

**Desarrollo de un software libre educativo implementado en el área de matemáticas para
grado noveno**

Jonny Alexander Mueses Imbacuan

Código: 1126448830

Universidad Nacional Abierta y a Distancia - UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería - ECBTI

Tecnología de Sistemas

Ceres Valle del Guamuez (Putumayo)

Noviembre de 2017

**Desarrollo de un software libre educativo implementado en el área de matemáticas para
grado noveno**

Jonny Alexander Mueses Imbacuan

Código: 1126448830

Trabajo de Grado para optar por el Título de Tecnólogo de Sistemas

Director:

Andrés Felipe Hincapié Saldarriaga.

Universidad Nacional Abierta y a Distancia - UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería - ECBTI

Tecnología de Sistemas

Ceres Valle del Guamuez (Putumayo)

Noviembre de 2017

Contenido

1. Resumen	5
2. Introducción	7
3. Planteamiento del Problema	9
4. Formulación de la Pregunta	11
5. Justificación	12
6.2. Objetivos Específicos	15
7. Delimitación Temporal	16
8. Delimitación Financiera	16
9. Marco Conceptual y Teórico	18
9.1 Marco Teórico	18
9.1.1 Software libre:	18
9.1.2 Software educativo:	22
9.1.3 Evolución del Software Educativo	28
9.2 Marco Conceptual	35
9.2.1 Software Educativo de Matemáticas	35
9.2.2 Fases de la ingeniería del software	38
9.2.3 Modelos para el diseño y proceso del software	40
9.2.4 Algunos softwares de matemáticas	45
9.2.5 Terminología:	48
10 Marco Contextual	52
11 Metodología	54
11. 1 Línea de Investigación	54
11.2 Método de Investigación	54
11. 3 Técnicas de Análisis de la Información	55
11. 4 Análisis y muestra de resultados	57
11. 4.1 Fase de exploración y análisis	57
11.4.2 Tratamiento de la información	60
11. 5 Desarrollo del software	68
11. 6 Etapa de análisis	73
11.7 Etapa de diseño	83
11.8 Etapa de codificación	142
11. 9 Etapa de pruebas	143

11.10 Etapa de utilización.....	157
11.11 Etapa de mantenimiento.	159
12 Conclusiones.....	160
13 Referencias	162
14 Anexos.....	165

1. Resumen

El presente proyecto aplicado tiene como objetivo el desarrollo de un software libre educativo útil en el área de matemáticas, como base de apoyo desarrollado ampliamente para el colegio I.E.R.J.AS, de la Inspección del Placer (Municipio Valle del Guamuez-Putumayo) para los estudiantes del grado noveno.

El área de las matemáticas requiere del uso de metodologías y estrategias para la enseñanza de la misma, es importante cómo el docente puede darse a comprender, y como el estudiante puede captar la enseñanza del docente. Además, es de conocer que, en los colegios, y más aún en los rurales, el uso de este tipo de software en esta área es nulo, no hay uso de las TIC's, no hay un conocimiento amplio sobre el tema del software educativo. La enseñanza de las matemáticas es de tipo tradicional docente – alumno, el uso de metodologías de enseñanza básicas como el uso de libros de apoyo y explicaciones de los temas en el tablero.

Este proyecto tecnológico (software libre), está enfocado al mejoramiento de los procesos de aprendizaje en el área de matemáticas, en la innovación del software libre educativo, en la interacción dinámica entre el estudiante y el aplicativo, esto con el fin de que el estudiante le halle un interés más cercano al área de matemáticas, y tenga un aprendizaje más significativo. Este proyecto se plantea con el fin de que los estudiantes tengan un apoyo didáctico o interactivo y sobre todo que comparen de forma rápida sus resultados con los resultados del aplicativo, de acuerdo al ejercicio seleccionado, así denoten los errores y tengan un dinamismo en volver a hacer los ejercicios planteados. El proyecto se aplica como apoyo al estudiante, independientemente de cómo sea su formación académica, cuyo objetivo está en el uso del software por parte de este ya sea en el colegio o en la casa.

Con lo anterior, se pretende dar a conocer el uso del software libre educativo, brindar apoyo mediante el uso de este aplicativo, en el área de matemáticas para los estudiantes, y se espera que los alumnos y docentes lo usen y vean los beneficios que puede brindar el uso de este programa, y sobre todo que tengan un aprendizaje complementado y se sientan más motivados por aprender matemáticas y con ello un interés por las tecnologías.

Palabras claves: Software Libre, Software, Matemáticas, Programas, Visual Basic, Lenguajes de Programación, Informática, Multimedia, Software Educativo, Aprendizaje Didáctico, Aplicativo, Programas Interactivos, Tipos de Software Educativo, Ingeniería de Software, Evolución del Software Educativo, Desarrollo de Software, Modelos.

2. Introducción

El uso del software educativo ha sido de gran relevancia en muchos ámbitos y centros educativos, además que ha ayudado en el proceso de aprendizaje en los estudiantes en diferentes áreas. De hecho, que es de notar que el software educativo lo podemos ver desde dos perspectivas, de uso libre y de uso comercial, esto de acuerdo a las necesidades de quien desarrolla estos programas educativos.

De igual forma, notamos un gran impacto del software educativo en el área de las matemáticas, puesto que vemos muchos aplicativos o software para el uso de las matemáticas, y vemos que la matemática necesita de un apoyo de enseñanza donde el uso de la tecnología informática ayude de forma didáctica e interactiva en el aprendizaje de los alumnos.

Las matemáticas conllevan al desarrollo social, y son parte de la vida y progreso del ser humano, con esto vemos que se ha logrado mediante el uso de programas. Con lo anterior, se constata que el software educativo no sustituye al docente, sino que actúa como una herramienta útil para potencializar aún más el aprendizaje de los estudiantes.

La iniciativa y uso del software educativo va teniendo mayor aceptación en los centros educativos, y además es una herramienta que puede ser, o es utilizada fuera de los centros educativos, y así potencializar a un más el aprendizaje de los estudiantes en casa o en cualquier lugar adecuado, esto ayudando a estudiantes que llevan un proceso de aprendizaje lento, y así pueden ejercitarse con la ayuda de este software libre de matemáticas. Como es de entender que un programa educativo simplemente es una ayuda o complemento considerable a la metodología o enseñanza del docente.

El presente proyecto, desarrollo de un software formativo de matemáticas, se dará a conocer como solución a la problemática observada, y con esto incentivar el uso del software educativo, y expandir los beneficios de las TIC's y sobre todo el impulso del software libre.

3. Planteamiento del Problema

Uno de los mayores problemas para un centro de educación es, sin duda, el presupuesto. El dinero o medios que puedan facilitar los gobiernos son limitados, con lo que muchas necesidades quedan a menudo sin satisfacer debido a los altos costes de los productos informáticos. Colombia, un país en vía de desarrollo, aún está a paso lento en cuanto a la implementación del software libre, y a un más en la implementación del software libre en el ámbito de la educación.

Ahora bien, los estudiantes del grado noveno del mencionado colegio I.E.R.J.A.S, presentan un grado de aprendizaje significativo, y no se han presentado casos de que hayan perdido el área de matemáticas, en los últimos 4 años.

La problemática se identifica en la ausencia del uso del software educativo en esta área de las matemáticas que pueden complementar el aprendizaje de los alumnos, y que pueden ayudar a comprender los temas que se ven en grado noveno.

Y a lo anterior le agregamos el mayor tiempo que debe ocupar un docente en dar a conocer un tema, excediendo el tiempo límite establecido, y con ello limitando los temas a ver, donde se dan casos que no se ven todos los temas en el transcurso del año, o si los ven, el docente lo hace de manera generalizada o rápida. El mayor tiempo en ver un tema es por qué algunos estudiantes tienen un aprendizaje lento, no todos avanzan al mismo ritmo, algunos requieren de más tiempo para aprender, y sobre todo si se trata de las matemáticas.

Descrito lo anterior, la falta de nuevas metodologías de enseñanza en la educación (media, grado noveno), la falta de implementación informática y tecnológica, en el área de matemáticas, y sobre todo la falta de implementación de software educativo libre aplicado a esta área, puede generar algo de deficiencia de conocimientos por parte de los estudiantes del grado

noveno, en esta área, de la institución anteriormente mencionada, los estudiantes pueden presentar dificultades para aprender ciertos temas de matemáticas, en el lapso del tiempo establecido en la malla curricular o plan de enseñanza de las matemáticas para grado noveno. Quizás la inexistencia de software libre aplicado a esta área, conlleva a los estudiantes a interactuar en un sistema tradicional por lo que no todos avanzan a un mismo ritmo de aprendizaje, y más aún si la metodología del docente no es apropiada, los estudiantes no avanzan en el aprendizaje. Y con estas falencias muchos llegan a la educación superior con conocimientos muy deficientes en el área de matemáticas, tienen dificultades, y ante esto tienen que buscar ayuda por su cuenta.

Ante esto le agregamos la falta de material didáctico efectivo y acorde a esta área, falta de programas aplicativos que permitan observar, analizar, traer información que complementen el tema. Entonces, vemos que a la falta de programas accesibles (software libre) los estudiantes no manejan un aprendizaje más interactivo, más dinámico, donde ellos mismos se sientan más cómodos, más interesados en aprender de las matemáticas, donde pongan a prueba sus conocimientos, y puedan adquirir conocimientos sólidos; y sobre todo “que hayan aprendido”. La falta de un software libre aplicado a las matemáticas hace que muchos estudiantes se queden nulos en conocimientos matemáticos, y más aún vemos que en el ámbito educativo la implementación de programas interactivos es deficiente.

actualmente vemos que hay software privativo o de pago que están relacionados con el aprendizaje y los hay para esta área, pero no todos tienen acceso a estos programas por su costo, y es muy costoso porque para cada computador donde se le va a instalar el programa se debe tener una licencia diferente, y las licencias siempre son costosas. Las instituciones educativas no

han optado por esta alternativa, pero tampoco han tenido en cuenta la alternativa de la implementación del software libre.

hay software educativo para el área de matemáticas y que contienen ciertas funciones matemáticas, pero que la gran mayoría presentan los resultados de forma muy resumida, sin pasos previos de cómo fue que se hizo la solución, además hay programas que no cumplen con las expectativas, en el caso de las temáticas de matemáticas para grado noveno, donde haya un programa que sea específico y este diseñado para ayudar a estudiantes de noveno grado y que tenga las funcionalidades de las temáticas de noveno grado. Además, hay pocos programas matemáticos que tengan temas funcionales de grado noveno (en el área de matemáticas), y sobre todo que tengan alternativas de ejercitación y de consulta.

4. Formulación de la Pregunta

¿Cómo potencializar el aprendizaje en los alumnos mediante el uso del software educativo en el área de matemáticas?

5. Justificación

El área de matemáticas es un área que requiere de mucha atención, requiere apoyarla con material didáctico interactivo (programas), para que los estudiantes aprendan y no le cojan miedo o apatía, porque, para muchos estudiantes, las matemáticas, son su gran “antagonista”, el peor obstáculo en el aprendizaje académico, y muchos no quieren saber de esta área, le tienen pavor.

este programa educativo estimula al estudiante (del grado noveno) a abrir su conocimiento y a comprender más sobre temas básicos y obtener así, mejores resultados en el área de matemáticas, en las pruebas saber y como herramienta para fortalecer el aprendizaje matemático. Lo conlleva al estudiante a ser más interactivo, más relacionado con las matemáticas, y con ello también se adentra al campo de las TIC's, de la informática.

Este aplicativo, por ser un software libre (libertad que tiene un usuario para modificar, copiar, distribuir y modificar un software sin que ninguna compañía o individual pueda emprender acciones legales contra él), puede ser estudiado y mejorado por docentes de ingeniería de sistemas, o en el área de sistemas. Se diferenciará de otro software gratuito en cuanto a la solución de los ejercicios, donde se mostrará de forma sencilla el procedimiento o formas de cómo fue que se hizo el ejercicio, o cómo fue que el aplicativo dio la solución.

Por eso, con el ánimo de obtener mejores resultados académicos de los estudiantes del grado noveno en el área de matemáticas se considera pertinente realizar un aplicativo, y principalmente un software libre, que ayude a los estudiantes a interactuar en el campo de las matemáticas y les ayude aumentar o complementar su nivel de aprendizaje.

Esta es una de las circunstancias clave de las que podría aprovecharse el software libre para hacer su entrada en el ámbito educativo. La inexistencia de licencias por uso hace a los programas libres infinitamente más accesibles. En el entorno educativo, además de las ventajas

de tipo económico, el uso e implementación de software libre trae consigo muchos beneficios. Con la implementación del software libre no se pretende sustituir al profesor. La informática ha de ser entendida como una herramienta que ayude al profesor a impartir su doctrina de una forma más fácil y comprensiva para los alumnos. Además, el diseño del software libre contendrá alternativas de explicación para que el estudiante vea como se dio solución a un determinado ejercicio.

Y porque de un aplicativo para estudiantes del grado noveno, porque son estudiantes que deben tener conocimientos más sólidos y tener bases para poder desenvolverse en los dos últimos grados, puesto que algunos temas de matemáticas del grado noveno son las bases para las matemáticas de los dos últimos grados. De acuerdo a lo anterior, la implementación del software educativo en el área de matemáticas es una ayuda para los estudiantes del grado noveno, donde pueden complementar sus conocimientos, y sobre todo porque los motiva a acercarse más a las matemáticas, y aprender de manera interactiva, y por qué el uso de este software conlleva a las instituciones a tener más potencialidad en el aprendizaje de los estudiantes.

La implementación del proyecto aplicado (diseño de un software educativo), puede ser de gran importancia y de gran ayuda tanto para los alumnos como para el docente, puesto que contiene las temáticas funcionales que requieren de mayor comprensión y atención, además porque puede potencializar el aprendizaje de manera colaborativa y autónoma, motiva más al estudiante a aprender las matemáticas, además se implementa en el colegio o centro educativo el uso del software educativo de forma libre, y con esto a generar un interés, por parte de docentes de informática y de matemáticas, por el diseño de más software educativo en esta área y en otras áreas fundamentales, además de acercar al estudiante con las matemáticas de diferentes formas usando metodologías más factibles y por supuesto usando las TIC's.

Como se mencionó anteriormente el uso del software educativo conlleva a que sea usado no solo en el aula sino fuera de ella, claro está, con fines educativos. Esto ayuda a que los estudiantes se acerquen más al computador se familiaricen más, conozcan los beneficios que pueden obtener al usar estas herramientas informáticas o programas. En este caso podría ayudar a los estudiantes que presentan un aprendizaje lento en matemáticas, ya que ellos, al acceder al programa podrían usarlo fuera del aula, por ejemplo, en la casa, en tiempo libre, con la ayuda de un computador, y así aprender de manera autónoma y ejercitarse más, e incluso ir avanzando en temas nuevos.

Con lo anterior, se agiliza la enseñanza por parte del docente, se abarcan todos los temas durante el año, además el docente puede apoyarse en estas herramientas para impartir sus clases de matemáticas e ir avanzando equilibradamente. En el software a implementar el estudiante no solo puede ver la parte funcional del programa o más bien dicho ver el resultado de un cierto problema ingresado, sino ver información en forma de consulta y además de ejercitarse mediante el uso de ejercicios planteados, y ver ejercicios resueltos paso a paso de diferentes temas.

6. Objetivos

6.1. Objetivo General

Diseñar e implementar un software educativo (libre) en el área de matemáticas, basado en temas básicos para el grado noveno de la I.E.R.J.A.S

6.2. Objetivos Específicos

- Implementar el software educativo en el área de matemáticas.
- Diseñar un software educativo fundamentado en temas básicos de matemáticas para grado noveno.
- Fortalecer el aprendizaje a través de las herramientas informáticas.
- Impulsar a los estudiantes a familiarizarse con las matemáticas.
- Motivar a los estudiantes con el uso de aplicaciones didácticas (programas).

7. Delimitación Temporal

Tabla 1: Cronograma de actividades para el desarrollo del software educativo

1. CRONOGRAMA DE ACTIVIDADES												
ACTIVIDAD	MES 1 dic 15	MES 2 ene 15	MES 3 feb 15	MES 4 mar 15	MES 5 abr 15	MES 6 may 15	MES 7 jun 15	MES 8 jul 15	MES 9 ago. 15	MES 10	MES 11	MES 12
Revisión de material bibliográfico	2016	2017										
Recolección de información mediante información	2016	2017										
Planificación del desarrollo del software		2017	2017									
Gestión del desarrollo del software			2017	2017	2017	2017						
Realización de ajustes finales							2017	2017				
Evaluación del software									2017			

8. Delimitación Financiera

Tabla 2: Recursos y presupuesto para el desarrollo del software educativo.

RECURSO	DESCRIPCIÓN	PRESUPUESTO
Equipo Humano	<p>Personas encargadas del diseño del software libre educativo, del análisis de la problemática, de las pruebas del software, del manejo del software.</p> <p>Aquí tenemos: estudiante, en este caso mi persona, orientadores, personas con conocimientos de lenguajes de programación.</p> <p>Tenemos también los actores como los estudiantes y el docente.</p>	Orientadores: 20000
Equipos y Software	Son recursos físicos y lógicos que permitirán el desarrollo del proyecto, la planeación y evaluación.	visual BASIC 2010 Ultimate:

	Aquí tenemos: computadores, sistema operativo Windows 10, visual BASIC 2010 Ultimate, Microsoft office, software de multimedia, software de modelado como DÍA, otros como: Cmap Tools, editor de ecuaciones.	10000
Viajes y Salidas de Campo	Son medios utilizados para el análisis de la problemática. Aquí se maneja lo que es la visita al centro educativo para presentar el software educativo, viajes a la ciudad para adquirir recursos tanto físicos como humanos para implementar el proyecto.	20000
Materiales y suministros	Son recursos útiles que servirán de apoyo para el diseño del proyecto y desarrollo del software, se maneja lo que es la búsqueda de información en internet, fuentes bibliográficas, libros de programación orientada a objetos, de informática, de Visual Basic, información en medios físicos y en internet, videos, impresiones.	10000
Bibliografía	Libros, revistas, artículos, fuentes de ciencias informáticas, internet.	8000
TOTAL: 68000		

9. Marco Conceptual y Teórico

9.1 Marco Teórico.

9.1.1 Software libre:

El Software Libre es aquel programa informático en el que los autores han dado la autorización explícita en la licencia de uso para que las personas podamos utilizarlo, copiarlo, distribuirlo, estudiarlo y modificarlo. De modo más preciso, se refiere a cuatro libertades que los autores de los programas le entregan a los usuarios:¹

1. Libertad para ejecutar el programa con cualquier propósito: comercial, educativo, etc.
2. Libertad para estudiar y modificar el programa con cualquier propósito. Para ello es condición necesaria que permita el acceso al Código Fuente, que es lo que permite saber cómo está hecho el programa.
3. Libertad para redistribuir copias del programa, tanto gratis como por un precio, con la condición de no restringir las libertades originales e indicar en dónde se encuentra la fuente original del programa.
4. Libertad para distribuir versiones modificadas del programa, de tal manera que otras personas puedan beneficiarse con sus mejoras. Para esto se requiere que el programa modificado se entregue con una Licencia de Software Libre y hacer el reconocimiento a los autores originales.

