adecua para desarrollar habilidades en los estudiantes.

Ubicando el software en el contexto educativo, este se dotará de ciertas características particulares, las cuales acorde a Marqués (s. f.) son:

- **Utilizan el ordenador** como soporte en el que los alumnos realizan las actividades que ellos proponen.
- **Son interactivos**, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- **Individualizan el trabajo de los estudiantes**, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- **Son fáciles de usar**, pues los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.

Dentro del ámbito educativo, encontramos diferentes áreas del conocimiento como la biología, lenguaje, matemáticas, entre otras. En un principio, a los softwares educativos se les clasificaba de acuerdo con el contenido del área que manejaban, pero al surgir softwares que se

orientan a varias áreas o softwares no educativos que se utilizan en este contexto; conllevó a que se tuviera que desarrollar una categorización más amplia. Diferentes autores dan una clasificación para ello, aunque se tendrá en cuenta la dada por Gros (2014):

- **Instructivos:** Programas enfocados a los procesos de enseñanza y aprendizaje.
- **Acceso a la información:** Programas que permiten el acceso a bases de datos.
- **Creación:** Proporciona herramientas simples con las cuales se pueden construir elementos más complejos o entornos, generalmente son softwares de programación.
- **Desarrollo de estrategias:** Programas orientados en aspectos procedimentales.
- **Comunicación:** Herramientas enfocadas a la interacción de las personas, como las redes sociales.

En correspondencia a lo dicho, el software que elaboramos apoyará los procesos de enseñanza y aprendizaje de las propiedades de las estructuras algebraicas, concretamente en los procedimientos de comprobación de las propiedades. Así mismo, contrastando el enfoque que se le quiere dar al programa y las categorías planteadas por Gros, nuestro software *SEPPEA* se localizaría tanto en softwares instructivos como en desarrollo de estrategias, pues servirá para promover habilidades como la comprensión de cada elemento que interviene dentro de las propiedades estudiadas, puesto que el programa se encargará de realizar las operaciones.

Ventajas y desventajas de los softwares hacia la construcción del aplicativo

Siguiendo la idea de construir el aplicativo, nos percatamos en el curso *Programación en Matemáticas*, que hay diversas herramientas en las cuales se puede llevar a cabo la elaboración de este como lo son GeoGebra 5, MIT App Inventor 2, Matlab Online, HTML, Java, entre otros. De manera que, se quiere construir el aplicativo en alguno de estos softwares que cumpla las

características mencionadas en el trabajo y solucione los limitantes encontrados en los otros programas.

Como consecuencia de esto y del análisis de diferentes programas en los cuales se puede desarrollar este, escogimos seis softwares para estudiarlos a partir de sus manuales de programación. De esta manera, resumimos en la siguiente tabla las ventajas y desventajas de cada uno de ellos, con el fin de ir descartando aquellos que no se adecuan a las necesidades para el desarrollo del trabajo y seleccionar el software que utilizaremos para el mismo.

	Ventajas	Desventajas
JavaScript	<ul style="list-style-type: none"> - El acceso es fácil y maneja un lenguaje muy sencillo. - Es el único lenguaje que permite trabajar modo Full Stack en cualquier tipo de desarrollo de programación. - Muchos programas tienen una extensión en la cual se puede incluir código de JavaScript. - Permite cambiar la interfaz del sitio web, escribiendo pocas líneas de códigos. 	<ul style="list-style-type: none"> - Las ventanas presentan una limitación en el tamaño. - No se puede acceder desde la ventana a los documentos para leerlos o para cambiar su contenido. - No es compatible de manera uniforme con todos los navegadores. - Para poder diseñar un sitio web completo se debe usar con otros lenguajes de programación. - Algunas acciones se limitarían al permiso del realizador del programa.

HTML	<ul style="list-style-type: none"> - Su sencillez hace que pueda diseñarse y desplegarse un sitio web en muy poco tiempo. - No depende de la disponibilidad de una herramienta concreta para poder crear páginas web, será suficiente con un simple editor de texto. - Este lenguaje es el encargado de convertir un inocente archivo de texto inicial en una página web con diferentes tipos y tamaños de letra, con imágenes impactantes, animaciones sorprendentes, formularios interactivos, etc. 	<ul style="list-style-type: none"> - La interpretación de cada navegador puede ser diferente. - Está más orientado a la creación de páginas web. - Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección. - En algunos casos se tendrá la necesidad de trabajar con dos archivos, por un lado, en el que se edita y por otro el que da acceso a la página web, causando complicaciones al momento de editar.
GeoGebra 5	<ul style="list-style-type: none"> - Para trabajar en él, se puede descargar en diferentes dispositivos, tales como, computador, celular o Tablet. - Presenta múltiples maneras de programar siendo unas de ellas el lenguaje propio de GeoGebra, JavaScript global y la opción de Cálculo Simbólico (CAS). 	<ul style="list-style-type: none"> - Al exportar o importar un archivo de GeoGebra 5 a 6, se pueden generar algunos cambios no deseados en el aplicativo. - No cuenta con una herramienta de animación automática de objetos, lo que limita su potencial al momento de visualizar los objetos.

	<ul style="list-style-type: none"> - Mayor facilidad en el diseño, debido a las diferentes vistas gráficas que ofrece el software. - Cuenta con un sitio web de acceso libre, en el cual se pueden encontrar y publicar los aplicativos. - Es mucho más interactivo que dibujar algo en la pizarra, lo cual fomenta la atención de los estudiantes, y se pueden dar más variaciones en las cuales ellos tengan mayor curiosidad. 	
MatLab Online	<ul style="list-style-type: none"> - MATLAB Apps son aplicaciones interactivas que combinan acceso directo a grandes colecciones de algoritmos con retroalimentación visual inmediata. - Con adecuadas optimizaciones, son rápidos en ejecución y de alta precisión. - Hay foros en la web donde se comparten experiencias como soporte e ilustración a los usuarios. 	<ul style="list-style-type: none"> - Requiere de una licencia para su uso. - Al tener el software instalado, necesita de una extensión, para que los archivos funcionen correctamente. - Para programar es necesario tener conocimiento del lenguaje programación BASIC, el cual es un lenguaje de alto nivel.

<p>MIT App Inventor 2</p>	<ul style="list-style-type: none"> - Permite crear aplicaciones por medio de bloques de manera intuitiva y gráfica, sin necesidad de saber código de programación. - No es necesario tener conocimientos extensos sobre códigos de programación. - Es de fácil acceso puesto que únicamente se debe tener una cuenta Gmail. 	<ul style="list-style-type: none"> - Es una herramienta para desarrollo en línea por lo que requiere conexión permanente a Internet. - No aporta demasiada flexibilidad para realizar futuros cambios como lo hace por ejemplo el código Java. - Es un entorno de desarrollo de aplicaciones exclusivamente para Android.
<p>Python</p>	<ul style="list-style-type: none"> - Es de código abierto y gratuito. - Es compatible con todos los sistemas operativos, puesto que el código se escribe una sola vez y lo ejecutará en todos los sistemas. - Es fácil de leer, escribir y aprender. 	<ul style="list-style-type: none"> - Consumo alto de memoria del dispositivo para ejecutar el programa. - Se puede volver lento el programa, dependiendo del código usado. - A causa del alto consumo de memoria, es poco viable su ejecución y desarrollo orientado hacia dispositivos móviles.

Tabla 4. Ventajas y desventajas de softwares de programación

A partir de lo anterior, JavaScript no se tendrá en cuenta debido a que no es compatible de manera uniforme con todos los navegadores y no se pueden modificar la programación del documento al mismo tiempo que se está ejecutando. En cuanto a HTML, se excluirá porque está

enfocado al desarrollo de páginas web y no de aplicativos, además de que se debe trabajar con dos archivos, uno en el que se edita y otro que da acceso a la página web. En relación con MatLab Online, no nos favorece que se maneje por medio de una licencia y que se deba instalar una extensión para la ejecución de los archivos. De igual manera, MIT App Inventor 2 dificulta el proceso por la necesidad de tener conexión a internet permanentemente y su exclusividad solamente para dispositivos Android. Acerca de Python, este no se utilizará para el desarrollo del programa, dado el alto consumo de memoria del dispositivo al ejecutar el programa y lo que genera que este se vuelva lento, entendiéndolo robusta y compleja que es la programación de las propiedades de las estructuras algebraicas.

Dicho esto, GeoGebra 5 es el software más viable desde nuestro análisis para el desarrollo del aplicativo, porque integra varias vías de programación, como lo pueden ser el CAS, la hoja de Cálculo o JavaScript. También, es favorable porque permite manejar múltiples vistas gráficas haciendo que el contenido esté más organizado y se pueda seleccionar el contenido en cada vista gráfica, de tal manera que la ejecución del programa sea intuitiva para el usuario.

CAPÍTULO III. APLICATIVO SEPPEA

Generalidades de GeoGebra

Teniendo en cuenta lo mencionado en la última parte de la sección anterior, GeoGebra 5 es la opción más adecuada y completa para el desarrollo del software que llevará como nombre *Software Educativo Para las Propiedades de las Estructuras Algebraicas (SEPPEA)*. Por ello, es importante hacer una introducción de manera breve y precisa de algunas funcionalidades y herramientas de GeoGebra 5.

Las diversas vistas y apariencias que ofrece el software para los diferentes objetos matemáticos permiten que el usuario use el software de acuerdo con la finalidad o intención del contenido matemático que se está trabajando.

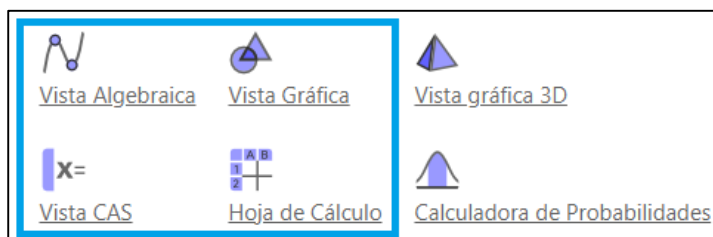


Ilustración 4. Vistas de GeoGebra

Particularmente, para el desarrollo del software se encontrarán involucradas cuatro de las seis vistas del software, siendo estas las que están delimitadas por el recuadro de la Ilustración 4. En ese sentido, se abordarán algunas generalidades o características de las vistas en cuestión, relacionándolas directamente con su funcionalidad dentro de la construcción del aplicativo.

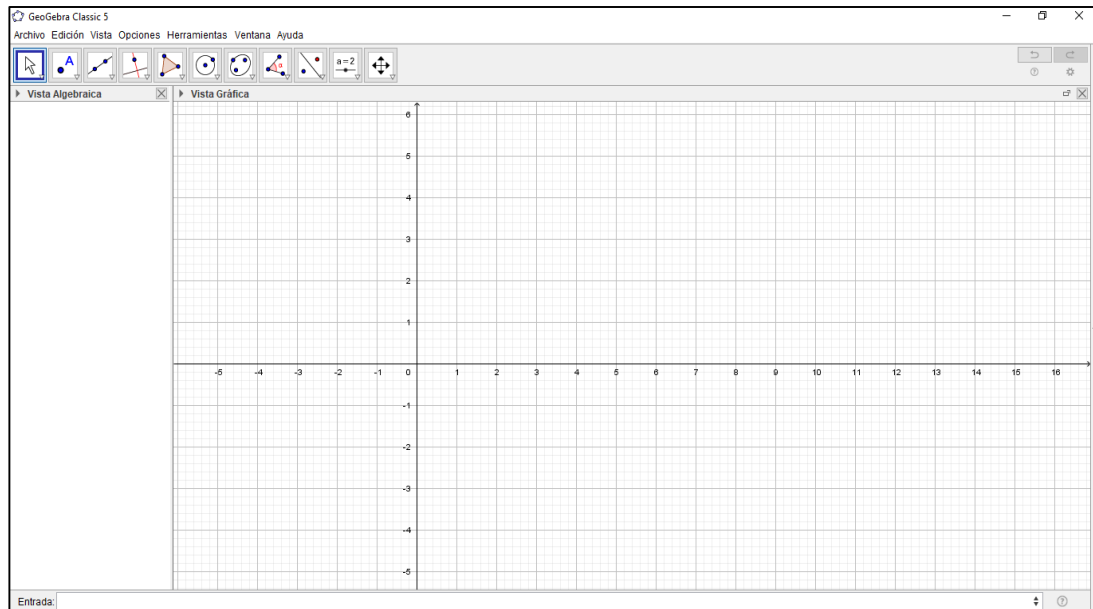


Ilustración 5. Vista principal de GeoGebra

a) Vista Algebraica: Esta vista es la única que no posee una barra de tareas, sin embargo, para interactuar con esta vista se puede hacer por medio de la barra de entrada que se encuentra ubicada en la parte inferior de la ventana principal de GeoGebra (Ver Ilustración 5). Adicionalmente, en esta vista se pueden encontrar todos aquellos objetos que se construyan en las otras vistas que se pueden representar tanto algebraica o aritméticamente, siendo esto la segunda forma de interactuar con esta vista. Un ejemplo de lo último mencionado es cuando en la vista gráfica construimos el punto en el plano cartesiano, el cual será representado en la vista algebraica con las coordenadas cartesianas que se le asigne el usuario, un deslizador o una función (Ver Ilustración 6 – Ilustración 7).

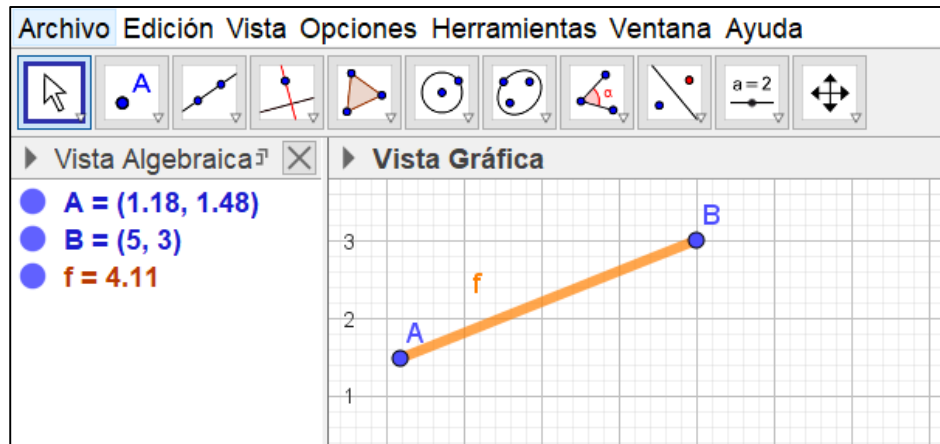


Ilustración 6. GeoGebra representación segmento

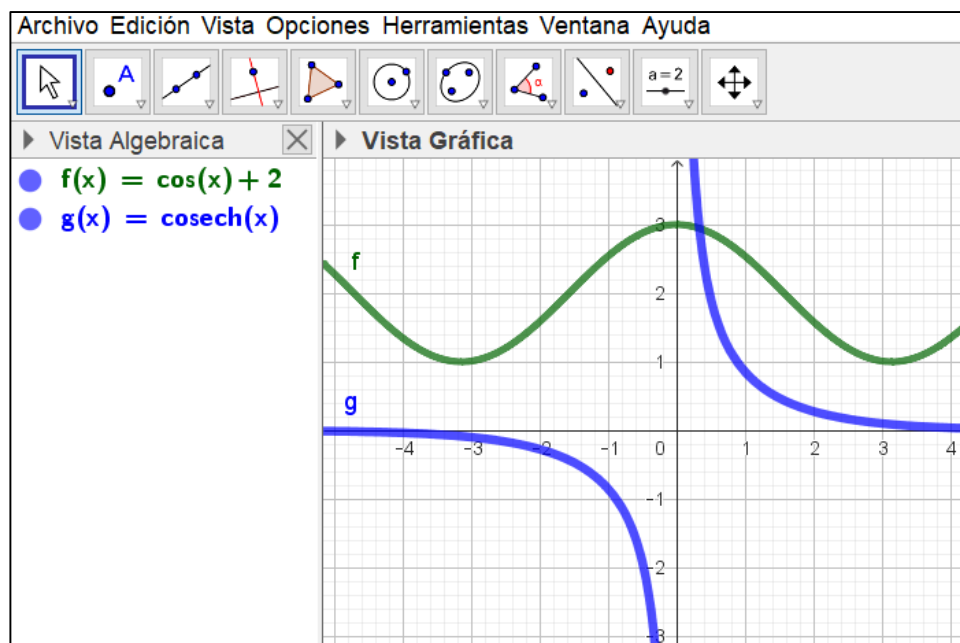


Ilustración 7. GeoGebra representación funciones

b) Vista Gráfica 2D: Esta vista cuenta con su barra de tareas (Ver Ilustración 8) en la cual podemos encontrar tres tipos de herramientas o funciones, las primeras son las herramientas geométricas que abarcan casi toda la barra, tales como la construcción de puntos, rectas, segmentos, ángulos, intersecciones, rectas paralelas, rectas tangentes. Las segundas funciones permiten interactuar con el plano cartesiano en la medida de poder mover,

seleccionar objetos, acercar o alejar el plano cartesiano. Las terceras herramientas dan la posibilidad de crear deslizadores, textos, imágenes, botones, casillas de control y entrada, siendo estas las que más se utilizarán para el desarrollo del aplicativo, pues por medio de las mismas es posible crear objetos condicionados, es decir que en el momento en que el usuario interactúe con estos, el programa realizará las tareas que nosotros le designamos, sin necesidad de modificar todo el programa.