El software libre contiene diferentes tipos tales como sistemas operativos, aplicaciones, utilidades, de diagnóstico, multimedia entre otros.

¹ Ulises Hernández Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

Diferencias del software libre:

El software se puede dividir en: libre, privativo, gratuito y de prueba. Cada uno, como cualquier bien o servicio, tiene sus características, modelo de negocio y público objetivo particular. Miremos sus diferencias:²

El Software Privativo explicita a los usuarios la prohibición de redistribuir, estudiar y modificar el software, ya que su modelo de negocios está centrado en la venta de permisos o licencias de uso. Es el software que la mayoría conoce, tal como el sistema operativo Windows, la suite ofimática de Microsoft Office, el programa de edición de imágenes Photoshop o el programa de edición de videos Adobe Premier.³

El Software Gratuito (Freeware), limita los usos que se pueden dar al programa, por ejemplo, indicando que sólo es para fines educativos, pero no comerciales. Además, no permite su estudio, ni su modificación. Algunos ejemplos son: el programa de mensajería instantánea Windows Live Messenger, el antivirus AVG o Avira, el lector de archivos PDF Adobe Reader, el reproductor de música Winamp o el cliente de videollamadas Skype. Generalmente las personas confunden Software Libre con Software Gratuito, porque ambos se obtienen sin incurrir en costos.⁴

Finalmente está **el Software de Prueba (Shareware)**. Este se consigue con funcionalidades limitadas y para usar de forma gratuita sólo por un período de tiempo. Si una vez probado se quiere seguir usando de forma legal, el usuario debe pagar la licencia

² Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

³ Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

⁴ Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

correspondiente. Entre los programas que se consiguen con esta característica está: los compresores de archivos WinRAR y WinZip, los antivirus NOD32 y Kaspersky y el quemador de CD/DVD Nero.⁵

Es común que las personas piensen que el Software Libre es un proyecto de unos cuantos estudiantes que lo toman por hobby, pues no ven en sus cuatro libertades un lugar para la rentabilidad y el lucro. Sin embargo, el Software Libre tiene un modelo de negocios poco conocido, pero que ha mostrado utilidades importantes en varios sectores de la economía.

Uno de los primeros aspectos que conviene destacar, al abordar el tema de las tecnologías de la información y la comunicación (TIC) desde la óptica educativa, es la relación existente entre la evolución tecnológica, el desarrollo de las tecnologías de la información en la sociedad y su introducción en los sistemas educativos. Las actuales tecnologías permiten la articulación de procesos sociales a distancia, ya sea en las diferentes áreas de forma interdisciplinaria o comercial. De manera que se entiende que en la evolución de estos prodigios se va dando la globalización enmarcados en las diferentes interacciones con el contexto.⁶

En relación con lo anterior, en la actualidad en el sector educativo es necesario comprender la evolución de la tecnología y de la sociedad, de una manera lúdica y participativa donde se puede cambiar la concepción de aquellos grandes temas que están marcando el siglo XXI, sin pensar sobre lo que se entiende, o sin medir la relación que los sistemas educativos están desarrollando en lo que hoy en día se suele llamar espacio virtual.⁷

⁵ Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

⁶ YORLENNY CASTILLO PIEDRAHÍTA, CRISTINA MARIA DUQUE FLÓREZ, ENORIS GUZMÁN MORENO

⁷ YORLENNY CASTILLO PIEDRAHÍTA, CRISTINA MARIA DUQUE FLÓREZ, ENORIS GUZMÁN MORENO

Razones para utilizar Software Libre en las Instituciones Educativas

Si las instituciones de educación enseñan a utilizar Software Libre, entonces los estudiantes utilizarán Software Libre cuando se gradúen, con lo cual se está favoreciendo un cambio en la mentalidad de la sociedad al promover:⁸

- ✓ La cooperación: Está bien compartir los programas que tengo si otro lo necesita.
- ✓ El valor de la libertad: Puedo utilizar los programas con cualquier propósito, no necesito pedirle permiso a nadie sobre lo que hago con ellos, y ambas cosas son legales.
- ✓ La emancipación del consumismo tecnológico: Yo escojo qué programas utilizó, qué versiones y en qué computadores sin importar la moda o las exigencias hardware.

Requerimientos para utilizar Software Libre en una Institución Educativa

Para empezar, se requiere entender que lo importante del Software Libre no está en los productos, sino en el tipo de valores (colaboración, autonomía, libertad) y dinámicas sociales (comunidades, desarrollo tecnológico endógeno) que promueve. Por lo tanto, el primer paso para la utilización de este tipo de software en una institución educativa, es el cambio de actitud frente a la tecnología por parte de los directivos y profesores. Esto implica promover una disposición para el autoaprendizaje, el trabajo colaborativo, la resolución de problemas y una profunda persistencia en torno al uso y aprovechamiento de la tecnología en procesos de formación y aprendizaje.⁹

Es indispensable contar, como mínimo, con uno o dos profesores que tengan facilidad para manejar y aprender sobre tecnologías, y que estén encargados de la sala de cómputo. En

⁸ Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

⁹ Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

contextos como el nuestro, la clave del Software Libre está en el desarrollo de competencias endógenas, por lo tanto, no se puede esperar a que personas externas a la institución educativa vengan a hacer el trabajo, ya que esto sería remplazar un tipo de dependencias por otra. Las asesorías y orientaciones de personas externas son necesarias, pero para lograr que las mismas personas de la institución educativa lideren la migración y aprendizaje correspondiente.¹⁰

9.1.2 Software educativo:

Son los programas que utilizan las computadoras para trabajar o funcionar, por lo tanto, un software educativo es cualquier programa computacional cuyas características estructurales y funcionales le permiten servir de apoyo a la enseñanza, el aprendizaje y la administración educacional. El software educativo es uno de los recursos más interesantes, necesarios y motivadores que se pueden utilizar para ayudar al desarrollo del proceso enseñanza-aprendizaje. Al introducir las computadoras en la educación, se produjo una forma más amena de aprender, logrando además la retención a más largo plazo del aprendizaje.¹¹

Características esenciales de los softwares educativos.

Teniendo como base la innovación y continuidad de la educación mediante el uso de nuevos métodos de enseñanza, donde cada vez los usos de las tecnologías informáticas son cada vez más relevantes, y que ciertos programas educativos computacionales acompañan al proceso de aprendizaje del estudiante mediante funcionalidades generalizadas o con temas centrados, estos softwares educativos presentan una serie de características que los podrían diferenciar de otros tipos de softwares. Veamos cuales son:

¹⁰ Ulises Hernandez Pino, Dr(c). en Ciencias de la Educación y Sandra Anaya Díaz, Ingeniera en Electrónica y Telecomunicaciones de la Universidad del Cauca.

¹¹ Yanitsia Blanco Fernández. Los softwares educativos.

- ❖ Son programas diseñados cuyo fin último u objetivo principal está centrado en la didáctica, para ser parte del aprendizaje para el estudiante, diseñados para fines educativos de complemento y de aprendizaje tanto colaborativo y autónomo, esto es estudiante – software.
- ❖ Como todo tipo de software, y como software en general, los softwares educativos utilizan el computador para su funcionalidad, aunque bien pueden ser instalados en otros dispositivos de visualización como Tablet, esto de acuerdo a la configuración del software. Teniendo como base el computador, los estudiantes puede explorar el aplicativo y ver sus funcionalidades, ejecutarlo, introducir datos, ver resultados, ejercitarse, consultar.
- ❖ Los softwares educativos, en su mayoría son interactivos algo que llama mucho la atención de los alumnos y los acerca más a las matemáticas, e incluso a otras áreas, los estudiantes se familiarizan más con la tecnología, con el computador y con los programas educativos. De igual forma que hay un intercambio de información entre el estudiante y el software reflejado en el computador. Al ser interactivos manejan un sin número de funciones como multimedia, animaciones, imágenes, audio, movimientos, el software como tal tiende a ser dinámico.
- ❖ Tienden a ser independientes, es decir, los estudiantes pueden aprender en sus actividades de forma individual, donde pueden avanzar o equilibrar en su aprendizaje, además el software educativo conlleva a la adaptación rápida de cada estudiante y manejar su ritmo de aprendizaje. Además, esto les facilita a los estudiantes a prender fuera del aula, aprender en casa y en tiempos libres, esto ayudando a estudiantes que tienen un lento aprendizaje o se les dificulta un poco aprender los temas de matemáticas.

- ❖ Los softwares educativos bien diseñados, son fáciles de manejar, de usar o ejecutar. Donde se requiere de conocimientos mínimos por parte del estudiante, básicos como la informática, el manejo del computador, el software se instala de forma sencilla, donde se puede aprender de forma rápida, de ahí esta en el uso ya del programa como tal, donde estos pueden tener a disposición su manual de uso. En términos generales son flexibles en su funcionamiento.
- ❖ Tienen una navegabilidad flexible y las interfaces son factibles y de calidad visual apta para los estudiantes.

Tipos de software educativo

Los softwares educativos tienen diferentes finalidades didácticas esto de acuerdo al área que manejen o según como estén diseñados, y de acuerdo a la finalidad didáctica, estos pueden ser de diferentes tipos o funcionalidades.

Según La Estructura (Márquez, 1995)

1. **Programas Tutoriales:** Son aquellos que dirigen en algún grado el trabajo de los estudiantes, este proceso se realiza a través de ciertas actividades previstas de antemano, los estudiantes ponen en juego determinadas capacidades y aprenden o refuerzan conocimientos y/o habilidades. Estos se subdividen a su vez en:
 - **Programas lineales:** que presentan al alumno una secuencia de información y/o ejercicios (siempre la misma o determinada aleatoriamente) con independencia de la corrección o incorrección de sus respuestas.
 - **Programas ramificados:** basados inicialmente también en modelos conductistas, siguen recorridos pedagógicos diferentes según el juicio que hace el computador sobre las respuestas de los alumnos para determinar la profundización de ciertos temas. Ofrecen

mayor interacción, más opciones, pero la organización de la materia suele estar menos compartimentada que en los programas lineales y exigen un esfuerzo más grande al alumno.

- **Entornos tutoriales:** Se basan en modelos pedagógicos cognitivistas, y proporcionan a los alumnos una serie de herramientas de búsqueda de información que pueden utilizar libremente para construir la respuesta a las preguntas del programa.
 - **Sistemas tutoriales expertos:** como los Sistemas Tutores Inteligentes (Intelligent Tutoring Systems), que, elaborados con las técnicas de la Inteligencia Artificial y teniendo en cuenta las teorías cognitivas sobre el aprendizaje, tienden a reproducir un diálogo auténtico entre el programa y el estudiante.
2. **Base de Datos:** Proporcionan datos organizados, en un entorno estático, según determinados criterios, y facilitan su exploración y consulta selectiva. Se pueden emplear en múltiples actividades como, por ejemplo: seleccionar datos relevantes para resolver problemas, analizar y relacionar datos, extraer conclusiones, comprobar hipótesis.
- **Bases de datos convencionales:** Tienen la información almacenada en ficheros, mapas o gráficos, que el usuario puede recorrer según su criterio para recopilar información.
 - **Bases de datos tipo sistema experto:** Son bases de datos muy especializadas que recopilan toda la información existente de un tema concreto y además asesoran al usuario cuando accede buscando determinadas respuestas.
3. **Simuladores:** Presentan un modelo o entorno dinámico (generalmente a través de gráficos o animaciones interactivas) y facilitan su exploración y modificación a los alumnos, que pueden realizar aprendizajes inductivos o deductivos mediante la observación y la manipulación de la estructura subyacente; de esta manera pueden

descubrir los elementos del modelo, sus interrelaciones, y pueden tomar decisiones y adquirir experiencia directa delante de unas situaciones que frecuentemente resultarían difícilmente accesibles a la realidad (control de una central nuclear, contracción del tiempo, pilotaje de un avión...).

- **Modelos físico-matemáticos:** Presentan de manera numérica o gráfica una realidad que tiene unas leyes representadas por un sistema de ecuaciones deterministas. Se incluyen aquí los programas-laboratorio, algunos trazadores de funciones y los programas que mediante un convertidor analógico-digital captan datos analógicos de un fenómeno externo al computador y presentan en pantalla un modelo del fenómeno estudiado o informaciones y gráficos.
 - **Entornos sociales:** Presentan una realidad regida por unas leyes no del todo deterministas. Se incluyen aquí los juegos de estrategia y de aventura, que exigen una estrategia cambiante a lo largo del tiempo.
4. **Constructores:** Son programas que tienen un entorno programable. Facilitan a los usuarios elementos simples con los cuales pueden construir elementos más complejos o entornos.
- ✓ **Constructores específicos:** Ponen a disposición de los estudiantes una serie de mecanismos de actuación (generalmente en forma de órdenes específicas) que les permiten llevar a cabo operaciones de un cierto grado de complejidad mediante la construcción de determinados entornos, modelos o estructuras, y de esta manera avanzan en el conocimiento de una disciplina o entorno específico.
 - ✓ **Lenguajes de programación:** como LOGO, PASCAL, que ofrecen unos "laboratorios simbólicos" en los que se pueden construir un número ilimitado de entornos. Aquí los

alumnos se convierten en profesores del computador. Además, con los interfaces convenientes, pueden controlar pequeños robots contruidos con componentes convencionales (arquitecturas, motores...), de manera que sus posibilidades educativas se ven ampliadas incluso en campos pre-tecnológicos.

5. **Programas Herramienta:** Son programas que proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos.
- ❖ **Procesadores de textos:** Son programas que permiten realizar actividades de producción de textos.
- ❖ **Gestores de bases de datos:** Sirven para generar potentes sistemas de archivo ya que permiten almacenar información de manera organizada y posteriormente recuperarla y modificarla.
- ❖ **Hojas de cálculo:** Son programas que facilitan la realización de actividades que requieran efectuar muchos cálculos matemáticos.
- ❖ **Editores gráficos:** Se emplean desde un punto de vista instrumental para realizar dibujos.
- ❖ **Programas de comunicaciones:** Son programas que permiten que computadores comunicarse entre sí a través de las líneas telefónicas y puedan enviarse mensajes, archivos, etc.
- ❖ **Programas de experimentación asistida:** A través de variados instrumentos y convertidores analógico-digitales, recogen datos sobre el comportamiento de las variables que inciden en determinados fenómenos.

Funciones del Software educativo.

El uso del software educativo en la educación presenta un apoyo didáctico, que como vemos, realizan funciones básicas de acuerdo a su diseño, cuyas funcionalidades dentro del aprendizaje pueden incrementar el aprendizaje de los alumnos en temas que requieren de más atención, es de notar que un software educativo no se comporta como una herramienta total que reemplace o sustituya al docente, ya que de ser así lo anterior sería un fracaso, puesto que un programa por sí solo no tiene la mayor funcionalidad u objetivo total de hacer aprender todos los temas, en este caso de matemáticas, al estudiante. El software educativo puede ser tanto efectivo como no tan efectivo dependiendo de cómo se lo use, del tipo de software escogido y las características que posee depende el potencial del aprendizaje del estudiante.

Entre las funciones que genera un software educativo tenemos:

- A. **Función informativa.**
- B. **Función instructiva.**
- C. **Función motivadora.**
- D. **Función evaluadora.**
- E. **Función investigadora.**

9.1.3 Evolución del Software Educativo.

Origen del Software Educativo y su relación con la Tecnología Educativa (Pedro torres-Ruiz).¹²

Pedro torres-Ruiz expone: “El software educativo tiene sus comienzos en Europa. Específicamente, en España en los años sesenta con el desarrollo de los medios audiovisuales en el mundo, por ejemplo, el proyector de diapositivas, acetatos, fotografías de prototipos, entre otros coincidiendo, además, con la propuesta de la Ley General de Educación en España en los

¹² Pedro Torres-Ruiz

años setenta pero, no fue sino hasta en los años ochenta cuando varios expertos en el campo de la educación habían puesto sus ojos en el Internet como tecnología de la información para su análisis, creación, desarrollo, difusión y divulgación de la enseñanza en cualquier nivel educativo. De todo lo anterior, surge la era digital en el que permitirá al ser humano interactuar puntos de vista con sus semejantes, a nivel mundial, al través de foros de discusión, mesas redondas virtuales e intercambio de ideas en sesiones on-line básicamente en tres formas:

- 1) Digitalización de la información. Es decir, leerlo en pantalla sin necesidad de tenerlo en impreso en papel
- 2) Manejar la información de forma virtual. Controlándolo y ordenándolo de acuerdo a las necesidades del usuario
- 3) Trabajar, en conjunto, con bases de datos heterogéneos (texto, imagen y sonido) así como la utilización de objetos digitales

Todo lo anterior, es gracias al avance tecnológico en cuanto a redes se refiere. Sobre todo, por el cambio de transmisión de datos analógicos a digitales propiciado a finales de la década de los noventa y también de las velocidades de recepción que cada vez más se manejan gigabytes por segundo.

La tecnología educativa, cuyo padre es David Skinner, ha propiciado la creación de campus virtuales. Por tal motivo, no deja de preocupar la desaparición paulatinamente de universidades y escuelas presenciales, así como de libros, periódicos y revistas, aunque difícilmente podrían disiparse del todo. Pues, es importante concretar como sociedad global que existe una vinculación directa entre la ciencia, la tecnología y la sociedad para la construcción de una tecnología educativa que va desde el análisis hasta la aplicación de los conocimientos en la

práctica sin descuidar las cátedras presenciales que, hoy en día, siguen aportando significativamente y que, es imposible erradicarlo de la noche a la mañana. Es importante señalar que la ciencia y la tecnología son elementos que dependen de la ideología y la cultura de cualquier sociedad del que se trate.

Hay que ahondar más en los conocimientos para plasmarlos en el software educativo como plantear problemas concretos que se presentan en la vida cotidiana de manera virtual e interactiva dando propuestas de solución o estar abierto a ello escuchando otras soluciones planteadas por usuarios que se encuentren en línea”.¹³

Como vemos, así como **el hardware ha evolucionado** también el software ha evolucionado y en conjunto siguen evolucionando, analizados estos en generaciones, fue de gran conveniencia que el hardware evolucionara de forma gradual para que el software también se pudiera acoplar al hardware y fueran compatibles. Desde los inicios existían programas especialmente para la programación y algunos softwares para uso científico y para análisis experimentales y de prueba, aun no se tenía en pensamiento lo del software educativo, es de notar que muy pocas instituciones educativas, especialmente superiores, tenían ciertos programas educativos. Posteriormente ya con la expansión de la tecnología, de la computación, de los computadores, de la nueva visión de la educación, de la integración de la tecnología con la educación, de los nuevos métodos de enseñanza, surge la iniciativa de desarrollar el software educativo a gran escala, de incorporar las TIC's a la educación como una ayuda al proceso de enseñanza académica, y vemos que esto se aplica tanto para educación básica, media, y para la educación superior, donde para cada situación se diseñan herramientas interactivas y didácticas de aprendizaje educativo.

¹³ Pedro Torres-Ruiz

De igual forma vemos a continuación los acontecimientos de la evolución del software educativo:

evolución del software educativo. ¹⁴

1975 - 1980: surgen los programas de propósito general y específico, estos son de apoyo en los procesos educativos, aunque en menor proporción y de manera limitada o reducida.