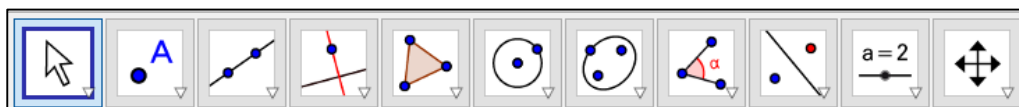


Ilustración 8. Barra de herramientas vista gráfica 2D

Adicionalmente, en esta vista se podrán visualizar sobre todo aquellos objetos que tengan una representación geométrica en el plano cartesiano o todo objeto con el cual se pueda interactuar como lo son los botones, entradas de casillas, deslizadores, textos, entre otros, al ser las que principalmente se utilicen más adelante y que se explicarán un poco más a fondo.

c) **Vista CAS:** Esta vista se puede considerar como la principal para programar, puesto que allí se pueden realizar comandos que pueden contener variables numéricas o algebraicas, que pueden indicar diferentes operaciones y algoritmos. Consiste en un grupo de celdas que tiene una línea de entrada donde se indican las instrucciones y en la parte inferior, una línea de salida que mostrará el resultado a la instrucción establecida. Además, posee su propia barra de herramientas (Ver Ilustración 9) con la cual se pueden evaluar las expresiones ingresadas y realizar cálculos. No obstante, esta barra de herramientas queda en un segundo plano, ya que los comandos son lo más importante de la vista y no se encuentran en el misma, sino que se pueden escribir directamente en las líneas de entradas o también en la

opción de ayuda donde se encuentran todos los comandos. Cabe aclarar que la opción de ayuda está disponible para todas las vistas, pero es en el CAS que se ve su mayor potencial y utilidad.

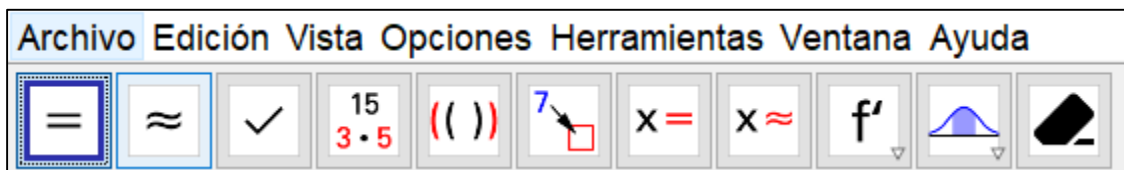


Ilustración 9. Barra de tareas vista CAS

El *Computer Algebra System* será una de las vistas más importantes para la construcción del aplicativo, debido a que en este se desarrollarán todas las líneas de programación que sustentarán el programa. Cuenta con su propio lenguaje de programación, el cual facilita el cálculo simbólico y adicionalmente, al ser parte de un software matemático se cuenta con comandos netamente matemáticos que nos ayudarán en algunos algoritmos. Por ejemplo, operaciones entre matrices tales como aplanar o transponer, operaciones entre listas, comandos lógicos relacionados con la validez de las instrucciones, entre otros (Ver Ilustración 10). En el Anexo 1 se podrá encontrar una tabla que especifica algunos comandos utilizados en el programa para interés del usuario.

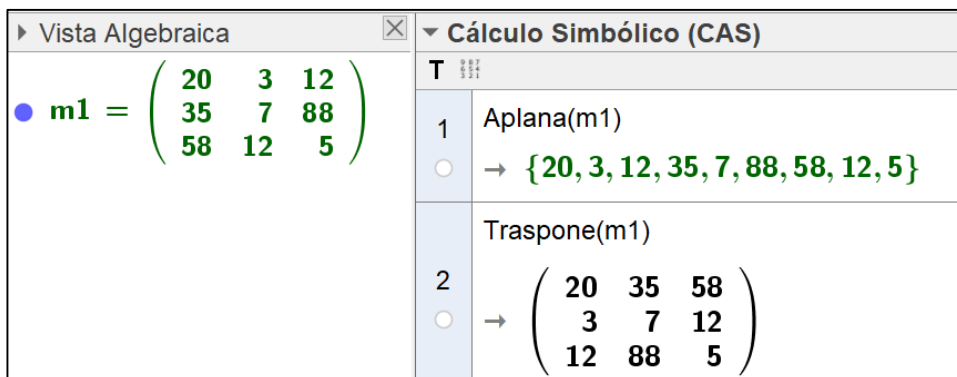


Ilustración 10. Ejemplo comandos vista CAS

d) Hoja de Cálculo: Esta vista contiene en su barra de tareas algunas herramientas que permiten trabajar con matrices, listas y datos estadísticos, siendo posible observar algunos objetos construidos en la vista gráfica, algebraica y CAS tales como listados de números o matrices y en sentido contrario (Ver Ilustración 11). Al mismo tiempo es posible utilizar funciones o comandos para realizar procesos con los datos que se encuentran en las casillas de la hoja, tales como las operaciones con los números y algunos comandos que son utilizados en las demás vistas gráficas.



Ilustración 11. Barra de herramientas Hoja de Cálculo

Primera versión del programa

En un primer momento, nuestra finalidad con la programación de las propiedades sería la manera en que el usuario ingresa los valores de los resultados de las dos operaciones en las Tablas de Cayley, puesto que estos son la base principal para que el programa tenga funcionamiento. La manera en que conseguimos facilitar este asunto fue por medio de la vista gráfica *Hoja de Cálculo* (Ver Ilustración 12), en donde los usuarios pueden modificar los valores de la estructura algebraica que estén estudiando o investigando, siempre y cuando cumpla la condición de que no cuente con menos de tres elementos o más de nueve, tomando esta decisión por funcionalidad del programa.

Hoja de Cálculo											
	A	B	C	D	E	F	G	H	I	J	K
1	11	0	1	2	3	4	5	6	7	8	9
2	0	0	1	2	3	5	5	6	0	0	0
3	1	1	0	0	2	7	6	10	0	0	0
4	2	2	0	0	0	6	10	11	0	0	0
5	3	3	2	0	0	10	11	12	0	0	0
6	4	4	5	6	10	11	12	13	0	0	0
7	5	5	6	10	11	12	13	14	0	0	0
8	6	6	10	11	12	13	14	15	0	0	0
9	7	0	0	0	0	13	0	0	0	0	0
10	8	0	0	0	0	0	0	0	0	0	0
11	9	0	0	0	0	0	0	0	0	0	0

Ilustración 12. Tablas de Cayley Hoja de Cálculo

Posterior a esto, fue necesario asociar las tablas presentes en la hoja de cálculo con una matriz para cada una de ellas (Ver Ilustración 13), de tal manera que pudiéramos trabajar con estas desde la vista CAS, gracias a que las matrices se veían reflejadas en la vista algebraica y de esta manera se vuelven objetos que pueden ser utilizados en la vista CAS. Al contar con las matrices en la vista CAS, esto nos permitía empezar a programar las líneas de instrucciones para cada una de las propiedades de las estructuras algebraicas. Antes de abordar la construcción de cada una de las propiedades, es necesario mencionar que al no contar con la definición de suma y multiplicación con la que el usuario trabaja en la estructura algebraica, se determinó trabajar las operaciones por medio de proposiciones lógicas obteniendo un valor verdadero o falso sobre cada una de ellas.

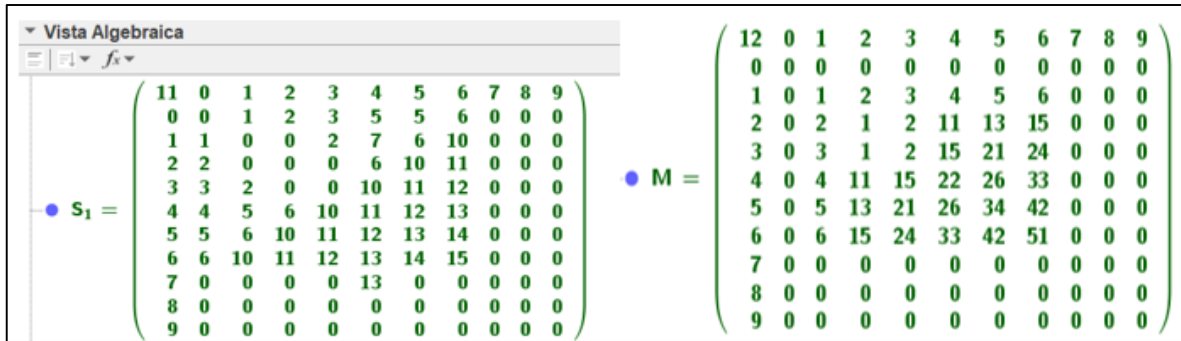


Ilustración 13. Matrices asociadas a las tablas

Al contar con las tablas en la vista CAS, el siguiente paso fue depurarlas con la finalidad de obtener las tablas con la cantidad de clases con las que el usuario estuviera trabajando (para practicidad de la explicación se tomará $n = 3$), puesto que las matrices S_1 y M nos muestran tablas con 10 clases. Por esta razón se crea una casilla de entrada en la cual el usuario especificará la cantidad de elementos o clases con las que cuenta la estructura algebraica, a la cual se le asignó la letra n y será aquella que condicionarán la gran mayoría de las líneas de comando que se encuentran en el aplicativo.

En este orden de ideas, conociendo la cantidad de clases de la estructura algebraica realizamos el siguiente algoritmo para obtener la tabla de Cayley del usuario:

1. Debemos eliminar las filas adicionales con las que cuenta la matriz y esto se hará por medio del comando *Extraer*, pues este tiene el propósito de tomar una matriz de la cual se extraerá una matriz que contará con la misma cantidad de columnas, pero no de filas. En este caso las especificaciones que nos pide el comando $Extrae(S_1, 1, n + 1)$, indica primero la matriz que será editada y segundo, desde que fila inicial hasta que fila final queremos dejar en la nueva matriz. De esta manera obtenemos la matriz m_1 (Ver Ilustración 14):

$$\textcircled{\bullet} \mathbf{m1} = \begin{pmatrix} 11 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 0 & 1 & 2 & 3 & 5 & 5 & 6 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 2 & 7 & 6 & 10 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 6 & 10 & 11 & 0 & 0 & 0 \end{pmatrix}$$

Ilustración 14. Matriz $m1$

2. Como el comando *extrae* solamente nos permite eliminar filas, lo que debemos hacer para eliminar el restante de columnas es obtener la traspuesta de la matriz m_1 y esto se hace por medio del comando *Traspone*. Este comando *Traspone*(m_1) solamente pide colocar la matriz a la cual se le realizará la operación y de esta forma conseguimos la matriz m_2 :

$$\textcircled{\bullet} \mathbf{m2} = \begin{pmatrix} 11 & 0 & 1 & 2 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 3 & 3 & 2 & 0 \\ 4 & 5 & 7 & 6 \\ 5 & 5 & 6 & 10 \\ 6 & 6 & 10 & 11 \\ 7 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 \end{pmatrix}$$

Ilustración 15. Matriz $m2$

3. Al contar con la matriz m_2 la cual es la traspuesta de m_1 , quiere decir que en m_2 se colocarán las filas de m_1 como columna, respetando su orden (la fila 1 de la matriz original será la columna 1 de la matriz nueva). Simplemente se extraerán las filas sobrantes con el comando *Extrae*($S_2, 1, n + 1$), consiguiendo la matriz m_3 .

$$m_3 = \begin{pmatrix} 11 & 0 & 1 & 2 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 \end{pmatrix}$$

Ilustración 16. Matriz m_3

4. A continuación, se transpone la matriz m_3 con el comando $Traspone(m_3)$, obteniendo la matriz m_4 en la cual se tendrán las filas y columnas ordenadas como originalmente el usuario las ingresó con sus respectivos valores.

$$m_4 = \begin{pmatrix} 11 & 0 & 1 & 2 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 \end{pmatrix}$$

Ilustración 17. Matriz final

Finalmente, la matriz m_4 será con la que se trabaja en el proceso de programación para la primera operación de la estructura algebraica, puesto que se construyó de manera que siempre tendrá el tamaño de la cantidad de clases que el usuario ingrese al programa y con el mismo proceso explicado anteriormente se obtendrá la matriz para la segunda operación de la estructura algebraica que se esté trabajando.

a) Propiedad conmutativa para la operación +:

Para comprobar la propiedad conmutativa realizamos el siguiente procedimiento. Primero, creamos una secuencia o ciclo con el comando 1 ($a:=\{\text{Secuencia}(\text{Secuencia}(\text{Elemento}(m_4,i,j))=\text{Elemento}(m_4,j,i),i,2,n+1,1),j,2,n+1,1)\}$), el cual tiene como propósito comparar cada uno de los valores de la matriz con su respectivo valor inverso o, en otras palabras, estamos comprobando si al sumar $2 + 1$ es igual que sumar $1 + 2$ en esa estructura algebraica. Por

ejemplo, se tomará el valor en la posición fila 3 columna 2 y se mirará si tiene el mismo valor con el que se encuentra en la posición fila 2 y columna 3 y así, con cada uno de los valores obteniendo una matriz (Ver Ilustración 18), la cual indicará si esto es verdadero o falso.

```
a:={Secuencia(Secuencia( Elemento(m4, i, j)==Elemento(m4, j, i), i, 2, n+1, 1), j, 2, n+1, 1)}
→ a := { ( true true true )
          ( true true true )
          ( true true true ) }
```

Ilustración 18. Comando 1

Seguido a esto, al obtener esta matriz la aplanamos, es decir que colocamos todos los valores de la matriz en una lista respetando el orden de las filas de la matriz para posteriormente sumar cada valor de la lista creada, ya que GeoGebra asocia al verdadero un 1 y al falso un 0, con la intención de concluir si los valores que componen la lista son iguales o no a n^2 para llegar a la conmutatividad (Ver Ilustración 10 – Ilustración 20). Al obtener este valor, se usará como una condición para que el programa muestre los textos indicando que cumple la propiedad, si no, es decir que no sea $2 \leq n < 10$, aparecerá que no cumple la propiedad conmutativa.

```
a1:=Aplana(a)
→ a1 := {true, true, true, true}
```

Ilustración 19. Comando 3

```
a2:=Suma(a1)
→ a2 := 4
```

Ilustración 20. Comando 4

b) Propiedad asociativa para la operación +:

Se sabe que para comprobar tal propiedad se necesitan de valores a, b y c que satisfacen la igualdad $a + (b + c) = (a + b) + c$. En tal sentido, validamos a través del comando 1 dicha propiedad, en donde se consideran las variables i, j y h (Ver Ilustración 21). Es importante diferenciar que, i, j y h en el comando varían en filas y columnas, lo que nos permite no realizar la operación de la estructura definida, sino comparar resultados dependiendo de la posición de la matriz. Por eso, es necesario establecer las condiciones de cada una de las variables, las cuales son el valor inicial, el valor final y el incremento, puesto que se está trabajando con la matriz m_4 (Ver Ilustración 17), entonces no se debe tener en cuenta la columna y fila 1 que son posiciones que no interesan considerar para la comprobación de la misma. Por esto, los valores iniciales de las variables i y j empiezan en 2 y el valor final va hasta $n + 1$, dado que la matriz tiene una fila y columna adicionales.

Así, el primer propósito del comando es ir a la posición en la matriz del resultado que se obtiene al sumar los paréntesis de la propiedad asociativa y luego, se desplazan a la posición del valor resultante al sumarle el valor de a o c como lo establece la propiedad, para comparar si los valores en estas posiciones son iguales (verdaderos) o diferentes (falsos).

Finalmente, con los resultados anteriores, se contabiliza la cantidad de verdaderos de tal manera que satisfagan la propiedad, pues para cuando son 3 clases deben resultar 36 valores verdaderos, o de manera general debe cumplirse que $n^2(n + 1)$.

```
a3:={Secuencia(Secuencia(Secuencia( Elemento(m4, i, j-h)==Elemento(m4, j-h, i), i, 2, n+1, 1), j, 2, n+1, 1),h,0,n,1)}  
→ a3 := { { ( true true ), ( true true ), ( true true ) } }
```

Ilustración 21. Comando 1

c) Propiedad del elemento neutro para la operación +:

El procedimiento para hallar el elemento neutro tiene dos pasos principales. El primero es determinar si la estructura posee esta propiedad, y el otro, si cumple dicha propiedad entonces cuál elemento es el neutro dentro de la estructura, sabiendo que este elemento deja una fila y una columna exactamente idénticas que son las que siempre tiene la matriz. Para el primer paso, son los comandos desde el 7 hasta el 14 de la vista CAS, los cuales consisten en:

- I) Tomar de la matriz m_4 una lista de las filas, respetando el orden (Ver Ilustración 22).

```
k:=Secuencia(Extrae(m6, i,i),i,1,n+1,1)
→ k := {( 0 1 ),( 0 1 ),( 1 0 )}
```

Ilustración 22. Comando 7

- II) Ahora, comparar el primer elemento de esa lista con los otros, con el fin de encontrar cuál fila es idéntica a ese elemento (Ver Ilustración 23).

```
k1:=Secuencia(Elemento(k,1)==Elemento(k,i),i,2,n+1,1)
→ k1 := {true, false}
```

Ilustración 23. Comando 8

Análogamente, se hace el mismo procedimiento con las columnas, de tal manera que el resultado obtenido para filas y columnas nos indica la posición en la cual se ubica el elemento neutro de la estructura dentro de la matriz m_4 (Ver Ilustración 24).

```
k20:=Elemento(m4, Elemento( k_14,1), Elemento(k_11, 1))
→ k20 := 0
```

Ilustración 24. Comando 22

d) Propiedad del elemento inverso para la operación +:

El proceso para comprobar que existen elementos inversos y poder al mismo tiempo encontrar el elemento inverso correspondiente a cada uno de los elementos de la estructura algebraica del usuario, consiste en primero buscar en la matriz aquellos elementos que sean iguales al valor correspondiente al neutro de la estructura (Ver Ilustración 25). Posteriormente debido a que GeoGebra nos muestra una matriz con valores lógicos (true o false), la modificaremos de manera que se les asignan valores numéricos a estas afirmaciones respectivamente $true = 1$ y $false = 0$ (Ver Ilustración 26), realizando este paso para que para la programación sea más sencilla de trabajar con números que varían en los elementos de las matrices o listas.

$$l := \text{Secuencia}(\text{Secuencia}(\text{Elemento}(m3, i, j) == N, i, 1, n+1, 1), j, 2, n+1, 1)$$
$$\rightarrow l := \begin{pmatrix} \text{true} & \text{true} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{true} & \text{true} \\ \text{false} & \text{false} & \text{true} & \text{true} \end{pmatrix}$$

Ilustración 25. Comando 27

$$a_3 := \text{Secuencia}(\text{Suma}(\text{Elemento}(a_2, i)), i, 1, n+1, 1)$$
$$\rightarrow a_3 := \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Ilustración 26. Comando 34

Después, tomamos la matriz a_4 y con esta construiremos otra (Ver Ilustración 27 – matriz derecha) que corresponde a las posiciones en las cuales no se encuentre el número 1, ya que está colocará el valor de $n + 1$ y en los lugares en los que se encuentre el valor 1 pondrá el valor que corresponda que se encuentre en la fila uno de la matriz que ingresó el usuario. Por ejemplo, como

en la matriz a_4 tenemos un uno en la posición fila 3 columna 3 y ese valor se encuentra ubicado en la fila del elemento dos, entonces en la matriz nueva colocará el valor dos.