1985: surgen los programas que buscan potencializar aún más los programas (software) educativos, afianzar más la pedagogía de los softwares educativos.

1990 - 1995: surgen los programas y herramientas que hacen el uso del internet, programas de consulta.

2000: nace el aprendizaje virtual, programas de aprendizaje vía online, las plataformas educativas virtuales. Se diseñan cursos completos en la internet.

2005: surge el software social, es decir herramientas informáticas virtuales, encontramos los blogs, las wikis, se establecen las herramientas informáticas que generan el aprendizaje colaborativo, se expande aún más el software libre, encontramos plataformas virtuales o medios de comunicación virtual donde ciertos grupos interactúan entre si exponiendo sus temas, esto es con frecuencia en temas de investigación, en temas de educación, científicos, en la medicina, entre otros aspectos.

Juana Zavala Martínez. En su trabajo de investigación: “la computadora como material didáctico para la enseñanza de la matemática”. Afirma que, entre los componentes educativos, la

¹⁴ Briceño, Sergio y Ruth Molina, Aspectos pedagógicos de la evolución del software educativo [en línea]. Scribd. 2009 [consultado 01 abril de 2011]. Disponible en internet: <http://bit.ly/jnwl24>

computadora se incorpora como un poderoso medio que adquiere sentido con las diversas mediciones del maestro; principalmente la mediación cognitiva y la mediación cognitiva.

La potencia de la computadora ayuda a promover la intuición matemática y el entendimiento del escolar. Ya que este recurso didáctico que ayuda al alumno a sentir la conformidad con el curso.

Los sitios que localizamos en la red, relativos a educación matemática, son extranjeros especialmente norteamericanos, y la presentación de los contenidos, lo hacen con este corte tradicional con el que hemos convivido durante décadas. Aun en nuestro país no hay colegios que ya estén utilizando la computadora en la clase de matemática, pero tenemos la referencia de otros colegios extranjeros que usan este recurso didáctico teniendo resultados favorables. Por lo pronto se ha estudiado las actitudes de los alumnos hacia la computadora y todas las respuestas han sido positivas. En el caso concreto de la educación en los niveles de educación secundaria, todavía no se utiliza la computadora como recursos didácticos para la enseñanza de la matemática el cual forma parte del tedio a este curso por parte del alumno.

Rubén A. Pizarro en la tesis de: “Las TICS de la enseñanza de las matemáticas. Aplicación al caso de métodos numéricos” Los estudios realizados llegan a la siguiente conclusión:

Entre estas actividades podemos mencionar el análisis de bibliografía relacionada con la elaboración de software educativo. Sin lugar a dudas, la elaboración de software educativo es un tema analizado por diversos autores que coinciden en la importancia del mismo y su rol determinante en el proceso de enseñanza – aprendizaje. Las diferentes teorías sobre la forma en

que se logra el aprendizaje incluyen en su análisis el rol del software educativo y las distintas formas de incluirlo.

Las matemáticas fueron, en el ámbito educativo, la primera actividad que incorporo recursos tecnológicos que facilitaron significativamente las tareas que esta ciencia desarrolla.

De acuerdo a lo anterior, vemos que a través del tiempo y de la innovación tecnológica se han venido implementando el software educativo en diferentes áreas y temas de las mismas, y como vemos la implementación de un software educativo, y por ende, libre, se ha venido dando con mayor potencialidad en el campo de las matemáticas, un área donde se requiere de un aprendizaje constructivo, de un aprendizaje de mucha solides, atención, y donde los métodos usados para el aprendizaje deben ser de mucha calidad y eficiencia para que los estudiantes tengan conocimientos sólidos, y de hecho que los softwares educativos han conllevado a un aprendizaje más activo y potencial a través del tiempo, y por consiguiente esta nueva técnica de aprendizaje se está implementando en todas las instituciones, colegios, universidades, y es por eso que es una alternativa donde todos le debemos apostar.

De igual forma vemos teorías que sustentan y dan validez al desarrollo e implementación del software educativo en el área de matemáticas, veamos unas de ellas:

Las teorías de aprendizaje desde el punto de vista psicológico estado asociadas a la realización del método pedagógico en la educación, por ello las actuales innovaciones proponen enfoques a la instrucción matemática.¹⁵

Las teorías cognoscitivas tienen como su principal exponente al constructivismo, que cubre a través de una serie de postulados la cognición o como aprende la persona de manera

¹⁵ Jose Gullen, Jorge Briceño

individual, porque el aprendizaje es visto como un proceso de construcción del individuo, de allí, que el uso de la tecnología es vista como una manera activa de aprender. ¹⁶

Al respecto Hernan (citado por Salcedo, 2002) explica que:

“El uso interactivo del computador hace parte de una tecnología educativa derivada del estructuralismo genético de Piaget. Esta ciencia base ha desarrollado los instrumentos conceptuales y epistemológicos necesarios para darle significado empírico a enunciados a cerca del desarrollo de la inteligencia.” (p 2) ¹⁷

Debido a los valores elevados del licenciamiento de software privativo o de pago, los colegios y universidades están observando que el software libre es la mejor solución que hay en el momento para no perderse de la evolución informática, en el mercado tecnológico de Colombia, es una de las opciones para que las empresas reemplacen el sistema obsoleto de tecnología que tienen, donde las empresas de software propietario piden grandes cantidades de dinero para brindar un producto que supuestamente tiene garantía y soporte, llevando a las empresas a sacrificar gran parte de su patrimonio y en ocasiones, cuando el negocio no es lo suficientemente rentable para sostener el cambio de tecnología, puede llegar a la quiebra total.

De manera que hay que resaltar que en un proceso de enseñanza intervienen dos factores, o debe haber dos personas el docente y el estudiante o alumno. El estudiante hace una parte del aprendizaje ya sea de manera individual o de manera grupal, y en ambos casos presenta ciertas características de aprendizaje. El docente, quien es el complemento o parte de apoyo en el aprendizaje del estudiante, y en donde mediante el uso de metodologías y estrategias hace que el estudiante aprenda de manera significativa.

¹⁶ Jose Gullen, Jorge Briceño

¹⁷ Jose Gullen, Jorge Briceño

Ahora bien, el uso del software educativo no sustituye al docente, sino que le es un apoyo para amenizar el aprendizaje del estudiante, sea más activo, y por ende vemos que la educación ha venido evolucionando desde el sistema tradicional, hasta el uso de la tecnología y la informática, desde el uso de libros, hasta métodos de enseñanza más dinámicas, activas, interactivas, el uso del software educativo, donde el estudiante puede ser más práctico y aprender más matemáticas desde su casa, aumentar su nivel de conocimiento en esta área, practicarse más, donde puede aprender tanto en la escuela como en la casa.

9.2 Marco Conceptual.

9.2.1 Software Educativo de Matemáticas.

En la actualidad el uso de las TIC es un tema innovador, al cual debemos prestar atención para explotar sus ventajas sobre los estudiantes. Existen una gran cantidad de los mismos los cuales sirven para descubrir las capacidades de los alumnos utilizando la informática.

Podemos encontrar una gran diversidad de softwares educativos entre los cuales se encuentran los relacionados con las Matemáticas, existen miles de actividades matemáticas clasificadas por edades y objetivos específicos.

El desarrollo de software educativo, tanto privativo como libre, en la actualidad ha tenido un gran potencial, se han desarrollado con las herramientas más efectivas para ingeniería o desarrollo de software, y esto ha contribuido a un aprendizaje complementario por parte del estudiante, ya que estos programas cuentan con diseños y funcionalidades más didácticas e interactivas, teniendo en cuenta los requerimientos y necesidades de cada nivel educativo, y cuyo diseño cumple con las expectativas en el sector educativo.

Estos programas se han venido implementando y usando en centros educativos como en centros universitarios. El desarrollo de software educativo maneja herramientas propias para su creación, puesto que se trata de un programa educativo y que debe ser acorde al nivel educativo, y a las temáticas que se ven en matemáticas para cada grado y debe estar diseñado para enseñar al alumno.

Galvis menciona que un buen software educativo debe tener las siguientes características¹⁸:

1. Que tome en cuenta las características de la población destinataria.
2. Se adecue a los niveles de dominio diferenciado de los usuarios.
3. Que tenga la capacidad de llenar vacíos conceptuales, detectándolos y teniendo la forma de satisfacer a los usuarios.
4. Que tenga la capacidad de desarrollar habilidades, conocimientos y destrezas circunstanciales en el logro de los objetivos de aprendizaje.
5. Que explote sus propias potencialidades técnicas y de interacción.
6. Que promueva la participación activa de los usuarios en la búsqueda, generación, apropiación y reconstrucción del conocimiento.
7. Que permita vivir y reconstruir experiencias a los usuarios, lo cual sería difícil o imposible de lograr a través de otros medios.

Ahora bien, los softwares educativos no necesariamente cuentan con todas estas características, ya que cada uno se desarrolla con propósitos específicos y características propias,

¹⁸ Ysauro González Neri, Victoria Carmona Martínez y Sara Espiritu Reyes. Evaluación del software educativo.

además de que su soporte tecnológico seguramente corresponderá hasta lo que en ese momento estaba en el mercado. Así, los programas computacionales son elaborados y diseñados con lógicas y objetivos propios, lo que ha permitido hacer diversas clasificaciones de ellos, de acuerdo con las características que presentan ¹⁹.

Aspectos a tenerse en cuenta al momento de desarrollar un software educativo.

Como vemos, al momento de desarrollarse un software educativo se deben considerar ciertos aspectos tales como pedagógicos, técnicos e informáticos o tecnológicos, puesto que de entre los múltiples programas que encontramos en el internet o por otros medios, algunos no tienen los aspectos anteriormente mencionados y simplemente actúan como modo de visualización de resultados, por lo que un software educativo debe ser más flexible y no tan resumido en su presentación y muestra de resultados.

De acuerdo a los aspectos tenemos lo siguiente:

Tabla 3: Aspectos pedagógicos del software educativo.

ASPECTO PEDAGOGICO	
CARACTERISTICA	INDICADOR
Objetivo	Cumplimiento de los objetivos
Forma de presentación de la información	Imágenes, lecturas, videos, juegos, animaciones
Contenido	Calidad del material, actualidad.
Estructura del contenido	Presentación o secuencia, organización.
Interacción	Capacidad de motivación, ambiente amigable, interactividad.
Evaluación	Evaluación adecuada, preguntas y ejercicios

¹⁹ Ysauro González Neri, Victoria Carmona Martínez y Sara Espiritu Reyes. Evaluación del software educativo.

	de motivación.
ASPECTO AREA TECNOLOGICA	
CARACTERISTICA	INDICADOR
Naturaleza y evolución	Proceso, estructura, sistema, componentes, recursos, optimización, función.
Apropiación y uso de la tecnología	Artefactos, procesos, sistemas, productos.
Solución de problemas	Manejo de estrategias, identificación, formulación, solución.
Tecnología y sociedad.	Actitud del estudiante, valoración social del estudiante, participación social.
ASPECTO TECNICO	
Instalación	Facilidad de instalación, requerimientos adicionales de software y hardware, tamaño de la aplicación, portabilidad, conectividad.
Navegación	Facilidad de navegación, acceso a las actividades, velocidad de acceso, secuencia para acceso al contenido.
Diseño	Calidad y estética de los elementos multimedia, texto, tamaño y color de letra, manual de usuario, sonido.
Fiabilidad	Tolerancia a fallas, recuperación de datos, grabar sesión de forma automática.

9.2.2 Fases de la ingeniería del software.

El trabajo que se asocia a la ingeniería del software se puede dividir en tres fases, con independencia del área de aplicación, tamaño o complejidad del proyecto ²⁰.

²⁰ Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

Fase de definición: Se centra sobre el **qué**. Identificar qué información ha de ser procesada, que función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Identificar los requisitos del sistema y del software.

Las tareas específicas de esta fase son:

- ✓ Ingeniería de Sistemas o de información.
- ✓ Planificación del proyecto software.
- ✓ Análisis de requerimientos.

Fase de desarrollo: Se centra en el **cómo**. Definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura de software, cómo ha de implementarse los detalles procedimentales, cómo han de caracterizarse interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo ha de realizarse la prueba.

Las tareas específicas de esta fase son:

- Diseño del software
- Generación de código
- Prueba del software.

Fase de mantenimiento: Se centra en el cambio y monitoreo del software diseñado.

- Corrección de errores.
- Adaptaciones requeridas a medida que evoluciona el entorno del software.
- Cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. Se

encuentran cuatro tipos de cambio:

- Corrección.
- Adaptación.
- Mejora.
- Prevención.

9.2.3 Modelos para el diseño y proceso del software.

Para el modelo y diseño del software, es importante tener en cuenta que se deben seguir ciertos procesos o fases, y además se debe disponer de herramientas y modelos a seguir para la creación, ejecución, evaluación y mantenimiento del software. Los métodos o modelos de diseño de software permiten tener un plan de inicio y final, y en cada una de estas etapas saber qué es lo que se va a hacer o que es lo que se debe hacer en cada proceso. Entre estos modelos encontramos los tradicionales, algunos más efectivos y fáciles de usar que otros, y modelos actuales que han permitido el desarrollo de programas con más eficiencia y calidad, donde se han llevado a cabo procesos más cuidadosos, y se han controlado con mayor eficacia las fallas y problemas. A continuación, vemos algunos de los siguientes modelos:

El modelo lineal secuencial ²¹

Llamado algunas veces “ciclo de vida básico” o “modelo en cascada”, el modelo lineal secuencial sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

Es un ciclo de vida en sentido amplio, que incluye no sólo las etapas de ingeniería sino toda la vida del producto: las pruebas, el uso (la vida útil del software) y el mantenimiento.

²¹ Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

Etapas:

Ingeniería del software: Análisis de las características y el comportamiento del sistema del cual el software va a formar parte.

Para un sistema nuevo: Se debe analizar cuáles son los requisitos funciones del sistema, y luego asignar un subconjunto de estos requisitos y funciones al software.

Para un sistema ya existente: se debe analizar el funcionamiento de la organización y sus operaciones y se asigna al software aquellas funciones que se van a automatizar. Está formado por diagramas y por descripciones en lenguaje natural.

Análisis: Se debe comprender cuáles son los datos que se van a manejar, cuál va a ser la función que tiene que cumplir el software, cuáles son las interfaces requeridas y cuál es el rendimiento y otros requisitos no funcionales que se esperan lograr.

Los requisitos, tanto del sistema como del software deben documentarse y revisarse con el cliente. Como resultado de la fase de análisis, se obtiene la especificación de requisitos del software. También está formado por diagramas y descripciones en lenguaje natural.

Diseño: El diseño se aplica a cuatro características distintas del software: la estructura de los datos, la arquitectura de las aplicaciones, la estructura interna de los programas y las interfaces.

El diseño es el proceso que traduce los requisitos en una representación del software de forma que pueda conocerse la arquitectura, funcionalidad e incluso la calidad del mismo antes de comenzar la codificación. En el diseño, los requisitos del software se traducen a una serie de

diagramas que representan la estructura del sistema software, de sus datos, de sus programas y de sus interfaces.

Codificación: Consiste en la traducción del diseño a un formato que sea comprensible para la máquina. Si el diseño es lo suficientemente detallado, la codificación es relativamente sencilla, y puede hacerse de forma automática, usando generadores de código.

Se traducen los diagramas de diseño a un lenguaje fuente, que luego se traduce - se compila - para obtener un programa ejecutable.

Prueba: El objetivo es comprobar que no se hayan producido errores en alguna de las fases anteriores, especialmente en la codificación. Se deben probar todas las sentencias, y todos los módulos que forman parte del sistema.

Utilización: El software se entrega al cliente y comienza la vida útil del mismo.

Mantenimiento: El software sufrirá cambios a lo largo de su vida útil. Estos cambios pueden ser debidos a tres causas:

- ❖ Que, durante la utilización, el cliente detecte errores en el software: los errores latentes.
- ❖ Que se produzcan cambios en alguno de los componentes del sistema.
- ❖ Que el cliente requiera modificaciones funcionales no contempladas en el proyecto.

El modelo Prototipado.

Comienza con la recolección de requisitos.

Cliente y desarrolladores definen los objetivos globales del software.

Además, identifican los requisitos conocidos y aquellos que deben ser más definidos.

Aparece un diseño rápido centrado en los aspectos visibles para el cliente (e.g. información de E/S)

El diseño rápido lleva a la construcción de un prototipo.

El prototipo lo evalúa el cliente y lo utiliza para refinar los requisitos.

El proceso se itera.

Modelos evolutivos:

Características:

Gestionan bien la naturaleza evolutiva del software.

Son iterativos: construyen versiones de software cada vez más completas.

Se adaptan bien:

Los cambios de requisitos del producto.

Fechas de entrega estrictas poco realistas.

Especificaciones parciales del producto.

Programación Extrema (XP).²²

Kent Beck, el padre de XP, describe la filosofía de XP como una metodología ágil que potencia las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el

²² Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y

disposición para enfrentar los cambios. XP se define como adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

El modelo DRA (Desarrollo Rápido de Aplicaciones) ²³.

El proceso DRA permite al equipo de desarrollo crear un “sistema completamente funcional” dentro de periodos cortos de tiempo (de 60 a 90 días). El enfoque DRA comprende las siguientes fases:

Modelado de gestión: El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce al proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?

Modelado de datos: Conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (atributos) de cada uno de los objetos y las relaciones entre estos objetos.

Modelado del proceso: Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir o recuperar un objeto de datos.

²³ Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012

Generación de aplicaciones: El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes o crear componentes reutilizables.

Pruebas y entregas: Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

9.2.4 Algunos softwares de matemáticas.

actualmente vemos una gran variedad de software educativo, en especial en el área de matemáticas y para diferentes niveles, los hay softwares educativos usados para escritorio y para diferentes sistemas operativos, por nombrar algunos ya sea en Windows, Linux entre otros, esto de acuerdo a su configuración y compatibilidad, como también hay aplicaciones de matemáticas en línea (on-line), que resuelven los problemas y dan los resultados en línea. Algunos programas más entendibles que otros, más flexibles, más didácticos e interactivos, otros los hay más resumidos en la muestra de resultados, esto de acuerdo al diseñador y al equipo de intervención en el desarrollo del software.

Los programas o softwares educativos se clasifican en diferentes categorías de acuerdo a sus funciones, sin embargo, un software educativo, en este caso de matemáticas, no está dentro de una categoría, sino que puede pertenecer a varias categorías puesto que abarca funcionalidades de otras categorías.

Hay programas educativos de matemáticas para graficas de funciones, para algebra, algebra lineal, para geometría, análisis, para estadística y probabilidad, para usos generales, es decir que tienen varias funcionalidades, también encontramos para programación lineal, para estadística y probabilidad, para trigonometría.

A continuación, vemos algunos ejemplos de softwares de matemáticas:

Cube test ²⁴: CubeTest sirve para practicar la visión espacial, mediante un sencillo juego que consiste en responder 10 preguntas tipo test. En cada una de estas preguntas se muestra un cubo con las posibles vistas del mismo cubo, pero sólo una es correcta. Este programa está dirigido a estudiantes de la escuela primaria o a profesores que deseen proponer este reto a sus alumnos.