Ilustración 27. Matrices

Al contar con la matriz a_6 la ubicamos en una lista respetando el orden de la matriz, obteniendo una lista (Ver Ilustración 28), de la cual extraemos los números 4 y así tenemos las posiciones de las filas en las cuales se encuentran ubicados los inversos. Ahora, debemos encontrar los valores de las columnas respectivamente a los valores de las filas que ya encontramos anteriormente en las cuales se encuentran los inversos y esto lo conseguimos a través del comando 49 (Ver Ilustración 29).

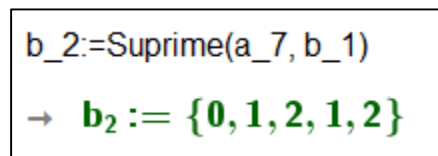


Ilustración 28. Comando 42

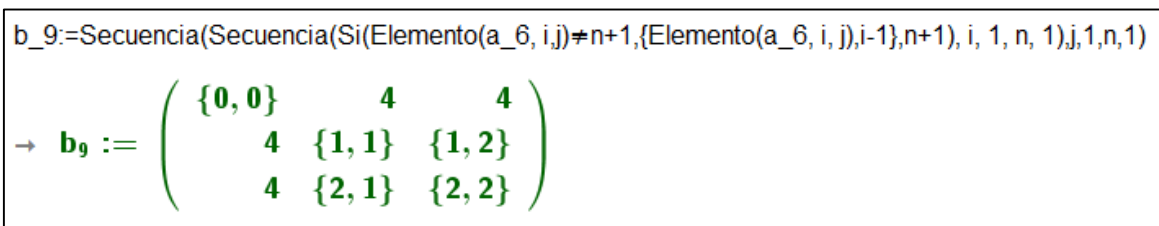


Ilustración 29. Comando 49

En el comando 49, traemos nuevamente los valores de las filas en las cuales se encuentran ubicados los elementos inversos, pero añadiendo el valor de la columna en la cual se encuentra este, por medio de la variable i que varía de 1 hasta n y, como esa variable comienza un valor después de los elementos de la estructura únicamente le restamos uno de manera que nos resultará el valor de la columna que estamos buscando.

En seguida, convertimos la matriz b_9 en una lista de la que se eliminan los números 4. Sin embargo, GeoGebra no nos permitió que dentro de la lista obtenida se agruparan las parejas de números dentro de los corchetes (Ver Ilustración 30) como se observa en la matriz del comando 49.

$L_2 := \text{Suprime}(b_{10}, L_1)$ $\rightarrow L_2 := \{0, 0, 1, 1, 1, 2, 2, 1, 2, 2\}$

Ilustración 30. Comando 52

Por lo tanto, la solución que planteamos a este inconveniente fue extraer de la lista L_2 los valores correspondientes a las filas en una lista y en otra los valores de las columnas (Ver Ilustración 31), fijándonos que los valores de las filas se encontraban ubicadas en las posiciones impares y los de las columnas en las pares.

$L_3 := \text{Secuencia}(\text{Elemento}(L_2, i), i, 1, 2v_2, 2)$ $\rightarrow L_3 := \{0, 1, 1, 2, 2\}$
$L_4 := \text{Secuencia}(\text{Elemento}(L_2, i), i, 2, 2v_2, 2)$ $\rightarrow L_4 := \{0, 1, 2, 1, 2\}$

Ilustración 31. Comando 53 y 54

Al contar con estas dos listas sabemos que el valor de la lista L_4 en la posición uno es el inverso del valor en la posición uno de la lista L_3 y así respectivamente con cada valor de las dos listas, solo si comprobamos que se cumpla que $a + b = b + a = e$, siendo b el elemento inverso de a y, e el neutro de la estructura algebraica. Para esto le sumamos a cada elemento de las listas el valor de 2 (Ver Ilustración 32), consiguiendo las posiciones de cada elemento dentro de la matriz y así, por medio del comando 57 (Ver Ilustración 33) verificamos que $a + b = b + a = e$ se cumpla para cada elemento.

```
L_5:=Secuencia(Elemento(L_3, i)+2, i, 1, v_2, 1)
→ L5 := {2, 3, 3, 4, 4}
L_6:=Secuencia(Elemento(L_4, i)+2, i, 1, v_2, 1)
→ L6 := {2, 3, 4, 3, 4}
```

Ilustración 32. Comando 55 y 56

```
L_7:=Secuencia(Elemento(m3, Elemento(L_5, i), Elemento(L_6, i))=Elemento(m3, Elemento(L_6, i), Elemento(L_5, i)), i, 1, v_2, 1)
→ L7 := {true, true, true, true, true}
```

Ilustración 33. Comando 57

Finalmente, creamos la lista L_{11} utilizando las listas L_4 y L_3 en las cuales tenemos agrupados los elementos con su respectivo inverso (Ver Ilustración 34) y estos serán los que el usuario visualizará en la vista principal del programa en GeoGebra.

```
L_11:=Suprime(L_10, L)
→ L11 := {(1, 2), (2, 2), (1, 1), (2, 1), (0, 0)}
```

Ilustración 34. Comando 62

Habiendo realizado los procesos anteriores, procedemos a comparar los valores de las listas respetando su posición dentro de las mismas y así comprobar si todos son iguales, lo que implica que la propiedad se cumple. Contando con las cuatro propiedades estructuradas para la operación suma, procedimos a realizar un razonamiento análogo para la programación de las propiedades de la operación asterisco u operación 12 como la nombramos en el programa, las cuales tienen como matriz principal la m_{10} .

e) Propiedad distributiva de * con respecto a +.

Para la comprobación de esta propiedad, iniciamos obteniendo los resultados de la parte izquierda de la igualdad $a * (b + c) = (a * b) + (a * c)$, dado que en la parte izquierda tenemos dentro del paréntesis $b + c$, que hace referencia a los valores que se encuentran dentro de la matriz de la suma, así que tomamos estos valores y le sumamos dos a cada valor de la matriz (Ver Ilustración 35), obtendremos las posiciones de tales valores en la matriz de la operación asterisco. Por ejemplo, el valor número 2 está en la posición cuatro de la fila y columna uno y de manera análoga, para cada valor de los elementos.

```
P_2:=Secuencia(Elemento(P_1+2, i), i, 1, n^2, 1)
→ P2 := {2, 3, 4, 3, 2, 2, 4, 2, 2}
```

Ilustración 35. Comando 116

Posterior a esto, debemos tomar estos resultados y realizar la operación asterisco por a , quien varía desde 0 hasta $n - 1$. Al no conocer como está definida tal operación, nuevamente le sumamos dos a los valores que puede tomar a (Ver Ilustración 36) y por medio del comando P_9 , lo que hacemos es ir a la matriz de la operación asterisco y ubicar todos los resultados de cada uno de los valores de la operación tomando como referencia que la lista para las filas es el comando P_8 , y para las columnas el P_2 .

```
P_8:=Secuencia(Elemento(P_7, i)+2, i, 1, n, 1)
→ P8 := {2, 3, 4}
```

Ilustración 36. Comando 121

Con lo anterior llegamos a los resultados de la parte izquierda de la propiedad distributiva, por lo que estos se deben comparar con la parte derecha de la misma. Ahora, como en la parte derecha de la igualdad se tienen dos paréntesis, hacemos primero la operación de un paréntesis (Ver Ilustración 37) y luego, el del otro paréntesis. En tal sentido, se toman los resultados del primer paréntesis como posiciones y los resultados del segundo paréntesis igualmente, obteniendo las posiciones en la matriz de la operación suma m_4 (Ver Ilustración 38).

```
P_17:=Secuencia(Elemento(P_12, i)+2, i, 1, n^2, 1)
→ P17 := {2, 2, 2, 2, 3, 4, 2, 4, 3}
```

Ilustración 37. Comando 129

```
P_25:=Secuencia(Elemento(m4, Elemento(P_17, Elemento(P_15, i)), Elemento(P_24, i)), i, 1, n^3, 1)
→ P25 := {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 2, 0, 0, 0, 2, 1, 2, 0, 0, 1, 0, 0}
```

Ilustración 38. Comando 136

De esta manera, completamos la parte fundamental del programa siendo esta la programación de las propiedades de las dos operaciones y la propiedad distributiva de la operación * respecto a la +, lo que posibilita a centrarnos en la interfaz que verá el usuario (estudiante), pues allí se muestran las matrices en donde ingresarán los datos para las estructuras, la verificación de las propiedades ya mencionadas y los tipos de estructura algebraica según el cumplimiento y validación de las anteriores.

Haciendo énfasis en un elemento de la interfaz como son los textos, debemos recalcar que cada uno de ellos están condicionados por varias variables del programa, aunque hay algunas variables que están sujetas a cada uno de los textos, y otras que, si son particulares de cada uno, por lo que los textos se hacen visibles en el programa acorde a las condiciones establecidas para cada propiedad y tipo de estructura.