Tux of Math Command ²⁵: es un juego educativo destinado a los niños que están aprendiendo las operaciones básicas matemáticas, tales como sumar, restar, multiplicar o dividir. Este sencillo juego consiste en que las operaciones van cayendo desde arriba y el jugador debe destruirlas antes de que lleguen al suelo escribiendo el resultado.

TuxMathScrabble ²⁶: es un juego matemático parecido al Scrabble, pero con números. Con él podrás practicar las operaciones elementales y dispone de niveles distintos para perfeccionar tu habilidad y conocimientos matemáticos. Para instalarlo es necesario elegir como carpeta de instalación "C:/ProgramFiles/TuxMathScrabble" para que funcione.

Derive: es un programa de matemáticas para ordenador. Procesa variables, expresiones, ecuaciones, funciones, vectores y matrices al igual que una calculadora científica sirve para

²⁴ <http://www.vandenoever.info/software/cubetest/>

²⁵ <http://www.newbreedsoftware.com/tuxmath/>

²⁶ <http://new.asymptopia.org/>

trabajar con números. DERIVE puede realizar cálculos numéricos y simbólicos, con álgebra, trigonometría y análisis, además de representaciones gráficas en dos y en tres dimensiones. El aspecto más sobresaliente de DERIVE es su trabajo simbólico unido a sus capacidades gráficas.

Características principales

- **Álgebra:** desarrollo y factorización de polinomios; simplificación de expresiones algebraicas; resolución de numérica y simbólica; resolución de sistemas lineales de ecuaciones...
- **Aritmética:** aritmética exacta y aritmética aproximada de precisión configurable; factorización de enteros; conversión de unidades métricas; calculadora científica, números complejos
- **Gráficos 2D:** en forma explícita, implícitas y paramétricos; coordenadas rectangulares y polares; funciones de variable compleja; especificación de colores; permite poner etiquetas de ejes y anotaciones sobre los gráficos...
- **Gráficos 3D:** mallado para funciones de dos variables; selección del punto de vista; cambio de escala; rotación de gráficos en tiempo real...
- **Cálculo:** cálculo simbólico de límites finitos e infinitos; primera y n-ésima derivadas; integrales definidas e indefinidas; integración numérica; sumas y productos finitos e infinitos; derivación implícita y paramétrica; desarrollos de Taylor y series de Fourier; longitud de arco, áreas y volúmenes.

Máxima: Es un programa para la manipulación de expresiones simbólicas y numéricas, incluyendo diferenciación, integración, expansión en series de Taylor, transformadas de Laplace,

ecuaciones diferenciales ordinarias, sistemas de ecuaciones lineales, y vectores, matrices y tensores.

Maple: es un programa matemático de propósito general capaz de realizar cálculos simbólicos, algebraicos y de álgebra computacional. Este software puede usarse como una calculadora numérica y simbólica.

Matlab: es un programa diseñado para el ámbito de las matemáticas, y también usado como lenguaje de programación, y que ofrece un entorno integrado. Entre sus funciones están cálculos matemáticos, estadísticos, algebra simbólica, entre otras funciones.

Geogebra: Software de matemáticas, libre, para enseñar y aprender. Gráficos interactivos, álgebra y planillas dinámicas para todos los niveles, desde el básico escolar al universitario. Lo más importante de GeoGebra es la interactividad; una vez construida una figura se puede mover cualquiera de los objetos independientes que la forman y automáticamente se modifican todos los que dependen de él.

Cmaptools: herramienta para la creación de mapas conceptuales interactivos. permite compartir los trabajos en red.

9.2.5 Terminología:

Hardware: se puede definir como la parte física del computador o sistema de cómputo, aquello que se puede tocar y manipular físicamente, aquí encontramos los dispositivos de entrada, de salida, de almacenamiento, de procesamiento. Un ejemplo de ello es el monitor, aquí veremos la aplicación informática a desarrollar.

Software: se puede definir como un conjunto de instrucciones lógicas, que no se pueden tocar, y que realizan un determinado proceso, que en conjunto con el hardware hace el

funcionamiento total de una computadora, también se entiende como una aplicación, y podemos ver diferentes tipos de software: software educativo, software de aplicación, software de programación, software de sistema, software de diagnóstico, software de gestión, software de ingeniería, software de utilidad.

Software privativo: es el software cuya licencia de uso está sujeta a las leyes de propiedad intelectual y por lo tanto sólo permite su uso y aún en forma restringida, reservando el resto de los derechos para el propietario del copyright o derechos de autor.²⁷

Software libre: programa informático de código abierto, cuya licencia nos permite hacer uso libre del programa y nos permite: copiarlo, distribuirlo, estudiarlo y modificarlo.

Software libre educativo: paquete informático distribuido con licencia de código abierto o software libre que tiene una aplicabilidad, ya sea para apoyar la gestión de los contenidos educativos (componente pedagógico), ya sea para facilitar el seguimiento del proceso educativo (componente administrativo).²⁸

Software educativo: se define como un programa informático cuyo objetivo es la enseñanza didáctica educativa, que puede presentarse en diferentes visiones o funcionalidades.

Software educativo de matemáticas: se define como un programa o conjunto de programas informáticos (software) diseñados especialmente para el área de matemáticas, que pueden ser de diferentes niveles y de diferentes temáticas.

²⁷ Prof. Ricardo P. Salvador. Software.

²⁸ Juan Cristobal Cobo. Conocimiento, creatividad y software libre.

Aplicativo: se puede definir como un programa (software) interactivo que puede contener una serie de recursos visuales y dinámicos, y donde es usado para una determinada área de enseñanza.

Multimedia interactivo: aplicación donde el usuario participa de forma interactiva, por lo que el control del conocimiento está en sus manos. Aun así, es la presencia del elemento “tutor” lo que convierte un sistema multimedia en verdaderamente formativo (SALINAS, 1996).

Software interactivo: se define como un programa que contiene una visualización dinámica e interactiva, como imágenes, animaciones, videos, sonidos, transiciones, imágenes en movimiento.

Microsoft Visual Basic: Se puede definir como un lenguaje de programación de alto nivel orientado a objetos, donde presenta una interfaz amigable, usado para desarrollar múltiples aplicaciones desde programas sencillos hasta programas complejos en diferentes campos. Es un lenguaje orientado a objetos y en las últimas versiones ha incorporado el .NET FRAMEWORK.

Lenguaje de programación: Es un software que se emplea como lenguaje artificial y que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Se compone de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas.²⁹

Algunos ejemplos de ello son: C, C++, C#, Visual Basic, Java, Lisp.

Matemáticas: es una ciencia formal y exacta basada en el estudio de entes abstractos, que pueden entenderse como el manejo de números, figuras geométricas, símbolos entre otros

²⁹ Módulo de Herramientas informáticas (2013). Lenguajes de programación (pág. 20)

aspectos donde la presencia de los números y planteamientos matemáticos son la explicación de cualquier evento.

Geometría: es una rama de las matemáticas que estudia las propiedades inalterables de las figuras, donde se ven la geometría plana y la geometría del espacio.

Trigonometría: es una rama de las matemáticas que estudia las relaciones que pueden existir entre los elementos y ángulos de un triángulo. Donde vemos las funciones trigonométricas, las resoluciones de los triángulos rectángulos y no rectángulos, las aplicaciones de la trigonometría.

Algebra: es una rama de las matemáticas que estudia las cantidades considerables del modo más general posible, donde las cantidades se representan mediante números y letras, a las que denominamos variables.

Discriminante: es la representación numérica de $b^2 - 4ac$ deducida de la ecuación cuadrática $ax^2 + bx + c = 0$

Ecuación cuadrática: o ecuación de segundo grado. Es una ecuación donde el mayor exponente es dos. $ax^2 + bx + c = 0$, $x^2 + bx + c = 0$

Función: es una regla que relaciona dos o más variables.

Función lineal: es una ecuación de primer grado donde la línea es una recta cuando se la representa gráficamente. $y = ax + x + c$, $y = x + c$

Función cuadrática: es una ecuación de segundo grado donde la gráfica de la función es una parábola. Es de la forma $y = ax^2 + bx + c$, $y = x^2 + bx + c$

Ecuación lineal: o llamada ecuación de primer grado, es una ecuación donde el mayor exponente es 1. $y = ax^2 + bx + c$, $y = x^2 + bx + c$

Ecuación: es una igualdad en la que hay una o varias cantidades desconocidas o incógnitas y que solo se verifica dando valores a las incógnitas.

Resolver una ecuación: es hallar las soluciones mediante operaciones algebraicas.

10 Marco Contextual

La Institución Educativa Rural José Asunción Silva, se encuentra ubicada en una zona rural, en la Inspección el Placer, Municipio Valle del Guamuez, Departamento del Putumayo.

Dicha ubicación hace que la metodología de enseñanza sea un poco más tradicional que moderna, debido a que la inversión en el ámbito de la educación va a paso lento, donde en temas de matemáticas, no se cuenta con material didáctico interactivo, más nuevos, donde se use las TIC's, solo cuenta con libros de enseñanza, una biblioteca no tan modernizada, y en cuanto a tecnología, solo cuenta con unos pocos computadores, y ante esto, la implementación del uso del software libre en temas matemáticos, no se ha dado, los profesores de matemáticas mantienen una enseñanza tradicional, apoyados solamente en los libros, lo que muchas veces no es suficiente para el aprendizaje de los estudiantes, ya que no todos se pueden acoplar a este método de enseñanza.

Sin embargo, en la institución (IERJAS) se han realizado capacitaciones, simulacros, en torno al área de matemáticas, los profesores han optado por usar métodos más efectivos para aumentar el aprendizaje en los estudiantes en el área de matemáticas. Y a pesar de que la metodología de enseñanza se centra en el docente, los estudiantes de grado noveno en su totalidad logran aprobar la materia de matemáticas, donde se resalta mucho las metodologías y

estrategias llevadas a cabo por el docente para que los alumnos logren un aprendizaje asertivo. Ahora bien, el propósito de este proyecto y del desarrollo del software es complementar el aprendizaje del estudiante en su etapa formativa en el área de matemáticas, que incentive su aprendizaje de forma activa e interactiva, que se relacione más con la tecnología informática o las TIC's, que pueda aprender no solo en la escuela sino también en casa o en un lugar adecuado, promoviendo así el aprendizaje autónomo.

Ubicación de la problemática y forma de solución

Una vez planteado la conceptualización, la información y recursos teóricos necesarios para tener en cuenta al momento del diseño del software educativo en el área de matemáticas, analizamos la situación problema y la solución correspondiente. Tras el planteamiento del problema, la búsqueda de información para tratar el problema y la alternativa de solución, se ha ubicado principalmente la problemática en el área de tecnología informática, es decir en el uso de las TIC's, donde se hace énfasis en el sector del software educativo, donde el centro educativo no cuenta con un medio complementario para la enseñanza de las matemáticas, que es el uso del software educativo. Y como anteriormente se ha mencionado, la alternativa de solución, es decir la propuesta planteada, que es el diseño de un software educativo en el área de matemáticas, contribuye a la solución del problema a través de su implementación y uso, además de complementar el aprendizaje de los estudiantes de grado noveno en el área de matemáticas, y con esto implementar el software educativo en el centro educativo, y así se pretende dar solución a la problemática de forma realista.

11 Metodología.

11.1 Línea de Investigación.

Para el desarrollo de este proyecto aplicado se hizo énfasis en el proyecto de desarrollo tecnológico (**investigación tecnológica aplicada**), (que es una actividad donde se opta por la creación o diseño de un nuevo producto o proceso, pruebas experimentales o ensayos, elaboración de prototipos previos a su explotación comercial o industrial), donde está centrado en el diseño o creación de un nuevo software tipo educativo en el área de matemáticas, licencia libre, especialmente para grado noveno.

La tecnología permite aplicar el conocimiento para diseñar proyectos que innoven, ayuden o beneficien a otros en determinados temas y motive a la vez en la creación de nuevos productos. De tipo libre, con el fin de que sea analizado por otras personas y lo puedan mejorar.

11.2 Método de Investigación.

Para el desarrollo del presente proyecto se ha implementado los siguientes métodos de investigación:

Exploratoria: Indagar acerca de una realidad poco estudiada: es decir, sondear, explorar descubrir posibilidades. En el método exploratorio el investigador intenta, en una primera aproximación, detectar variables, relaciones y condiciones en las que se da el fenómeno en el que está interesado. En otros términos, trata de encontrar indicadores que puedan servir para definir con mayor certeza un fenómeno o evento, desconocido o poco estudiado. Se utiliza cuando el tema a investigar es poco conocido, o ha sido poco o nada estudiado. Punto de partida para realizar estudios de mayor profundidad.

De acuerdo a lo anterior, este método permite explorar las condiciones del centro educativo, detectar variables, explorar el área a trabajar, explorar la problemática, identificar aspectos básicos que servirán para la implementación del proyecto. Y con lo anterior plantear la solución eficaz. Con lo anterior se busca implementar una solución tecnología a la situación actual de la problemática del I.E.R.A.S.

Descriptiva: descripción de las características de un conjunto de sujetos. Por otra parte, se trata de describir las características más importantes de un determinado objeto de estudio con respecto a su aparición y comportamiento, o simplemente el investigador buscará describir las maneras o formas en que éste se parece o diferencia de él mismo en otra situación o contexto dado.

Con este método llevado a cabo se podrá describir los eventos y características más relevantes que se dan durante la exploración, y aquí se muestran datos mediante técnicas cualitativas o muestras numéricas de los análisis, tanto en primera fase de inicio como en la fase final del proyecto.

Donde se tendrá en cuenta la investigación cuantitativa y cualitativa, en la primera, donde mediante el uso de técnicas y métodos de investigación, se denotará mediante resultados de análisis de forma numérica, en la segunda abarca lo que es la descripción y muestra de resultados donde se hace de manera teórica.

11. 3 Técnicas de Análisis de la Información.

Para la recolección de la información se manejará previamente lo que son las fuentes de información primaria y secundaria, y las técnicas de recolección de la información.

Fuentes de información.

Fuente primaria: son aquellas fuentes donde la información se obtiene de forma directa y la información es convincente puesto que se obtiene de primera mano. La información se obtiene mediante el acercamiento con las personas, organizaciones, acontecimientos, lugares.

Aquí las fuentes serán el docente del área de matemáticas del grado noveno, docentes de informática, los alumnos del grado noveno, el ambiente de enseñanza, el área de informática.

Fuente secundaria: son fuentes que ofrecen información sobre lo que se están investigando o se va a investigar, y que estas fuentes no provienen de la fuente principal u original. Aquí encontramos lo que son los libros, las revistas, artículos, documentos, documentales y otros medios de información.

Aquí las fuentes de tipo secundario serán los documentos de apoyo relacionados con la temática a plantear o proyecto a diseñar, información en internet y medios físicos sobre el tema del software educativo, su uso en la educación, en los centros educativos, información sobre el desconocimiento del software libre educativo, la falta de uso del software educativo.

Técnicas de recolección de la información.

Para recolectar la información se usarán las siguientes técnicas:

Observación: La observación es un acto en el que entran en una estrecha y simultánea relación el observador (sujeto) y el objeto; dependiendo del tipo de investigación el objeto tomaría el lugar del sujeto(s) observable(s). Este método tiene como principal ventaja, que los datos se recogen directamente de los objetos o fenómenos percibidos mediante registros caracterizados por la sistematicidad de la recolección y por la maleabilidad de las condiciones en que se proyecta realizarla.

Con esta técnica se manejará lo que son registros sobre el número de estudiantes que hay en el grado noveno, registros de inventarios de equipos de cómputo en el área de informática. La observación consiste en mirar detenidamente las particularidades del objeto de estudio para cuantificarlas.

Entrevista: Se utiliza para recabar información en forma verbal, a través de preguntas que propone el investigador o entrevistador. Consiste en una conversación entre una o más personas en la cual uno es el entrevistador y el otro u otros son los entrevistados o informantes clave.

Con esta técnica se interactuará con el docente, alumnos, donde mediante la técnica se diseñará unas preguntas estructuradas.

Encuesta: Es una técnica para obtener información, generalmente de una muestra de SUJETOS. La información es recogida usando procedimientos estandarizados de manera que a cada individuo se les hace la misma pregunta en más o menos la misma manera.

Aquí se usará lo que son preguntas con opciones múltiples, cuestionarios.

11. 4 Análisis y muestra de resultados.

Luego de recolectar la información y posteriormente tratarla, los resultados se mostrarán mediante graficas estadísticas, donde se interpretarán los resultados, y mediante conclusiones las que no manejen análisis numéricos.

11. 4.1 Fase de exploración y análisis.

Entrando en ejecución del proyecto, y realizado la exploración del caso, se constata que la enseñanza del área de matemáticas en el grado noveno del colegio I.E.R.J.A.S. (ubicado en la Inspección El Placer, Municipio Valle del Guamuez – Putumayo), es especialmente de docente a

estudiante, donde el docente usa su propia metodología de enseñanza y su herramienta es el libro de matemáticas de grado noveno de acuerdo a la editorial y actualización, y también se apoya en otros libros y fuentes de información referentes al área de matemáticas de grado noveno para así hacer más didáctica su enseñanza.

Ante esto, se observó que el docente no hace uso de algún software de matemáticas específico para grado noveno, no hace uso de las tecnologías informáticas o de las TIC's, no hace uso del software educativo en el área de matemáticas. Además, el colegio cuenta con sala de informática donde tiene 16 computadores, tienen sistema operativo Windows 8. de igual forma se constató que la metodología usada por parte del docente, para enseñar matemáticas, es buena ya que como el docente lo indico, en sus años de profesor en esta área, los estudiantes han aprobado el área de matemáticas, aunque cabe notar que algunos estudiantes logran pasar el curso con más facilidad que otros, el aprendizaje es variado, y por ende el docente tiene que emplear más horas o tiempo para que los estudiantes menos ágiles puedan comprender las temáticas y así puedan aprobar la materia. El docente del área de matemáticas de grado noveno (Henry Córdoba), dio a conocer que en el área de matemáticas se requiere del uso de estrategias y metodologías efectivas, ya sean propias o a través de fuentes externas, para que los alumnos puedan captar la enseñanza y los temas, puesto que las matemáticas es un área donde se requiere de total atención y dedicación, de total entrega, donde hay que practicar mucho para aprender, y que el estudiante debe dar de sí lo que tiene a su disposición y esforzarse por aprender. Como es visto, da a conocer que hay estudiantes con gran intelecto y que les gusta las matemáticas, y tienen una gran capacidad para aprender matemáticas, estos estudiantes son los que captan los temas más rápidos, de igual forma también da a conocer que hay estudiantes que tienen poco gusto hacia las matemáticas o que tienen una capacidad de aprendizaje más lenta en esta área.