Teniendo presente que uno de nuestros principales objetivos, es la accesibilidad al programa para los usuarios, entonces cargamos el Applet en uno de nuestros perfiles de GeoGebra Recursos, donde se puede publicar material y los estudiantes o población interesada en ello, pueda interactuar fácilmente con el programa e incluso descargarlo para manejarlo desde el escritorio de su propio ordenador (<https://www.geogebra.org/m/rmjdgrj5>).

Un ejemplo del funcionamiento del programa y cómo se ven los elementos mencionados anteriormente, se puede ver en la Ilustración 39, en donde se observa una estructura algebraica de 3×3 .

Operaciones internas

Número de clases = $2 < n < 10$

La operación 11 es conmutativa
La operación 11 es asociativa
El elemento neutro de la operación 11 es 0
Los inversos de 11 son $\{(1,2), (2,2), (1,1), (2,1), (0,0)\}$

La operación 12 es conmutativa
La operación 12 es asociativa
El elemento neutro de la operación 12 es 1
Los inversos de 12 son $\{(2,2), (1,1)\}$

La operación 12 es distributiva respecto a 11

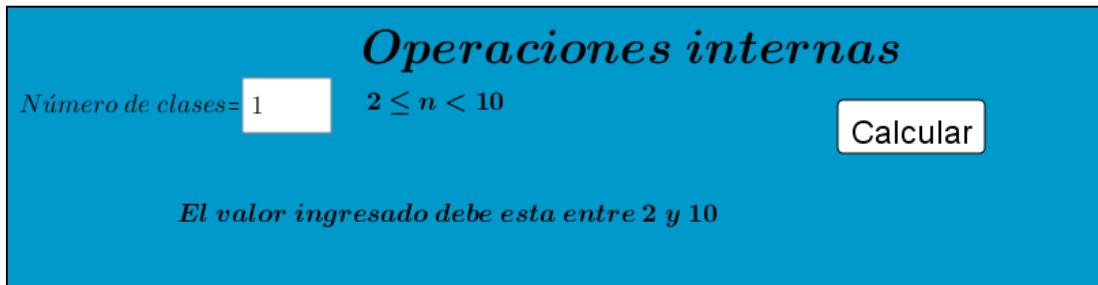
La 11 es un Grupo Abeliiano
La 11 y la 12 son un Grupo Abeliiano entonces son un Campo o Cuerpo

	A	B	C	D	E	F	G	H	I	J	K
1	11	0	1	2	3	4	5	6	7	8	9
2	0	0	1	2	3	5	5	6	0	0	0
3	1	1	0	0	2	7	6	10	0	0	0
4	2	2	0	0	0	6	10	11	0	0	0
5	3	3	2	0	0	10	11	12	0	0	0
6	4	4	5	6	10	11	12	13	0	0	0
7	5	5	6	10	11	12	13	14	0	0	0
8	6	6	10	11	12	13	14	15	0	0	0
9	7	0	0	0	0	13	0	0	0	0	0
10	8	0	0	0	0	0	0	0	0	0	0
11	9	0	0	0	0	0	0	0	0	0	0
12											
13	12	0	1	2	3	4	5	6	7	8	9
14	0	0	0	0	0	0	0	0	0	0	0
15	1	0	1	2	3	4	5	6	0	0	0
16	2	0	2	1	2	11	13	15	0	0	0
17	3	0	3	1	2	15	21	24	0	0	0
18	4	0	4	11	15	22	26	33	0	0	0
19	5	0	5	13	21	26	34	42	0	0	0

Ilustración 39. Ejemplo 3×3

La estructura con la cual presentamos el ejemplo se ingresa en las tablas de Cayley, teniendo en cuenta que el número de clases en este caso es de 3. En el caso en el que el usuario ingrese un

número de clase menor a 2 y mayor a 9, el programa le indicará que necesariamente debe ser un número entre este intervalo (Ver Ilustración 40), lo cual condiciona a todos los textos para que no se muestren si no cumplen que $2 \leq n < 10$.



Operaciones internas

Número de clases= 1 $2 \leq n < 10$ Calcular

El valor ingresado debe esta entre 2 y 10

Ilustración 40. Número de clase igual a 1

Al mismo tiempo, cuando se ingresan los valores de las tablas de Cayley y se hace uso del botón denominado “Calcular”, el programa verificará cada una de las propiedades y determinará cuáles de ellas cumple, con la intención de evidenciar qué tipo de estructura algebraica es la que se conforma.

En tal sentido, se puede ver que ambas operaciones cumplen las cuatro propiedades necesarias para ser un Grupo abeliano y adicionalmente, la operación 12 es distributiva con respecto a la operación 11, de tal manera que las operaciones forman la estructura algebraica denominada Cuerpo o Campo.

Google Sites

Antes de realizar el pilotaje en el curso de Sistemas Numéricos del semestre 2022 – II, nuestro asesor nos enseñó la herramienta gratuita *Google Sites*, en la cual podríamos subir el programa y complementarlo con información relacionada al mismo, logrando así crear un sitio web (<https://sites.google.com/view/seppea/seppea>).

Google Sites es una herramienta gratuita que ofrece *Google* para la creación de espacios web sencillos, en el que se pueden crear pestañas y subpestañas agregando en las mismas contenido

como textos, imágenes, videos, GIF e incluso contenido vinculado de otras páginas, como en este caso de GeoGebra, todo esto gracias a que *Google Sites* da la opción de adjuntar el programa a través de código HTML, que se genera en el momento en que el Applet está disponible en el sitio en línea de GeoGebra.

El uso de este sitio permite crear varias pestañas y definir secciones acordes a nuestro interés, ya que en algún momento se pensó que se podría separar cada una de las propiedades en pestañas independientes, reduciendo el tiempo de ejecución al no tener todas las propiedades en un mismo programa. Sin embargo, como hay propiedades y variables que dependen de otras, no resultaba conveniente dejar las propiedades por separado sin que se tuviera alguna alteración.

Resultados del pilotaje.

El pilotaje se realizó a través de *Google Sites* en uno de los cursos de Sistemas Numéricos del semestre 2022 – II, en el cual los estudiantes ya tenían conocimientos previos de las estructuras algebraicas como lo es su definición, las propiedades, los tipos de estructuras, entre otros. Dicha actividad fue orientada por nuestro asesor y profesor a cargo de este espacio académico, teniendo la intención de que los estudiantes interactuaran con el programa y enseguida, respondieron un breve cuestionario en Forms, que consistía en las siguientes preguntas:

- ¿Qué opinión tiene sobre el programa?
- ¿Qué le pareció la interfaz del programa?
- ¿Qué tan complejo es el programa?
- Nombre una característica positiva y un aspecto por mejorar del aplicativo.

A partir de las respuestas de los estudiantes, resultaron dos aspectos principales a mejorar en el programa e incluso fuera del mismo. El primero es mejorar la interfaz del aplicativo, pues visualmente no resultaba tan agradable para la mayoría de los estudiantes tal como mencionó un

estudiante al decir que “los colores pueden cambiarse para verse mejor”. El segundo aspecto, es generar una introducción o guía breve del funcionamiento del programa para aquellos que no tengan conocimientos previos acerca de las estructuras algebraicas, pues un comentario sobre esto es que “resulta confuso para las personas que no tienen los conocimientos” (Anexo 2).

Modificaciones al programa

De acuerdo con los aspectos por mejorar referente a la interfaz, la construimos de tal manera que suplieran las falencias de los programas encontrados y descritos brevemente en el Capítulo II, haciendo el uso más provechoso de las herramientas que ofrece GeoGebra para la creación de la interfaz de SEPPEA. Por lo tanto, tomamos la decisión de cambiar los colores de la interfaz y reubicar algunos textos, pues por cuestiones que dificultan la ejecución del programa, es complicado agregar varios elementos visuales o imágenes (Ver Ilustración 41).

SEPPEA

Número de clases= $2 \leq n < 10$

La operación 11 es conmutativa
La operación 11 es asociativa
El elemento neutro de la operación 11 es 0
Los inversos de 11 son $\{(2, 1), (1, 2), (2, 2), (0, 0)\}$

La operación 12 es conmutativa
La operación 12 es asociativa
El elemento neutro de la operación 12 es 1
Los inversos de 12 son $\{(2, 2), (1, 1)\}$

La operación 12 no es distributiva respecto a 11

	A	B	C	D	E	F	G	H	I	J	K
1	11	0	1	2	3	4	5	6	7	8	9
2	0	0	1	2	3	5	5	6	0	0	0
3	1	1	1	0	2	7	6	10	0	0	0
4	2	2	0	0	0	6	10	11	0	0	0
5	3	3	2	0	0	10	11	12	0	0	0
6	4	4	5	6	10	11	12	13	0	0	0
7	5	5	6	10	11	12	13	14	0	0	0
8	6	6	10	11	12	13	14	15	0	0	0
9	7	0	0	0	0	13	0	0	0	0	0
10	8	0	0	0	0	0	0	0	0	0	0
11	9	0	0	0	0	0	0	0	0	0	0
12											
13	12	0	1	2	3	4	5	6	7	8	9
14	0	0	0	0	0	0	0	0	0	0	0
15	1	0	1	2	3	4	5	6	0	0	0
16	2	0	2	1	2	11	13	15	0	0	0
17	3	0	3	1	2	15	21	24	0	0	0
18	4	0	4	11	15	22	26	33	0	0	0

Ilustración 41. Nueva interfaz

Frente al aspecto de la guía o introducción, la decisión más acertada era crear por fuera una explicación para no modificar la programación. Por tal motivo, dentro de la misma herramienta de *Google Sites*, creamos unas pestañas para tener un entendimiento más completo del programa, pues

el hecho de tener este únicamente en GeoGebra inhibe la comprensión adecuada del programa y, por el contrario, allí se puede incluir un complemento para el abordaje del mismo.

Específicamente, se crearon unas pestañas para aludir al concepto de estructuras algebraicas, sus propiedades y operaciones inmersas, como también una breve guía para entender el funcionamiento del programa (Ver Ilustración 42 - Ilustración 43).



Ilustración 42. Vista Google Sites Computador



PASO 1: Ingresar el número de clases a trabajar en el programa

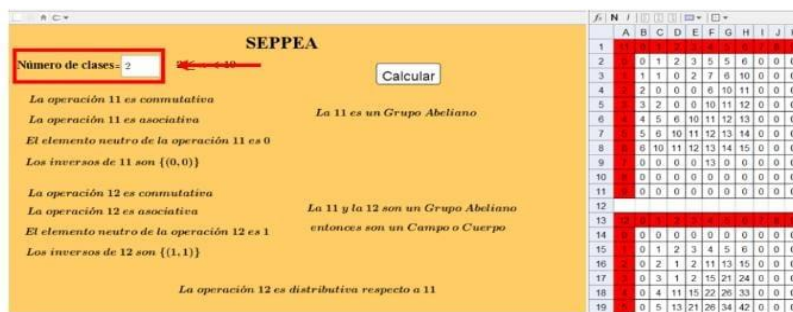


Ilustración 43. Vista Google Sites Celular

CAPÍTULO IV. CONCLUSIONES

En este capítulo se presenta un análisis comparativo en torno a los objetivos planteados y el desarrollo de la investigación, haciendo una reflexión sobre el alcance y contribución de esta propuesta de trabajo de grado. Dicha reflexión refleja nuestros conocimientos, habilidades y actitudes adquiridas a lo largo de estos años, con el fin de reconocer como esto constituyó una base fundamental para lo que fue el desarrollo de este trabajo y viendo una vez más que estamos “a hombros de gigantes”, dado que, sin los antecedentes referidos en el documento, no se habría dado de la forma esperada la conformación y construcción de la propuesta.

En resumen, hicimos una revisión de algunos referentes bibliográficos asociados a la construcción de softwares sobre estructuras algebraicas, los cuales fueron tres principalmente: *Actividades matemáticas para el desarrollo de procesos lógicos: Representar estructuras algebraicas finitas y enumerables* de Luque, Jiménez y Ángel (2009), *Software para la construcción de operaciones distributivas con respecto a otras dadas* desarrollado por Molina, Luque y Ángel (2005) y *HESEI* una propuesta de Martínez (2010) en Cuba. Teniendo en cuenta esto, se analizaron algunos softwares que permitieran desarrollar nuestra idea y prever una solución a las limitantes encontradas a través de herramientas como HTML, Java Script, MatLab Online, MIT App Inventor 2, GeoGebra 5 y Python. Además, una búsqueda en bases de datos como Google Play y Chrome para encontrar aplicativos o softwares asociados a la enseñanza e investigación en álgebra, pero más específicamente alrededor del álgebra abstracta y las estructuras algebraicas.

Con la finalidad de determinar el alcance y cumplimiento del objetivo general, es necesario en un primer momento revisar los objetivos específicos. Respecto al primer objetivo realizamos una búsqueda de los diferentes softwares desarrollados (tanto en computadores y dispositivos móviles) en diferentes categorías del álgebra enfocados hacia la investigación y la enseñanza, como

insumo para reconocer particularmente el desarrollo alrededor del álgebra abstracta, hallando diferentes aplicativos sobre estructuras algebraicas y sus propiedades, así como se mencionó previamente, siendo estos aplicativos los referentes más importantes para la construcción de nuestro propio programa.

En ese sentido, se estudiaron las falencias y limitaciones, como también las fortalezas de los programas mencionados, con la intención de que durante el desarrollo de nuestro aplicativo se solventaran estas y se tuvieran en cuenta las fortalezas encontradas en los programas. Lo anterior, requería de un estudio y análisis de diferentes softwares como se planteó en el segundo objetivo específico del trabajo, en vista de que las herramientas y funcionalidades del software seleccionado para la programación del aplicativo, nos diera la posibilidad de dar solución efectiva a las falencias y al mismo tiempo, sacando el mayor provecho de las fortalezas de tales programas. Por lo tanto, después del estudio enfocado desde las ventajas y desventajas, optamos por escoger GeoGebra, pues este software se familiariza con un lenguaje de programación conocido previamente y satisface una de las necesidades más relevantes que propusimos, siendo esta la accesibilidad gratuita y en línea al programa.

Una vez seleccionado el software, nuestra meta fue concentrarnos en la programación de las propiedades de las estructuras algebraicas, teniendo en cuenta las particularidades que queríamos aludir en el aplicativo, tales como que el programa le indicará al usuario que propiedades o no cumplía la estructura a partir de las tablas de Cayley, determinando que tipo de estructura se conforma dependiendo del número de clases que asigna inicialmente el usuario. De esta manera, se concretó la primera versión del aplicativo SEPPEA en GeoGebra, atendiendo a los aspectos mencionados anteriormente y como se alude en el capítulo III.

Dada la construcción del programa en GeoGebra, era ideal publicar este en la plataforma en línea de GeoGebra, facilitando su accesibilidad. Sin embargo, queríamos que la accesibilidad

en línea fuera mayor y no se enfocara únicamente en la plataforma de GeoGebra, por lo que buscamos una herramienta para solucionar esto. Por eso, es que con relación al objetivo tres, se tiene una primera versión del programa y por recomendación de nuestro asesor se adecua el programa en *Google Sites*, con el fin de que el programa tenga un rango de visibilidad mayor en la web.

Así, debíamos poner en práctica la ejecución del programa en un grupo de estudiantes, como se menciona en el objetivo cuatro, por lo que hicimos la implementación del aplicativo en el curso de Sistemas Numéricos del semestre 2022 – II, y a partir de dicho proceso se obtuvieron sugerencias y comentarios de los estudiantes recogidos en una breve encuesta de *Forms*. Con base en esto, se realizaron algunas modificaciones al programa en GeoGebra y a la disposición de la página en *Google Sites*, como lo fue la reestructuración de la interfaz para que sea más agradable para el usuario y la creación de pestañas en *Google Sites*, en las cuales se pretende que la interacción entre el usuario y el programa sea más satisfactoria y un poco guiada. En consecuencia, obtenemos la versión final del programa, respondiendo a los objetivos cuatro y cinco planteados en el trabajo.

El producto de los objetivos específicos y su impacto a lo largo del trabajo, se convierten en un insumo para determinar el alcance del objetivo general. Finalmente, se logró la construcción del aplicativo SEPPEA, el cual analiza las propiedades y operaciones de una estructura algebraica, determinando aquellas que cumplen o no. No obstante, y a pesar de nuestra intención por resolver aquellos problemas encontrados en otros programas, la ejecución del programa en algunos casos es un poco lenta debido a la gran cantidad de códigos de programación que debe realizar el programa, ya que muchos comandos dependen entre ellos mismos para la verificación de las propiedades.

Desde otro punto de vista, la creación de este aplicativo nos muestra una visión para nosotros mismos de la comprobación de las propiedades de las estructuras desde una perspectiva

lógica, en el sentido que no comparamos los resultados de la operación de la estructura, sino de las posiciones dentro de las matrices, ya que el programa quedaría aún más lento para su ejecución si hubiéramos optado por determinar la programación de las propiedades desde su definición.

En síntesis, el desarrollo de esta propuesta nos dejó para nuestro quehacer docente una herramienta que podemos aplicar en un futuro con nuestros estudiantes, como un medio para la enseñanza y el aprendizaje del estudio de las propiedades de las estructuras algebraicas desde una metodología diferente a la que hacemos mención en el primer capítulo. Además del aporte en el aspecto profesional y académico al involucrarnos en un proceso investigativo frente a un objeto matemático y su orientación hacia la educación, siendo esto uno de los aspectos más importantes como educadores matemáticos, en el sentido de promover la creación de investigaciones en el ámbito educativo, tanto dentro y fuera del aula, con el fin de fomentar en los estudiantes habilidades y competencias matemáticas.