Sobre el uso del software libre educativo, dio a conocer que no se tenía mucho en cuenta este factor, no había orientación efectiva ni motivaciones o iniciativas que conlleven al uso del software educativo. Además, el colegio como tal aún no ha potencializado o implementado la inserción de esta herramienta educativa como complemento al aprendizaje. Por otra parte, el no uso del software libre educativo en el área de matemáticas se debe quizás a la falta de investigación, orientación, un desconocimiento. Además, expone que el uso del software educativo podría de manera complementaria potencializar aún más el aprendizaje de los estudiantes, y sobre todo de aquellos que tienen un ritmo regular en su aprendizaje, ya que podrían usar el programa tanto en el colegio como en casa, o en tiempos libres. Con lo anterior, el docente además expuso lo importante que es el usar la tecnología informática en el sector educativo, el usar las TIC's puede incentivar a los estudiantes a hallarle gusto por las matemáticas, donde las herramientas didácticas e interactivas pueden potencializar a un más el aprendizaje del alumno, ya que los jóvenes tiene un acercamiento más activo con la computación, la informática, la internet, y esto les ayuda mucho en cuanto al uso del software libre educativo, lo interactivo les fascina más a los estudiantes, y las matemáticas requiere de este factor.

Por otra parte, se observó que por la ubicación del centro educativo hace que el uso de las TIC's no sea a gran escala, y ante esto no se conozca mucho sobre el software educativo o programas afines, esto porque el centro educativo está ubicado en zona rural y donde la impartición de enseñanza es docente – estudiante.

La información se obtiene más que todo, de primera mano, del docente, quien ha estado por muchos años en el colegio y ha estado enseñando matemáticas en grado noveno, y como él lo específica, seguirá en el centro educativo por otros años más, y quien es un factor importante

para recolectar información, ya que también describe su situación como docente y el grado de aprendizaje de los estudiantes de grado noveno, en el área de matemáticas. De igual forma se recabo información tanto del docente como de los estudiantes del grado noveno del presente año. Con lo anteriormente descrito y analizado, se pretende que el docente sea el agente donde repose el diseño del proyecto, sea quien lo ponga en práctica y use el software educativo en su enseñanza como complementación, lo implemente cada año, lo dé a conocer a cada grupo de alumnos nuevos que entran cada año al grado noveno.

También se manejó lo que es la observación del lugar, la situación actual, los comportamientos, los recursos con que se disponía.

11.4.2 Tratamiento de la información.

Con el docente se llevó a cabo una entrevista donde se trataron temas sobre el uso del software libre educativo, los efectos positivos que genera, la metodología de enseñanza, el grado de conocimiento de los estudiantes de grado noveno, en el área de matemáticas, sobre qué opina sobre usar el software educativo en el área de matemáticas. Los resultados lo vemos anteriormente en la descripción.

También se llevó a cabo una encuesta tipo cuestionario (de preguntas) aplicada a los estudiantes de grado noveno (un total de 30 alumnos). (**Ver anexo**).

Análisis de resultados actuales.

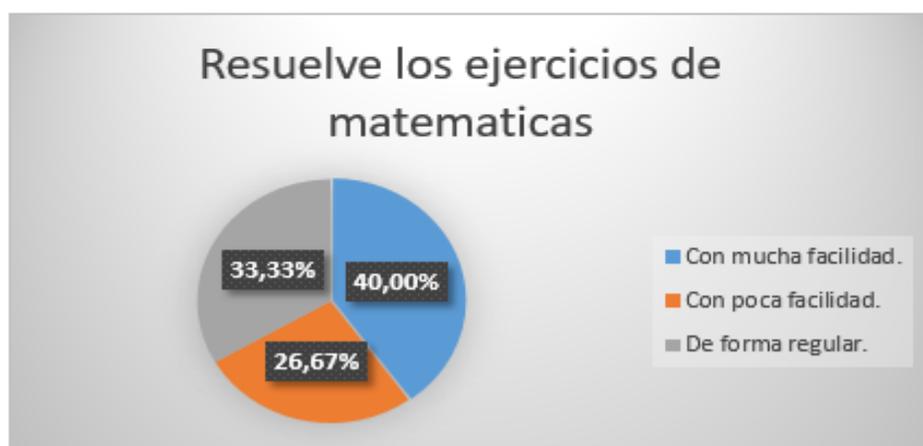
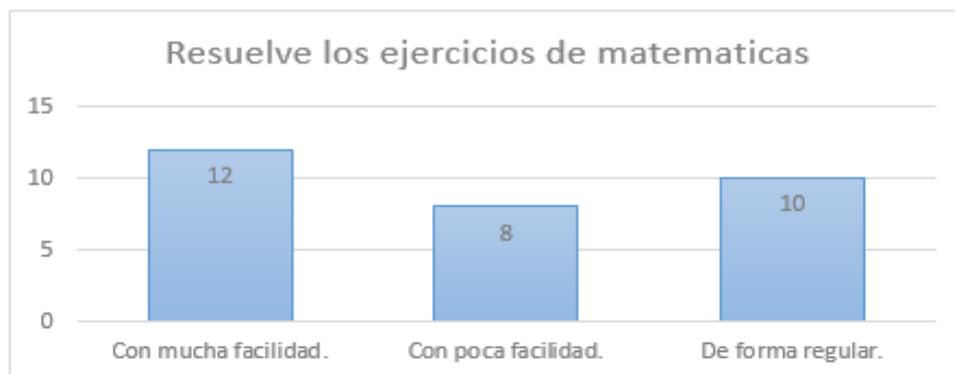


Figura 1: muestra de resultados de la pregunta N° 1. Mueses, J. (2017). Fuente: autoría propia.



Figura 2: muestra de resultados de la pregunta N° 2 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

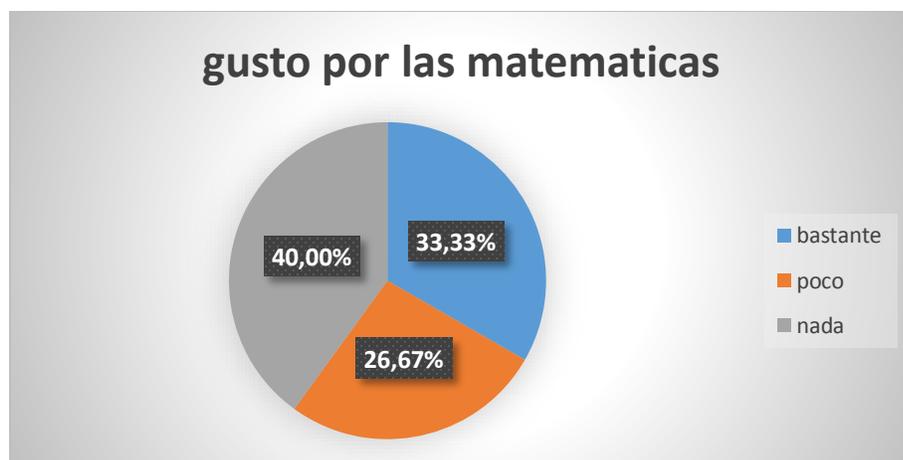


Figura 3: muestra de resultados de la pregunta N° 2 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

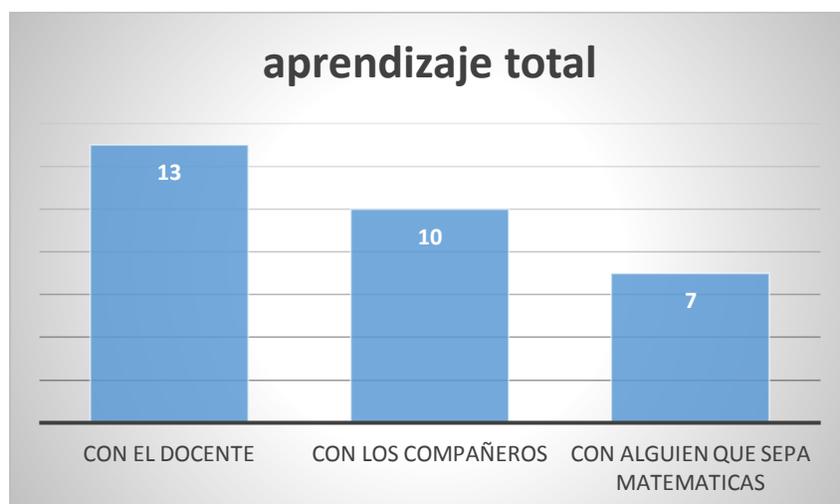


Figura 4: muestra de resultados de la pregunta N° 3 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

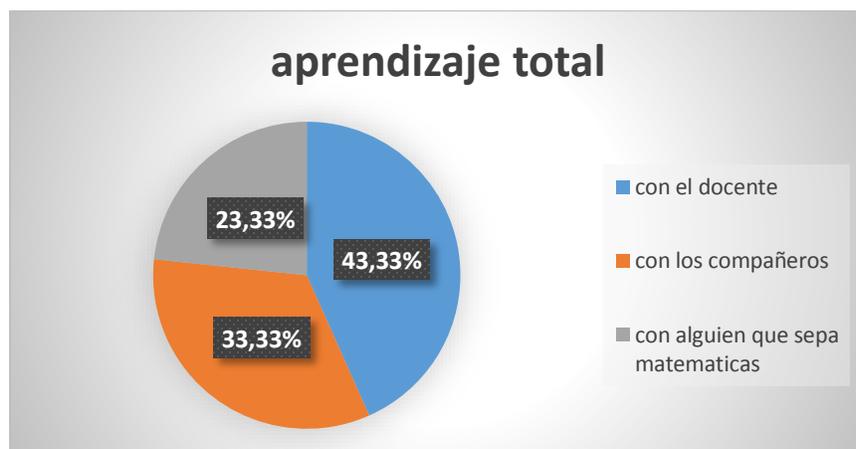


Figura 5: muestra de resultados de la pregunta N° 3 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

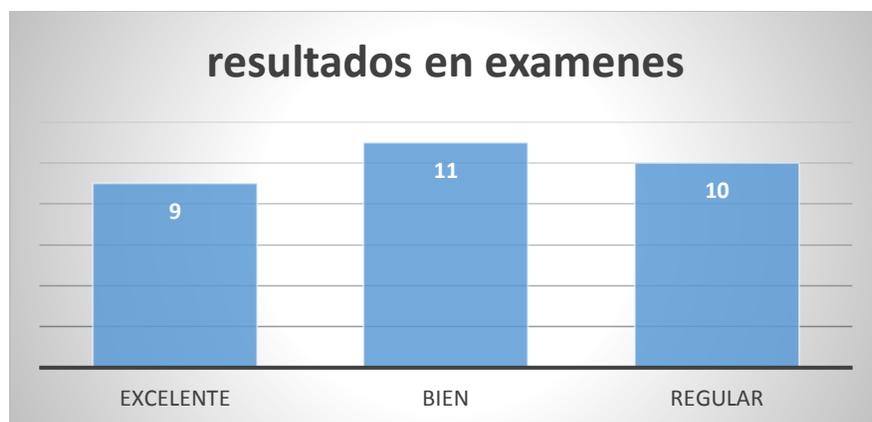


Figura 6: muestra de resultados de la pregunta N° 4 (gráfica de barras). Mueses, J. (2017). Fuente: autoría propia



Figura 7: muestra de resultados de la pregunta N° 4 (gráfica porcentual). Mueses, J. (2017). Fuente: autoría propia

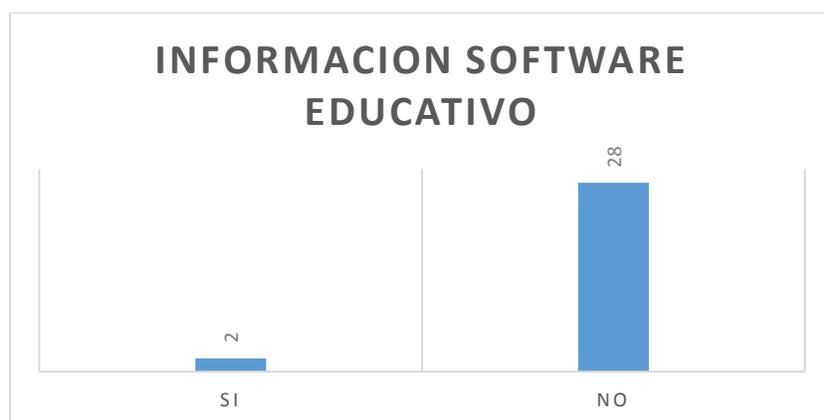


Figura 8: muestra de resultados de la pregunta N° 5 (gráfica de barras). Mueses, J. (2017). Fuente: autoría propia



Figura 9: muestra de resultados de la pregunta N° 5 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

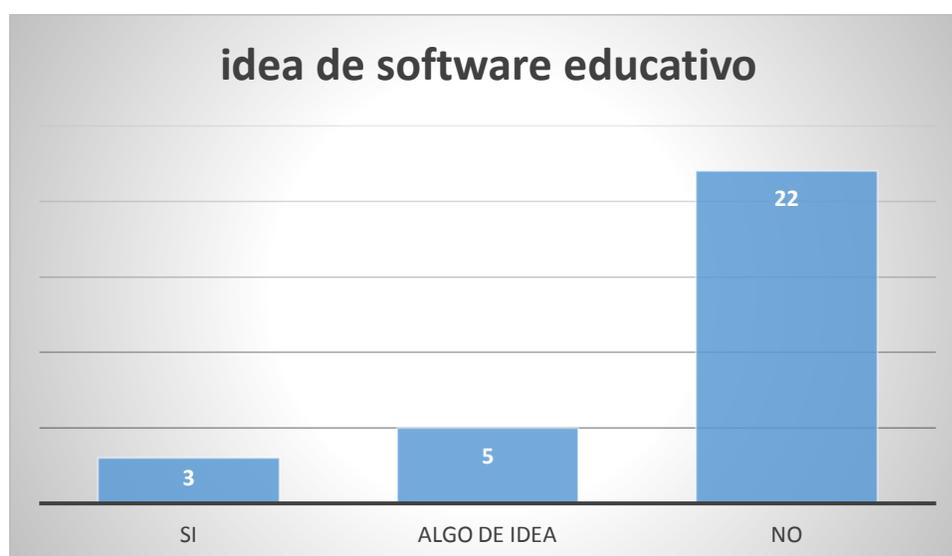


Figura 10: muestra de resultados de la pregunta N° 6 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

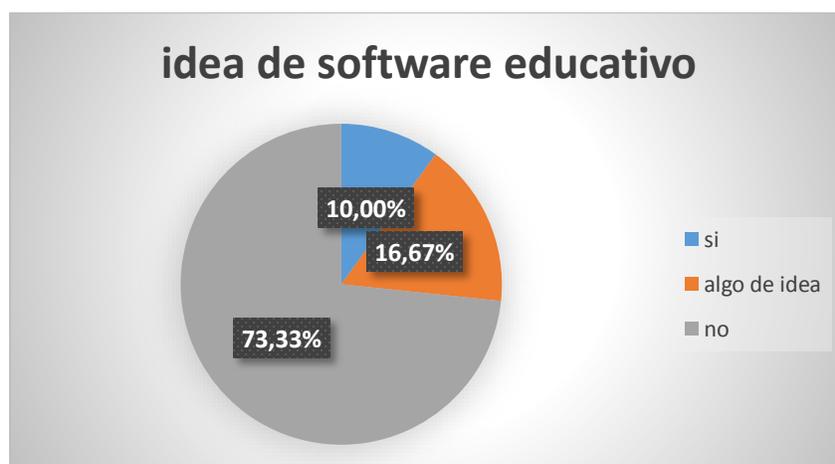


Figura 11: muestra de resultados de la pregunta N° 6 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

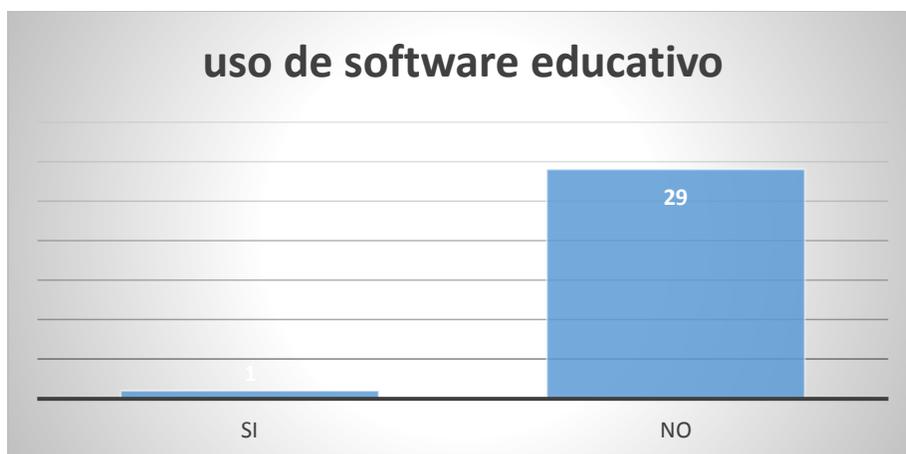


Figura 12: muestra de resultados de la pregunta N° 7 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

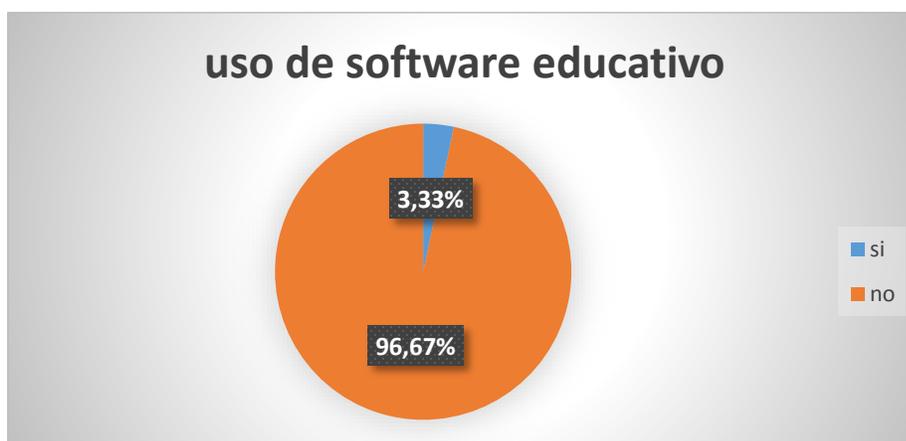


Figura 13: muestra de resultados de la pregunta N° 7 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

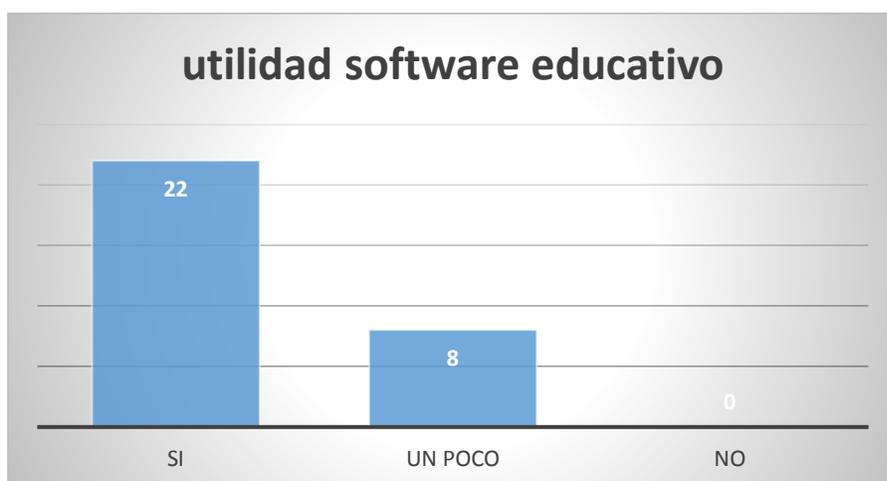


Figura 14: muestra de resultados de la pregunta N° 8 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