REFERENCIAS

- Ángel, L., Molina, O., y Luque, C. (2005). *Software para la construcción de operaciones distributivas con respecto a otras dadas*. Memorias XV Encuentro de Geometría y III de Aritmética. Tomado de:
<http://funes.uniandes.edu.co/5983/1/AngelSoftwareGeometr%C3%ADa2005.pdf>
- Arcavi, A. (2018). Hacia una visión integradora de la enseñanza y el aprendizaje de las matemáticas. *Educación Matemática*. 30(2), pp. 33 – 48.
<https://doi.org/10.24844/EM3002.02>
- Ávila, A. (2016). La investigación en Educación Matemática en México: Una mirada a 40 años de trabajo. *Educación Matemática*. 28(3), pp. 31 – 59.
<https://doi.org/10.24844/EM2803.02>
- Barrera, F. (2007). *Historia del álgebra*. Universidad Nacional Autónoma de México. Tomado de: <http://profesores.dcb.unam.mx/users/ericagv/algebra/historia%20del%20algebra.pdf>
- Cáceres, A. (2004). *Lenguajes de Programación*. Instituto Tecnológico de Monterrey.
- Casado, M. (2006). *Manual básico de MatLab*. Universidad Católica de Manizales. Tomado de: <https://webs.ucm.es/centros/cont/descargas/documento11541.pdf>
- Castro, S., Guzmán, B., y Casado, D. (2007). Las TIC en los procesos de enseñanza y aprendizaje. Instituto Pedagógico de Caracas. *Revista de Educación*. 13(23), pp. 213 – 234. Tomado de: <https://www.redalyc.org/pdf/761/76102311.pdf>
- Centro de Innovaciones y Soluciones Empresariales y Tecnológicas (s. f.) *Software – Concepto y tipos de software*. Tomado de: <https://www.ciset.es/glosario/480-software>

Delgado, S. (2023). *Aprende Python*. Tomado de:

<https://aprendepython.es/downloads/907b5202c1466977a8d6bd3a2641453f/aprendepython.pdf>

Ferrer, J., García, V., y García, R. (s. f.). *Curso completo de HTML*. Tomado de:

<http://es.tldp.org/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>

Fundación Esplai. (s. f.). *Crea tus Apps con MIT App Inventor: Materiales de inicio a la programación con Android*. Tomado de:

https://alfabetizaciondigital.fundacionesplai.org/pluginfile.php/9584/mod_resource/content/6/Crea%20tus%20APPS%20con%20MIT%20App%20Inventor.pdf

Galán, A., y Patiño, M. (2019). *Visualización de diferentes objetos que se estudian en la topología algebraica con ayuda de software de geometría dinámica*. Universidad Pedagógica Nacional. Tomado de:

<http://repository.pedagogica.edu.co/bitstream/handle/20.500.12209/10521/TE-23358.pdf?sequence=1&isAllowed=y>

Gros, B. (2014). *Del software educativo a educar con software*. Universidad de Barcelona.

Tomado de: https://www.researchgate.net/profile/Begona-Gros/publication/237645375_DEL_SOFTWARE_EDUCATIVO_A_EDUCAR_CON_SOFTWARE/links/00463528f423c9a29f000000/DEL-SOFTWARE-EDUCATIVO-A-EDUCAR-CON-SOFTWARE.pdf

Luque, C., Jiménez, H., y Ángel, L. (2009). *Actividades matemáticas para el desarrollo de procesos lógicos. Representar estructuras algebraicas finitas y enumerables*. Universidad Pedagógica Nacional. <http://repository.pedagogica.edu.co/handle/20.500.12209/3437>

Mañas, J. (2013). *Utilización de las TIC en el aula: Geogebra y Wiris*. Universidad de Almería.

Tomado de:

<http://repositorio.ual.es/bitstream/handle/10835/2289/Trabajo.pdf?sequence=1>

Marqués, P. (s.f.). *El software educativo*. Universidad Autónoma de Barcelona. Tomado de:

https://recursos.salonesvirtuales.com/assets/bloques/educativo_de_pere_MARQUES.pdf

Martínez, Y. (2010). *La enseñanza de las estructuras algebraicas con la utilización de un*

sistema inteligente. Tomado de: [https://1library.co/document/zpnv6x4y-ensenanza-](https://1library.co/document/zpnv6x4y-ensenanza-estructuras-algebraicas-utilizacion-sistema-inteligente.html)

[estructuras-algebraicas-utilizacion-sistema-inteligente.html](https://1library.co/document/zpnv6x4y-ensenanza-estructuras-algebraicas-utilizacion-sistema-inteligente.html)

Mellar, H. (1997). Diseño de software educativo o de software escolar. Universidad de los Andes.

Revista Informática Educativa.

Moreno, R. (2016). *Desarrollo de aplicaciones para Android usando MIT App Inventor 2*.

Tomado de: <https://www.bolanosdj.com.ar/MOVIL/LENGUAJES/Usando-MIT-App-Inventor-2.pdf>

Pérez, O. (2018). La evolución del concepto de estructura en la matemática moderna (1830 –

1850). *Scripta Philosophia Naturalis*. 13, pp. 39 – 55. Tomado de:

<https://dialnet.unirioja.es/servlet/articulo?codigo=7476639>

Rives, P. (2011). *Manual Java Script: Formación para el empleo*. Editorial CEP. Tomado de:

<https://waltercarnero.com/cfp/tpprgweb/Libro2.pdf>

Sánchez, J. (2003). *Java Script: Manual de referencia*. Tomado de:

[https://ns2.elhacker.net/descargas/manuales/Lenguajes%20de%20Programacion/Javascript/javascript%20\(2\).pdf](https://ns2.elhacker.net/descargas/manuales/Lenguajes%20de%20Programacion/Javascript/javascript%20(2).pdf)

Segura, A. (2008). ¿La docencia y la investigación son aspectos complementarios?

Investigaciones Andina. 10(17), pp. 46 – 57.

ANEXOS

Anexo 1. Lista comandos de GeoGebra

Comando	Explicación
1. Secuencia (<Expresión>, <Variable>, <Valor inicial>, <Valor final>, <Incremento>)	Da como resultado una lista de todos los objetos que se obtienen al evaluar la variable en la expresión, mientras esta varía desde el valor inicial hasta el valor final con un incremento determinado. (si no se especifica el incremento GeoGebra lo asume como 1).
2. Elemento(<Lista>, <Número (posición)>) 3. Elemento(<Matriz>, <Fila>, <Columna>)	Toma el valor u objeto que se encuentra en una lista o matriz, en el que cada caso respectivo pide la posición o la fila y la columna.
4. Aplana(<Lista>)	En el caso de tener varias listas, este comando las une creando una nueva lista con los elementos de cada una respetando el orden de las listas. Dado el caso de una matriz coloca los elementos de la matriz en una lista respetando el orden por filas.
5. Suma(<Lista>)	Suma los valores de la lista, tanto en afirmaciones como numéricamente (verdadero = 1 y falso = 0).
6. Extrae(<Lista>, <Posición inicial>, <Posición final>)	Toma los valores de una lista desde una posición inicial hasta una final y los ubica en una nueva lista o en el caso de las matrices sirve para eliminar filas únicamente.
7. Traspone(<Matriz>)	Da como resultado la traspuesta de la matriz, es útil para poder eliminar columnas usándolo junto al comando extrae.
8. Suprime(<Lista>, <Lista>)	Elimina de la primera lista los elementos que se encuentren en la segunda lista.
9. Si(<Condición>, <Entonces>, <Si no>)	Se obtiene como resultado lo expresado en el “entonces” sí se verifica la condición, si no se cumple el comando dará como resultado lo colocado en “si no”.
10. Ordena(<Lista>)	Ordena los valores de una lista de menor a mayor.

Anexo 2. Respuestas del pilotaje

¿Cuál es su opinión respecto al programa?	¿Qué le pareció la interfaz del programa?	¿Qué le pareció la interfaz de la página?	¿Qué tan complejo es el programa?
Esta divertida e interesante.	Bueno, aunque podría tener elementos que llamen más la atención (otros colores, letra, etc.).	Podría ser más didáctica y con más explicación.	Resulta confuso para personas que no tienen los conocimientos necesarios.
Ninguna			
Es muy interesante.			
Se debería crear una explicación breve en la introducción de cómo funciona el programa.	Un poco confusa la instrucción inicial no da entender que se toma en la tabla 3*3.	Bueno, aunque podría tener elementos que llamen más la atención (otros colores, letra, etc).	Si no sabemos como usarlo se nos volverá complejo a la hora del uso.
El programa debería ser más claro para su uso apropiado.	Es un poco lento, pero sé que sale de las manos de los	No es muy atractiva. Mejorar los colores, el azul cansa la vista.	No es fácil de entenderlo.
No se entiende muy bien como usaría.	Los colores pueden cambiar para verse mejor.		Mucho, sería bueno que indicaran cada cosa, porque así es muy difícil saber que es cada cosa del programa y de la página.
Más claridad respecto a la función del programa.	Esta muy interesante		
Se debe dar una explicación previa al programa.	Es un poco lento, algo entendible, pero igual incómodo. Los colores no combinan.		
Ser más claros del como usar el programa.			
El programa está muy bueno, solo creo el número debe estar entre $2 < n < 5$.	No es muy clara y poco agradable.	Después de una explicación es entendible y "fácil de manejar".	
Muy interesante, se necesita buenas bases en sistemas numéricos para poder entenderlo.			Pues en cierta forma es complejo porque no es claro que significa los números que se ingresan.

Anexo 3. Código de programación de HTML del aplicativo SEPPEA

- <https://docs.google.com/document/d/1IQquC8d0zCq5Jqc3lsS8sCPvmONjWSBi/edit?usp=sharing&oid=113003769957762176511&rtpof=true&sd=true>