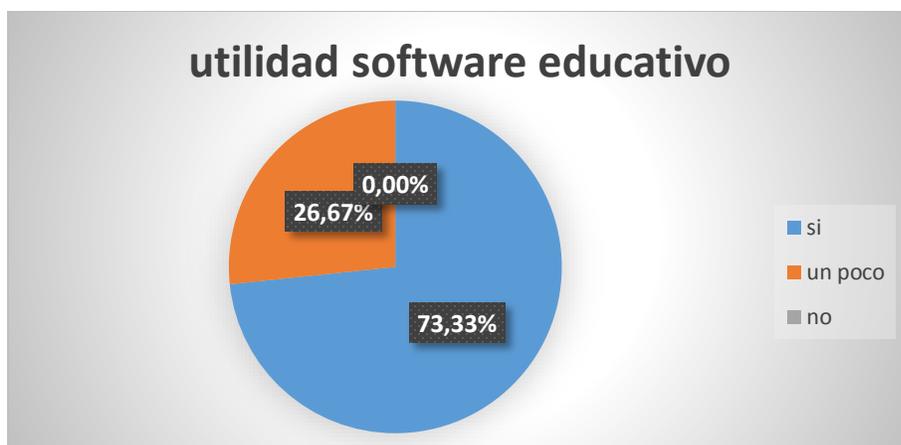


Figura 15: muestra de resultados de la pregunta N° 8 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

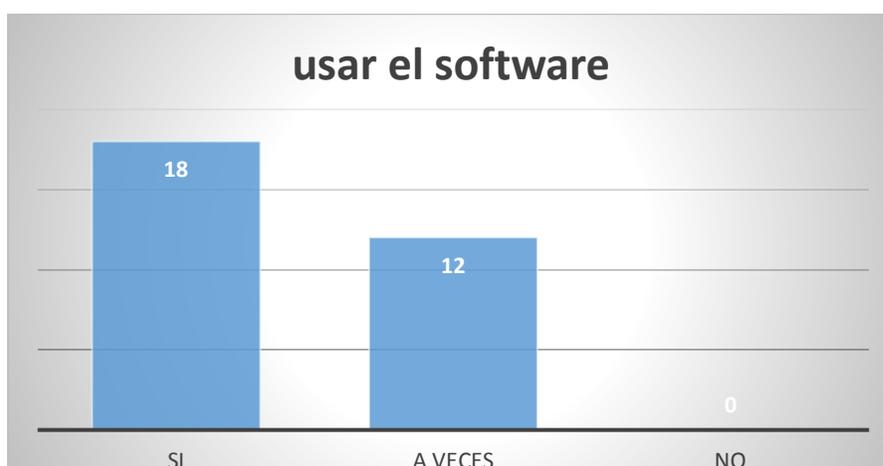


Figura 16: muestra de resultados de la pregunta N° 9 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

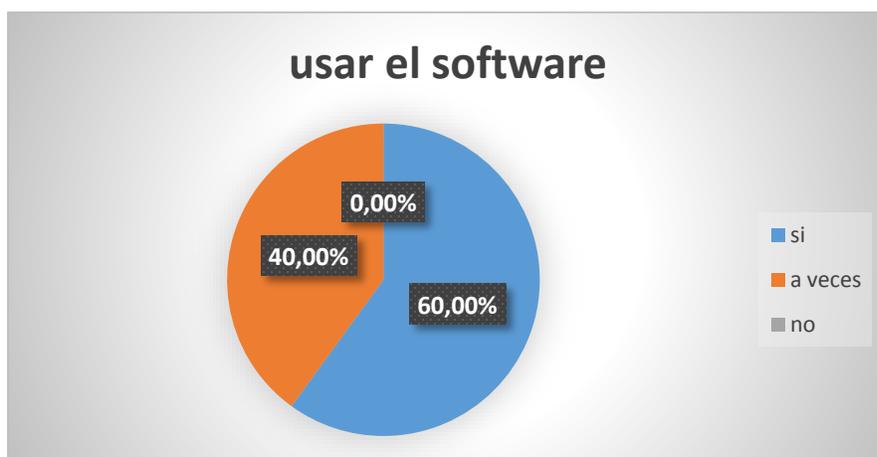


Figura 17: muestra de resultados de la pregunta N° 9 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia.



Figura 18: muestra de resultados de la pregunta N° 10 (grafica de barras). Mueses, J. (2017). Fuente: autoría propia

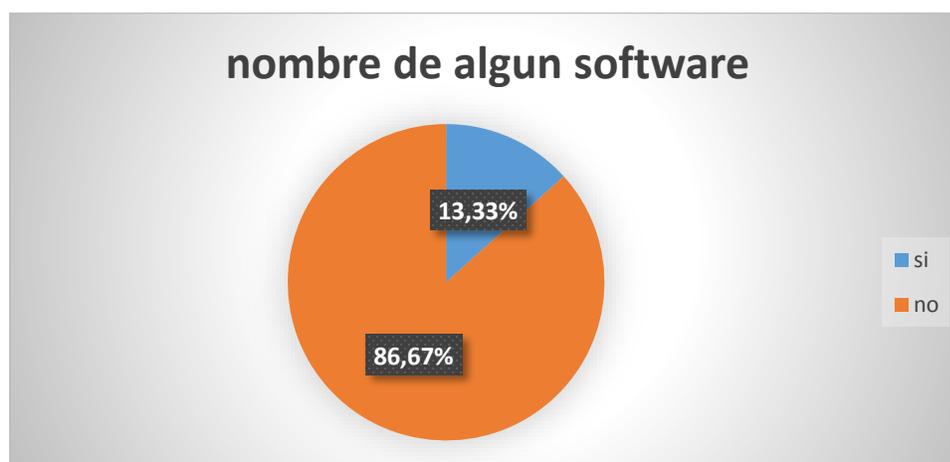


Figura 19: muestra de resultados de la pregunta N° 10 (grafica porcentual). Mueses, J. (2017). Fuente: autoría propia

Observando algunos datos en cuanto a la muestra de 30 alumnos del grado noveno, vemos que la mayoría de estudiantes es practico en la realización de los ejercicios, los entiende, y se esfuerzan por hacerlos de forma correcta. Aunque es de notar que muy pocos les gusta las matemáticas en su totalidad, donde las matemáticas para muchos es un área difícil de dominar.

Por otra parte, el aprendizaje lo complementan los alumnos apoyándose en otros estudiantes y de personas que sepan matemáticas y les pueden explicar. De igual forma, vemos que en los exámenes la mayoría los aprueba ya sea con notas altas o cercanas al 5.0, y que muy

pocos pasan con notas regulares. Además, en cuanto al tema del software educativo, se observa que no hay mucho conocimiento de ello, los estudiantes no tienen nociones sobre el uso del software educativo.

11. 5 Desarrollo del software.

En el desarrollo de software, independientemente del método de desarrollo usado, se llevan a cabo las siguientes fases:

Fase de definición: Se centra sobre el **qué**. Identificar qué información ha de ser procesada, que función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Identificar los requisitos del sistema y del software.

Las tareas específicas de esta fase son:

Ingeniería de Sistemas o de información

Planificación del proyecto software

Análisis de requerimientos³⁰

Fase de desarrollo: Se centra en el **cómo**. Definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura de software, cómo ha de implementarse los detalles procedimentales, cómo han de caracterizarse interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo ha de realizarse la prueba.³¹

Las tareas específicas de esta fase son:

³⁰ Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

³¹ Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

Diseño del software

Generación de código

Prueba del software.

Fase de mantenimiento:

Se centra en el **cambio**.

- Corrección de errores
- Adaptaciones requeridas a medida que evoluciona el entorno del software
- Cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente
- Se encuentran cuatro tipos de cambio:

Corrección

Adaptación

Mejora

Prevención³²

de igual manera en el desarrollo de software intervienen actores primarios y secundarios.

Actores primarios:

Analista de sistemas: es la persona encargada del estudio de la situación actual. Un analista de sistemas estudia los problemas y las necesidades de una empresa para determinar

³² Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

cómo podrían combinarse los recursos humanos, los procesos, los datos, las comunicaciones y la tecnología para obtener mejoras en la empresa.

Diseñador de sistemas: es la persona quien se en carga de diseñar, analizar e implementar nuevos sistemas automatizados según las necesidades de las áreas solicitantes, así como actualizar los ya existentes. Además, lleva a cabo los modelos funcionales, los diseños de prototipos y modelo de datos.

Programador: es la persona encargada de implementar componentes, localizar y corregir defectos. Asume la responsabilidad de implementar componentes y subsistemas de prueba, así como desarrollar artefactos de instalación. Además, es quien lleva a cabo la estructuración del prototipo a su nivel funcional.

Ingeniero de Software: es la persona que lleva a cabo la gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

Clientes: El cliente es un factor importante en el éxito de un proyecto, tanto como cualquier otro miembro del equipo, por eso es importante contar con la participación activa del cliente dentro del proyecto. Especifican los requerimientos para el desarrollo de software.

Usuarios: Interactuar con el sistema de información y diagnosticar su funcionamiento cuando se encuentre implementado. Interactúan con el software una vez que se liberan.

El diseñador gráfico: Es importante reconocer el papel del diseñador en un proyecto. Es bueno tener alguien encargado de la disposición general de una aplicación. Esto puede ir desde el diseño completo de la interfaz de usuario, hasta el definir sólo algunas directrices de interfaz de

usuario que los desarrolladores deban cumplir. Incluso si el diseño está determinado por los desarrolladores, es una responsabilidad importante crear un diseño consistente en toda la aplicación.

El Tester: Las pruebas son una parte importante para asegurar que el software funciona de la manera que debería. El papel de ‘Tester’ se realiza a menudo por los desarrolladores para los aspectos técnicos y los usuarios para los aspectos funcionales. Un problema que surge de hacer a los desarrolladores probar su propio código es que, no importa lo bueno que sean, se ven influidos por la forma de su código fue creado. Cuando se prueba, se tendrá en cuenta esas mismas situaciones que ya se tuvieron en cuenta a la hora de escribirlo.

El Administrador de Código: El Código es importante y debe ser tratado como tal, el código necesita ser gestionado. Si varios de los desarrolladores están trabajando en conjunto, el código que escriben debe integrarse en algún momento, independientemente del sistema de control de versiones utilizado.

El Capacitador: Cuando un proyecto se haya completado, los usuarios pueden necesitar ser capacitados, en particular si en el proyecto se desarrollado una aplicación. No es común capacitar a los usuarios de un sitio web, pero a menudo hay un back-end que los administradores tendrán que ser aprender a usar. El Capacitador relaciona las soluciones que se han creado con el usuario final. Una importante responsabilidad del Capacitador es explicar cómo la aplicación resuelve el problema del cliente y, como tal, juega un papel importante en asegurar que las expectativas del cliente sobre el software están en línea con lo que ha sido creado.

Actores secundarios:

Hardware: se requiere de computadores completos, servidores, redes de transmisión de datos, impresoras, dispositivos de almacenamiento. Computadores para pruebas.

Software: encontramos los lenguajes de programación a usar, el sistema operativo (plataforma donde se va a instalar el lenguaje de programación), programas de modelado, programas de análisis y evaluación de software, software de ofimática, software de aplicaciones, programas para diseño de sitios web, sistema de seguimiento y de prueba, servidores web, programas adicionales para diseño de software y diseño de páginas web, y soporte. Bases de datos.

Documentación informativa: libros de ingeniería de software, técnicas, herramientas y soporte documental para el desarrollo de software, métodos de diseño y desarrollo de software, técnicas de evaluación de software.

Otros recursos: cuadernos, hojas, block, donde se plasme los análisis previos de observación, requerimientos, ideas, graficas, diseños, necesidades del cliente, del usuario, costos. Utilería como lápiz, lapiceros.

Herramientas de comunicación, tales como teléfono fijo y móvil, que permita ser localizado y localizar rápidamente a otros miembros del equipo. El correo electrónico y grupos de discusión también son herramientas de uso frecuente.

Herramientas de administración de proyectos, que permitan definir y modificar diagramas Gantt y de flujo de actividades. Ejemplos de esto son MS Project 2000 y MS Project Central, y PRIMAVERA.

Herramientas que permitan contabilizar el uso de recursos, tal como una planilla de cálculos. Ejemplo de esto es MS Excel.

11. 6 Etapa de análisis.

Modelo usado en el desarrollo del software.

Para el desarrollo del presente software se usará “el modelo en cascada”, que es una secuencia lineal de fases y que para seguir a la siguiente fase es necesario tener terminada la fase anterior.

El primer modelo publicado sobre el proceso de desarrollo de software se derivó a partir de procesos más generales de ingeniería de sistemas (Royce, 1970). Debido al paso de una fase en cascada a otra, este modelo se conoce como “modelo en cascada” o ciclo de vida del software. El modelo en cascada es un ejemplo de un proceso dirigido por un plan; en principio, usted debe planear y programar todas las actividades del proceso, antes de comenzar a trabajar con ellas³³.

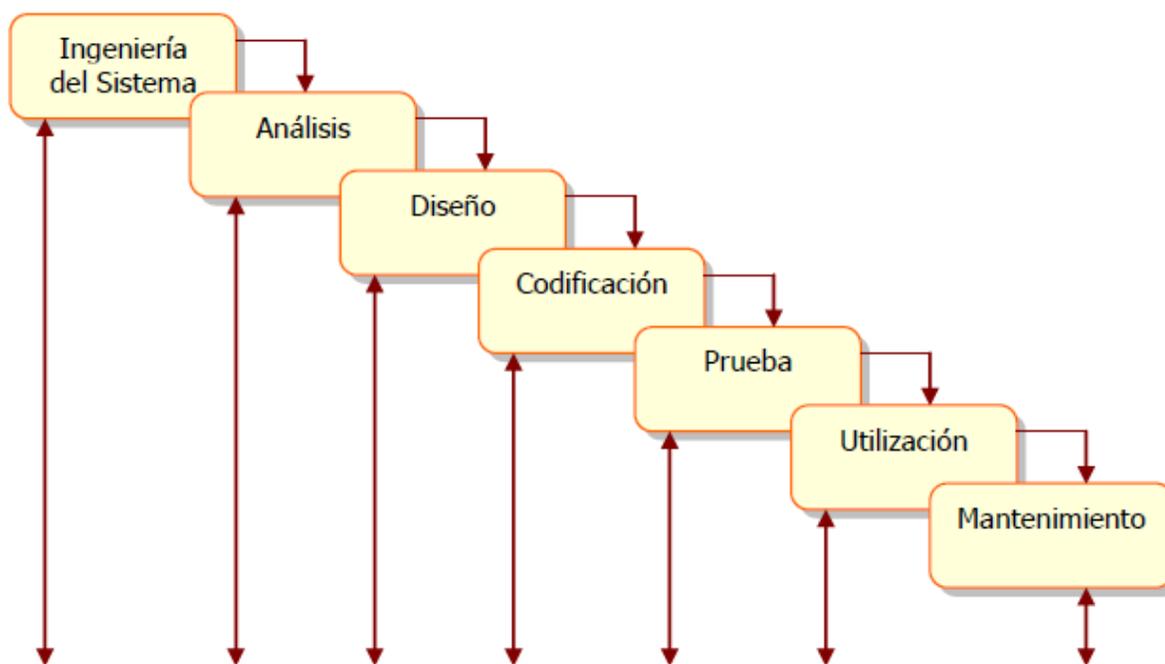


Figura 20. *Imagen modelo en cascada.* Aparicio, A., Pilar, A. M. (2012). Tomada del módulo ingeniería de software. UNAD.

³³ Ian Sommerville. Ingeniería de software 9 edición. modelo en cascada. Pág. 30

Fases del modelo en cascada³⁴:

Ingeniería del Sistema: Análisis de las características y el comportamiento del sistema del cual el software va a formar parte.

Para un sistema nuevo: Se debe analizar cuáles son los requisitos funciones del sistema, y luego asignar un subconjunto de estos requisitos y funciones al software.

Para un sistema ya existente: se debe analizar el funcionamiento de la organización y sus operaciones y se asigna al software aquellas funciones que se van a automatizar. Está formado por diagramas y por descripciones en lenguaje natural.

Análisis: Se debe comprender cuáles son los datos que se van a manejar, cuál va a ser la función que tiene que cumplir el software, cuáles son las interfaces requeridas y cuál es el rendimiento y otros requisitos no funcionales que se esperan lograr.

Los requisitos, tanto del sistema como del software deben documentarse y revisarse con el cliente. Como resultado de la fase de análisis, se obtiene la especificación de requisitos del software. También está formado por diagramas y descripciones en lenguaje natural.

Diseño: El diseño se aplica a cuatro características distintas del software: la estructura de los datos, la arquitectura de las aplicaciones, la estructura interna de los programas y las interfaces.

El diseño es el proceso que traduce los requisitos en una representación del software de forma que pueda conocerse la arquitectura, funcionalidad e incluso la calidad del mismo antes de comenzar la codificación.

³⁴ Alexandra Aparicio, Pilar Alexandra M. Módulo de Ingeniería del Software. UNAD. Diciembre 2012.

En el diseño, los requisitos del software se traducen a una serie de diagramas que representan la estructura del sistema software, de sus datos, de sus programas y de sus interfaces.

Codificación: Consiste en la traducción del diseño a un formato que sea comprensible para la máquina. Si el diseño es lo suficientemente detallado, la codificación es relativamente sencilla, y puede hacerse de forma automática, usando generadores de código.

Se traducen los diagramas de diseño a un lenguaje fuente, que luego se traduce - se compila - para obtener un programa ejecutable.

Prueba: El objetivo es comprobar que no se hayan producido errores en alguna de las fases anteriores, especialmente en la codificación. Se deben probar todas las sentencias, y todos los módulos que forman parte del sistema.

Utilización: El software se entrega al cliente y comienza la vida útil del mismo.

Mantenimiento:

El software sufrirá cambios a lo largo de su vida útil. Estos cambios

pueden ser debidos a tres causas:

Que, durante la utilización, el cliente detecte errores en el software: los errores latentes.

Que se produzcan cambios en alguno de los componentes del sistema.

Que el cliente requiera modificaciones funcionales no contempladas en el proyecto.

Procedimiento.

Ingeniería del sistema: en esta fase se determina de cómo va a ser el software y cómo va a interactuar con el sistema. Ante esto, se pretende desarrollar un software de matemáticas para

el grado noveno, que tenga las temáticas básicas y funcionalidades básicas, cuyo objetivo o fin sea el estudiante y docente, el software se ejecutara bajo el sistema operativo Windows. Aquí se programa y se planea todo lo que se va a hacer en las siguientes fases teniendo en cuenta los requerimientos funcionales, no funcionales, el hardware, el software y de cómo va a ser su funcionamiento y cómo va a interactuar con el usuario.

Identificación de actores:

Actor	Función
<p>Desarrollador de software (Jonny Alexander Mueses Imbacuan), quien hará las veces de:</p> <ul style="list-style-type: none"> ✓ Analista de sistemas. ✓ Diseñador de sistemas. ✓ Programador. ✓ Tester 	<p>Analista:</p> <ul style="list-style-type: none"> • Estudio de los problemas y las necesidades del grado noveno del colegio I.E.R.J.A.S, en el área de matemáticas, para determinar cómo podrían combinarse la tecnología para obtener mejoras en la enseñanza de matemáticas. • Captura, especificación validación de requisitos, interactuando con el usuario (estudiantes, docente). • Identificación de los requerimientos del sistema o software y hardware. • Elaboración del Modelo de Análisis. • Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. • Descripción general completa del problema a resolver. • análisis, síntesis, organización y comunicación.

	<ul style="list-style-type: none">• Definir y desarrollar los requisitos del software y de las interfaces. <p>Diseñador:</p> <ul style="list-style-type: none">✓ Diseño, e implementación del software según las necesidades establecidas.✓ Diseño de la arquitectura general del software.✓ Representaciones de interfaz y algoritmos.✓ Manejo de los requisitos de software.✓ Producción del diseño del sistema.✓ Generar el diseño arquitectónico y diseño detallado del software basándose en los requisitos.✓ Generar prototipos rápidos del sistema para chequear los requisitos.✓ Generar el documento de diseño arquitectónico de software, y mantenerlo actualizado durante el proyecto. <p>Programador:</p> <ul style="list-style-type: none">➤ Construcción de prototipos.➤ Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.➤ Generación del código del programa (software).➤ Depuración del código del programa➤ Diseño de algoritmos, tipos y estructuras de datos.
--	---

	<ul style="list-style-type: none"> ➤ crear la documentación, planificar y realizar la integración de módulos. ➤ Elaboración del modelo de datos. ➤ Preparación de las pruebas funcionales, elaboración de la documentación. ➤ Evaluar el software. <p>Tester:</p> <ul style="list-style-type: none"> ❖ probar y depurar el código generado. ❖ Prueba del software desarrollado ❖ detección/corrección de errores “finos” que no son detectados en las fases anteriores. ❖ Gestión de requisitos, gestión de configuración y cambios.
<p>Usuario final: (estudiantes del grado noveno, docente)</p>	<ul style="list-style-type: none"> ➤ Ejecutar el programa. ➤ Interactuar con el software. ➤ diagnosticar su funcionamiento (del software) cuando se encuentre implementado. ➤ Descripción de requerimientos, especificaciones y necesidades sobre el software. ➤ Descripción de cómo debe funcionar el software o lo que debe hacer. ➤ Especificación de los requerimientos para el desarrollo de software. ➤ Evaluar el software.

Tras observar la situación problema, se identifica que, en el grado noveno, especialmente para el área de matemáticas, no cuentan con un software educativo para dicha área. El colegio

además cuenta con un área de informática, donde hay disponibilidad de computadores. Y a esto le agregamos la capacidad que tienen los estudiantes para manejar o ejecutar algunos programas sencillos que se ven en el área de informática, lo cual facilita la flexibilidad y uso del software a desarrollar.

Los temas de matemáticas que se ven en grado noveno abarcan mucho lo que el álgebra, y ante esto, no todos los estudiantes captan los temas al mismo tiempo, por lo que a muchos les lleva más de tiempo para comprender mejor estos temas.

Teniendo en cuenta lo anterior, y tras haber realizado una observación de tallada en cuanto a infraestructura tecnológica y del medio de enseñanza por parte del docente y aprendizaje por parte del estudiante, se pretende desarrollar un software de matemáticas escrito en lenguaje de programación visual Basic 2010 Ultimate, y que podrá ser usado como complemento en el aprendizaje dentro y fuera del colegio, y tendrá las siguientes características:

Requerimientos funcionales:

- El programa estará diseñado por módulos o interfaces.
- El programa tendrá menú de opciones para lo cual el estudiante requiera hacer. La interfaz principal tendrá una calculadora básica, y tendrá las siguientes opciones:
geometría, algebra, trigonometría, funciones, consultas, ayuda.
- El programa realizara los siguientes cálculos o funciones, de acuerdo al menú u opción elegida:
 1. **Opción ALGEBRA: solución de ecuaciones lineales** de primer grado o una variable, con dos variables, y con tres variables. **Solución de una ecuación cuadrática**, completas

e incompletas. **Solución de determinantes** 2×2 y 3×3 , con números y con variables.

Hallar puntos de corte. Hallar la pendiente de una recta. Hallar la ecuación de una recta.

2. **Opción GEOMETRIA: cuadriláteros:** hallar área y perímetro del cuadrado, rombo, rectángulo, romboide, trapecio. **Cuerpos redondos:** hallar área y volumen de la esfera, cilindro, cono. **Poliedros:** hallar área y volumen del cubo, paralelepípedo, pirámide cuadrada, pirámide regular, prisma regular, tetraedro, prisma triangular. **Polígonos:** hallar área y volumen del triángulo, pentágono, hexágono, heptágono, octágono. **Cónicas:** hallar área del círculo, elipse.
3. **Opción TRIGONOMETRIA: teorema de Pitágoras:** hallar el lado faltante conociendo dos lados. **Funciones trigonométricas:** conocer el valor de todas las funciones trigonométricas conociendo los 3 lados del triángulo rectángulo. **Solución tipo 1, solución tipo 2.**
4. **Opción FUNCIONES: función lineal:** conocer los valores de “y” dando valores a x. **función cuadrática:** conocer los valores de “y” dando valores a x.
- 5 **Opción CONSULTAS:** se verá lo que son **ejemplos:** donde se verán desarrollo de ejercicios de los temas más básicos y que requieren de mayor atención. **Actividades:** de tipo cuestionario, de selección, de diferentes temas, para que el estudiante se ejercite y vea su avance. **Información:** donde se encontrarán textos informativos sobre diferentes temas para que el estudiante pueda consultar y aprender.

6 Opción AYUDA: guía del usuario: es un manual donde le permitirá al estudiante, docente o usuario, conocer el funcionamiento del programa y como debe usarlo para que no haya inconvenientes de funcionalidad y manejo.

- El programa además de mostrar la solución, mostrara un proceso de como se hizo la solución.
- El programa estará implementado con unas actividades
- El programa tendrá la opción de consulta teórica de temas básicos de matemáticas en relación a los temas que maneja el programa.

Requerimientos no funcionales:

- El software tendrá una interfaz amigable y de colores aceptables.
- El software deberá ser de fácil instalación y uso (operabilidad).
- Funcionará bajo el sistema operativo Windows.
- Deberá tener instrucciones claras y entendibles.

Diagrama casos de uso general.

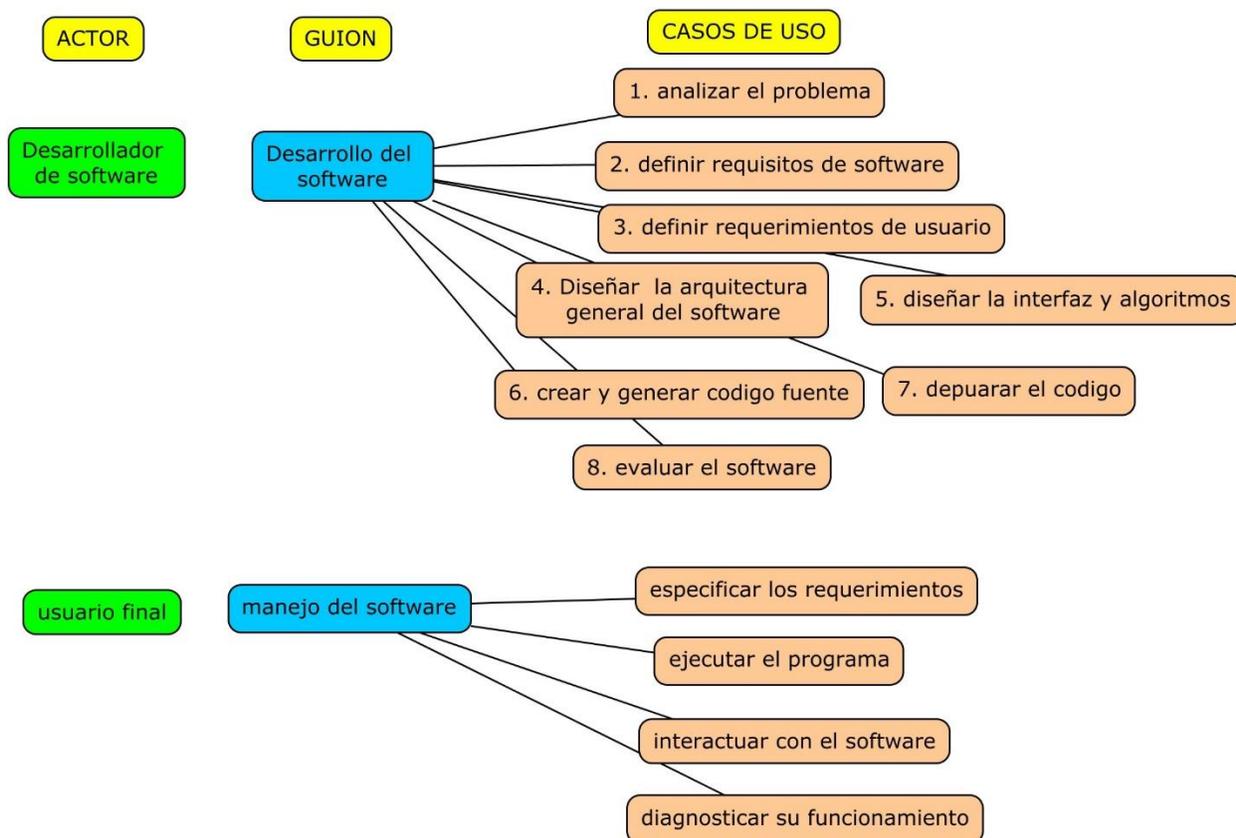


Figura 21: casos de uso del desarrollador del software y del usuario final. Mueses, J. (2017). Fuente: autoría propia

Relaciones entre los casos de usos general.

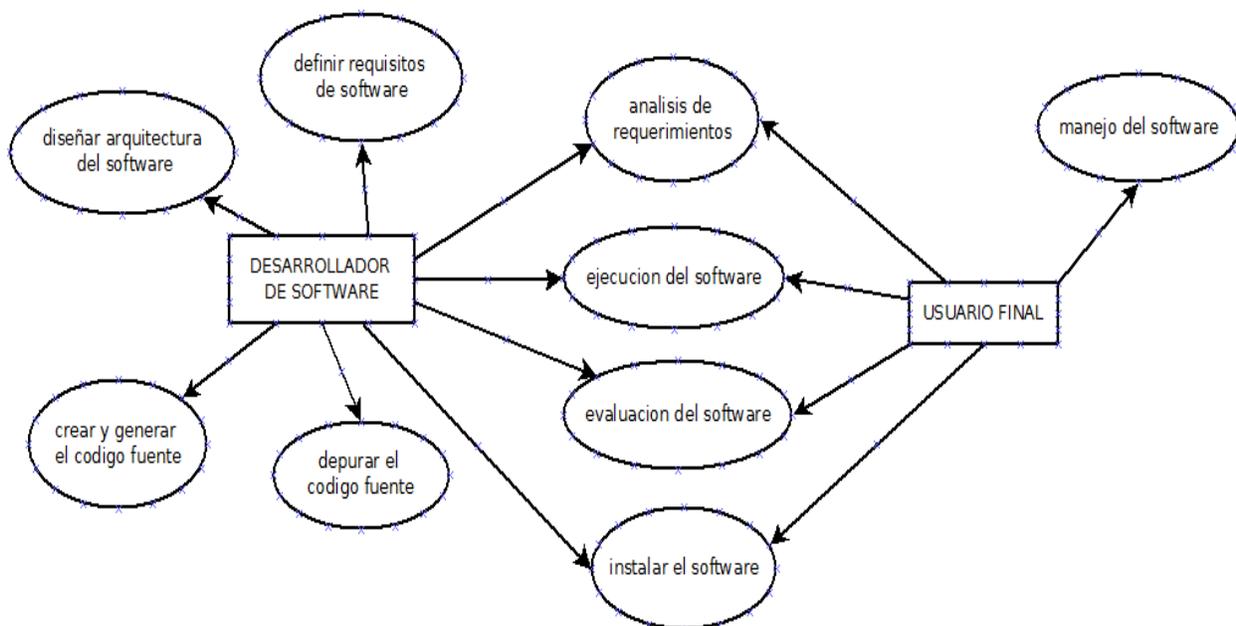


Figura 22: relaciones entre casos de uso del desarrollador y el usuario. Mueses, J. (2017). Fuente: autoría propia

11.7 Etapa de diseño.

Diagramas de caso de uso usuario – sistema.

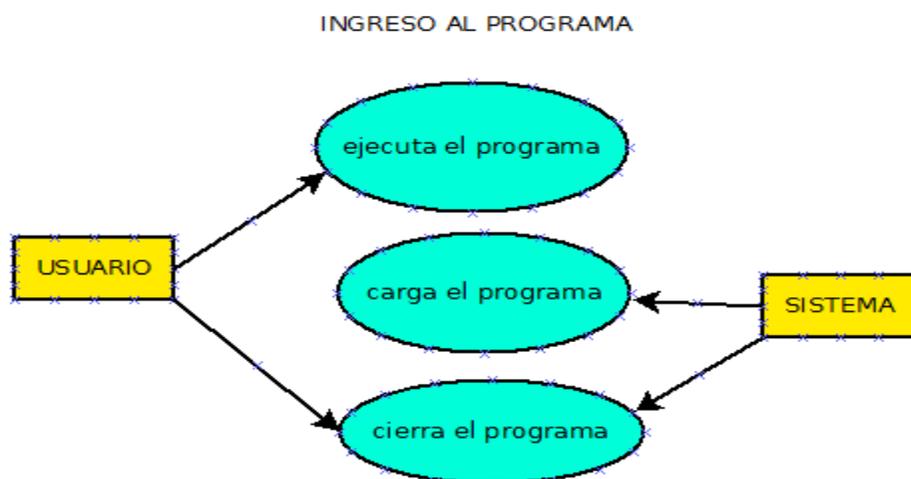


Figura 23: caso de uso de ingreso al programa. Mueses, J. (2017). Fuente: autoría propia

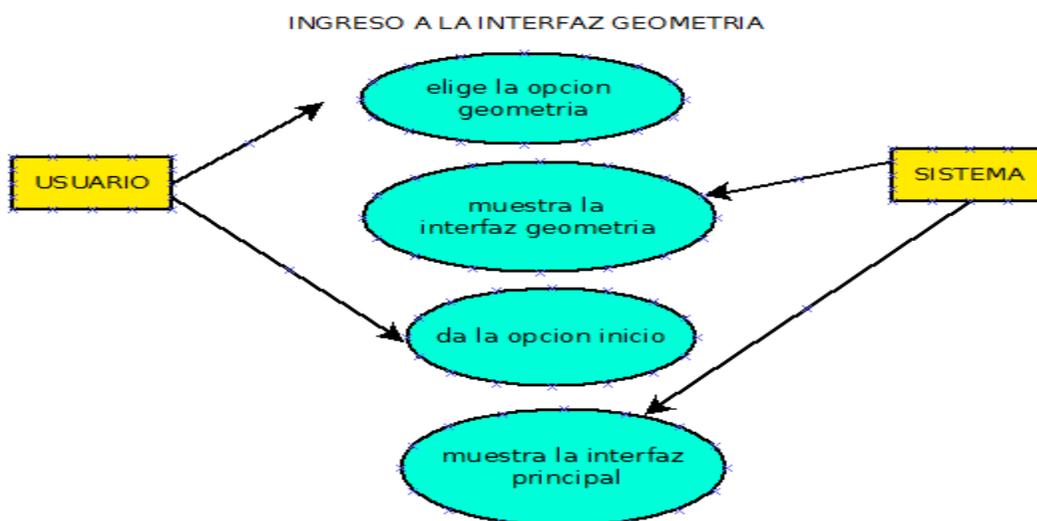


Figura 24: caso de uso ingreso a la interfaz geometría. Mueses, J. (2017). Fuente: autoría propia

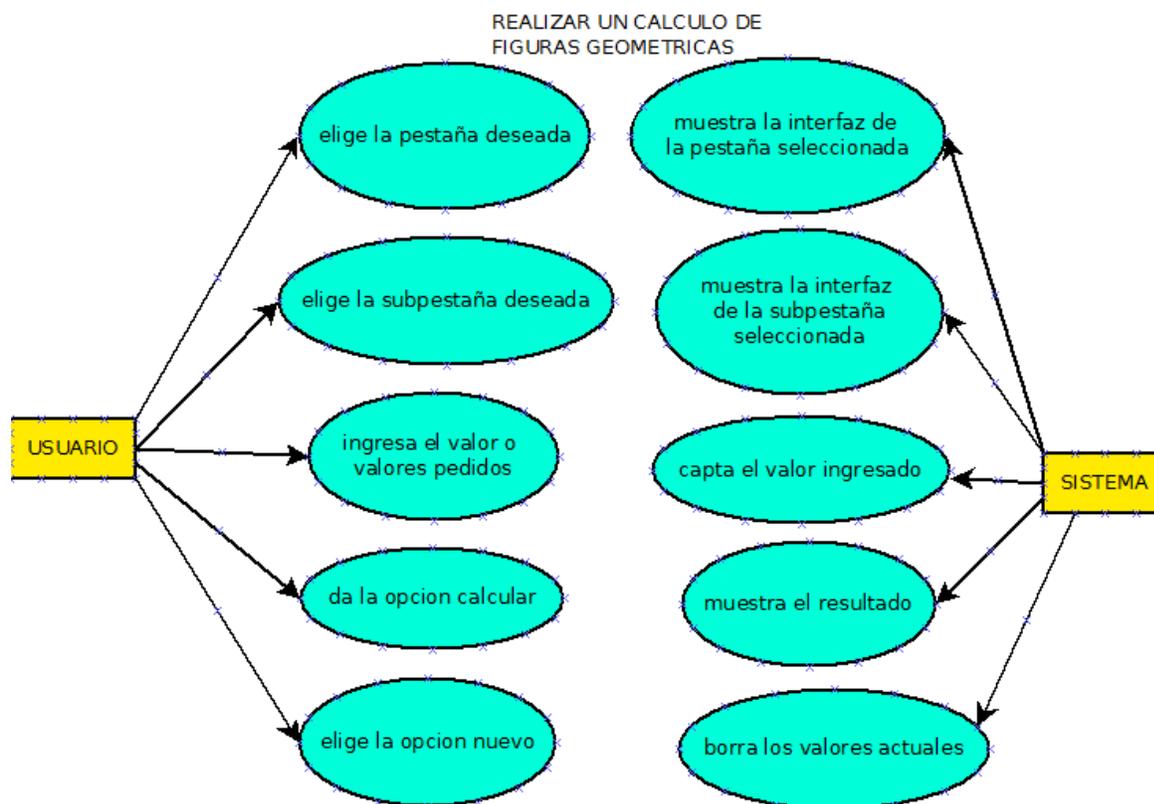


Figura 25: caso de uso: calcular áreas geométricas. Mueses, J. (2017). Fuente: autoría propia

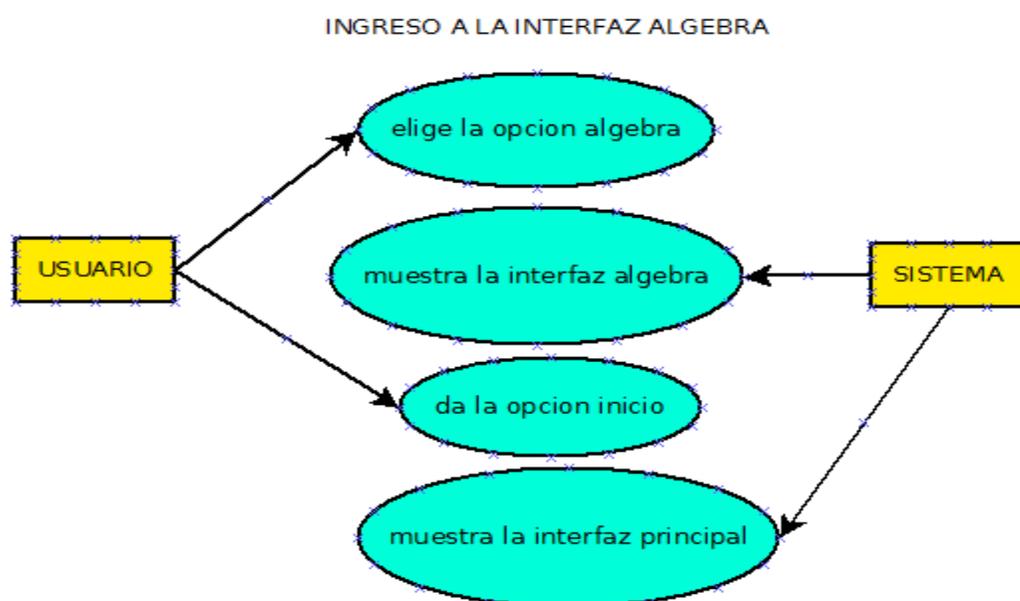


Figura 26: caso de uso: ingreso a la interfaz algebra. Mueses, J. (2017). Fuente: autoría propia



Figura 27: caso de uso: realizar cálculo de temas algebraicos. Mueses, J. (2017). Fuente: autoría propia

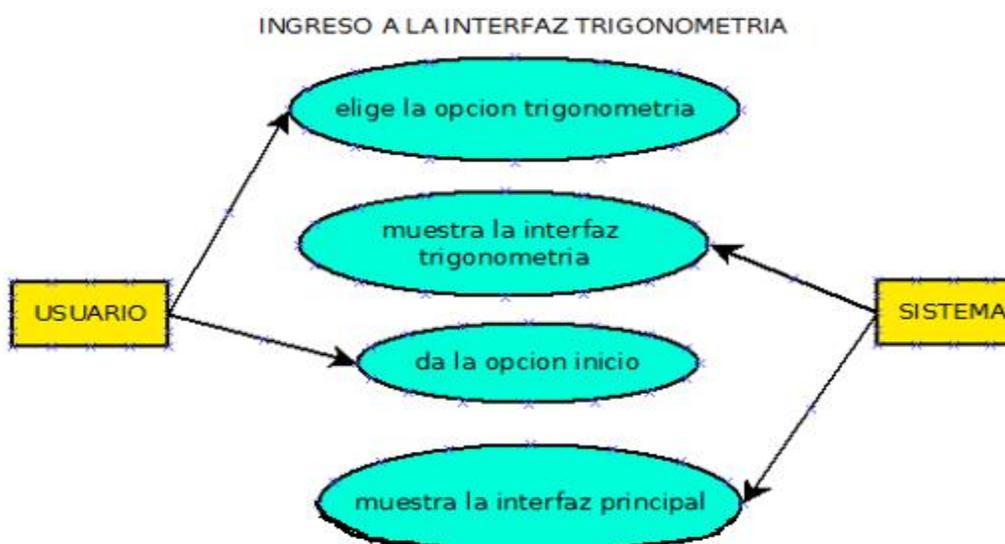


Figura 28: caso de uso: ingreso a la interfaz trigonometría. Mueses, J. (2017). Fuente: autoría propia

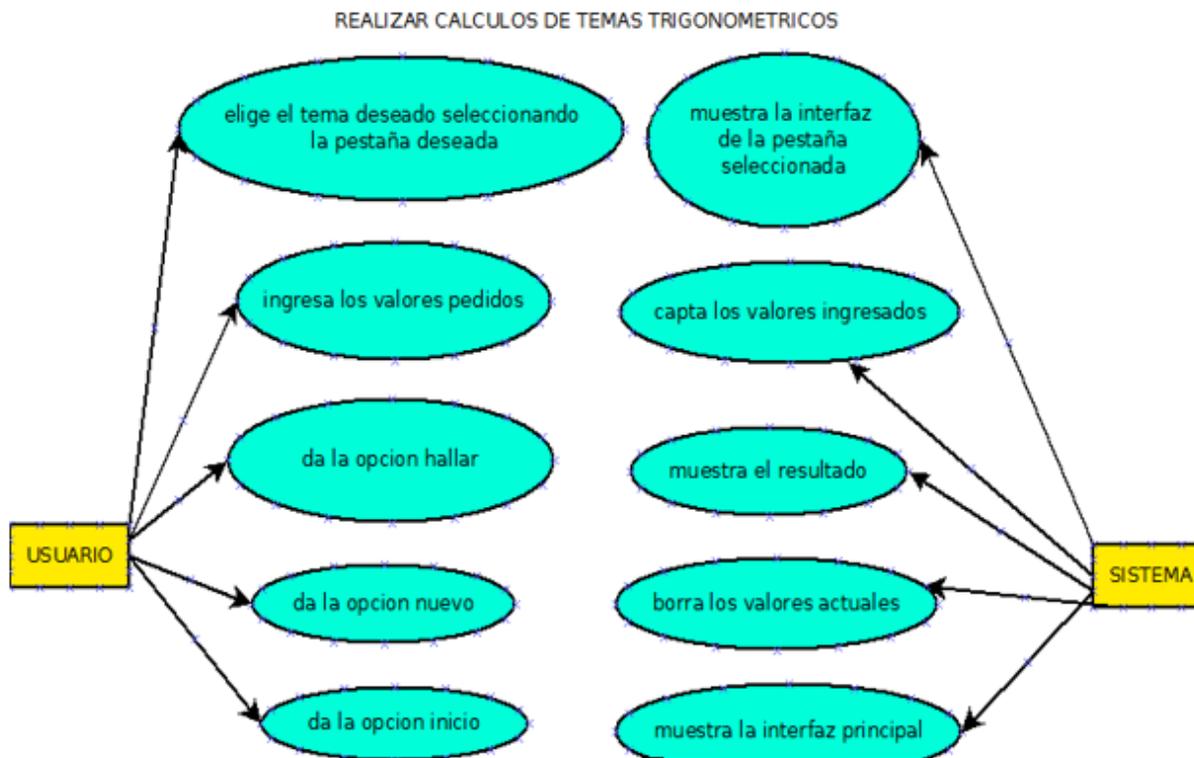


Figura 29: caso de uso: realizar cálculos de temas trigonométricos. Mueses, J. (2017). Fuente: autoría propia

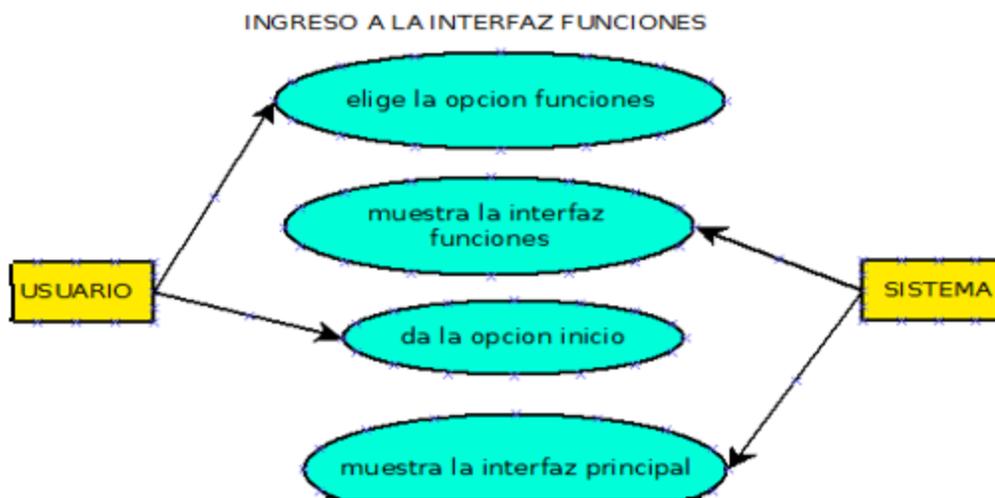


Figura 30: caso de uso: ingreso a la interfaz funciones. Mueses, J. (2017). Fuente: autoría propia



Figura 31: caso de uso: hallar el valor de "y" en una función. Mueses, J. (2017). Fuente: autoría propia

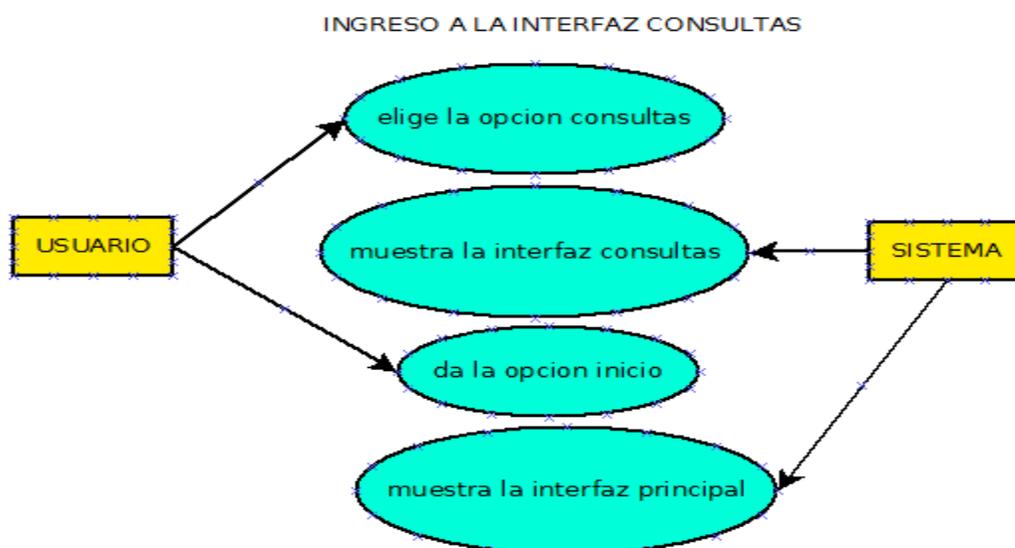


Figura 32: caso de uso: ingreso a la interfaz consultas. Mueses, J. (2017). Fuente: autoría propia

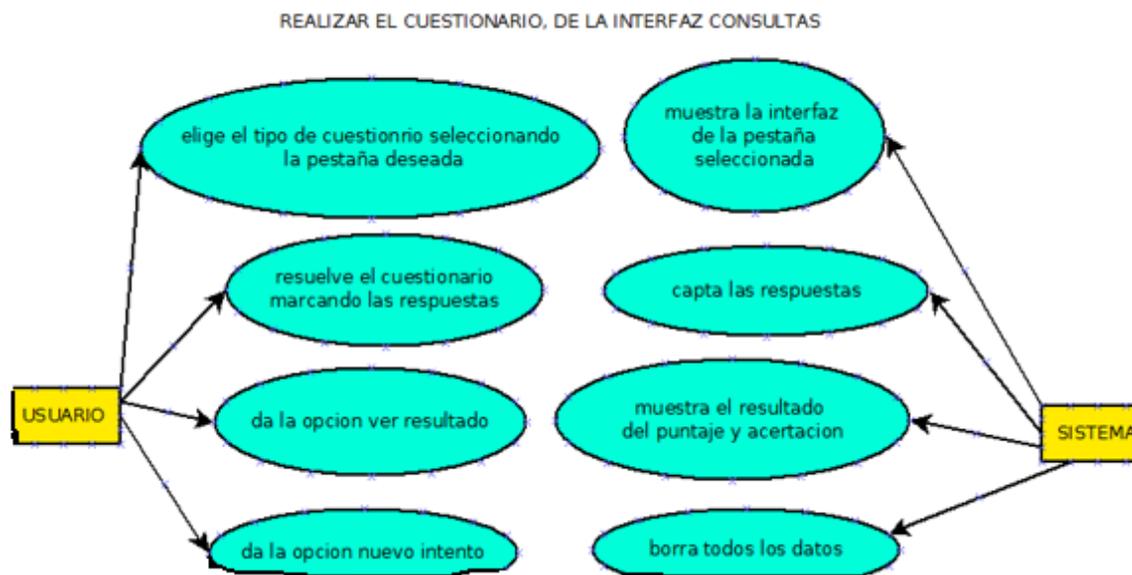


Figura 33: caso de uso: realizar un cuestionario. Mueses, J. (2017). Fuente: autoría propia

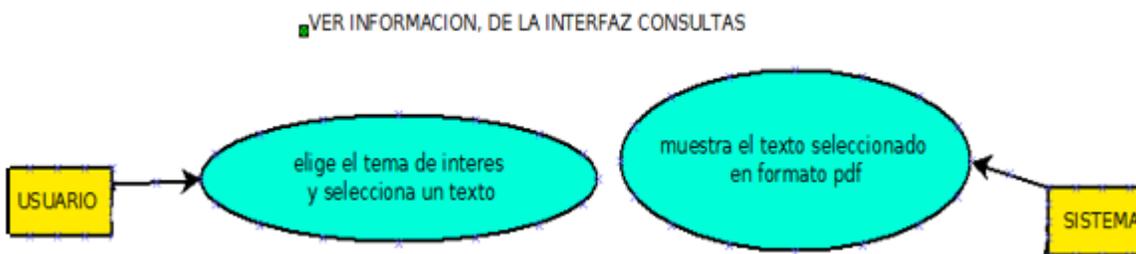


Figura 34: caso de uso: ver información sobre un tema. Mueses, J. (2017). Fuente: autoría propia

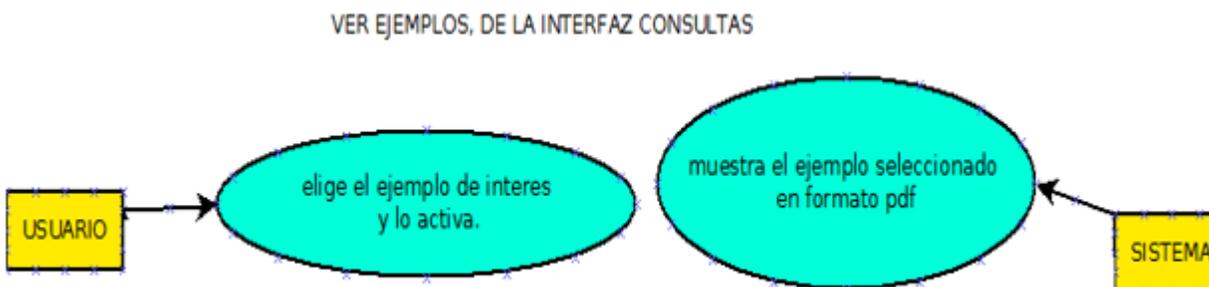


Figura 35: caso de uso: ver ejemplos sobre un tema. Mueses, J. (2017). Fuente: autoría propia

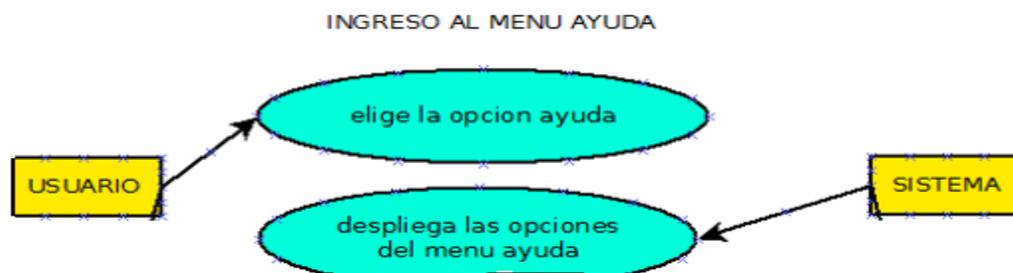


Figura 36: caso de uso: ingresar al menú ayuda. Mueses, J. (2017). Fuente: autoría propia



Figura 37: caso de uso: ver información del producto (software). Mueses, J. (2017). Fuente: autoría propia



Figura 38: caso de uso: ver el manual del producto (software). Mueses, J. (2017). Fuente: autoría propia

Identificación de casos de uso entre el sistema y usuario.

Clase.	Ingresar al programa	
	Usuario	Sistema (programa)
Acciones	1. ejecuta el programa. 4. da clic en el botón cerrar.	2. el programa se carga. 3. el sistema muestra la interfaz principal. 5. el sistema se cierra.

Clase.	Ingresar a la interfaz geometría	
	Usuario	Sistema (programa)
Acciones	1. da clic en el botón geometría. 3. da clic en el botón inicio.	2. el sistema muestra la interfaz geometría. 4. el sistema muestra la interfaz principal.

Clase.	Calcular el área y perímetro de un cuadrado	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz geometría, elige la pestaña cuadriláteros. 3. dentro de la pestaña elige la subpestaña cuadrado. 5. ingresa el valor del lado. 7. da clic en el botón hallar. 9 da clic en el botón nuevo. 11. el usuario da clic en hallar sin ingresar el valor.	2. el sistema muestra el interfaz cuadrilátero. 4. el sistema muestra el interfaz cuadrado. 6. sistema capta el valor. 8. el sistema muestra el resultado. 10. el sistema borra los valores anteriores. 12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.

Clase.	Calcular el área y perímetro de un rombo	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz geometría,	2. el sistema muestra el interfaz

	<p>elige la pestaña cuadriláteros.</p> <p>3. dentro de la pestaña elige la subpestaña rombo.</p> <p>5. ingresa el valor del lado, de la diagonal mayor, de la diagonal menor.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>cuadrilátero.</p> <p>4. el sistema muestra la interfaz rombo.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	--	--

Clase.	Calcular el área y perímetro de un rectángulo	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña cuadriláteros.</p> <p>3. dentro de la pestaña elige la subpestaña rectángulo.</p> <p>5. ingresa el valor de la base y altura.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra el interfaz cuadrilátero.</p> <p>4. el sistema muestra el interfaz rectángulo.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y perímetro de un romboide	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña cuadriláteros.</p> <p>3. dentro de la pestaña elige la</p>	<p>2. el sistema muestra el interfaz cuadrilátero.</p> <p>4. el sistema muestra la interfaz romboide.</p>

<p>subpestaña romboide.</p> <p>5. ingresa el valor de la base, lado y altura.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	---

Clase.	Calcular el área y perímetro de un trapecio.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña cuadriláteros.</p> <p>3. dentro de la pestaña elige la subpestaña trapecio.</p> <p>5. ingresa los valores de: base mayor, base menor, los lados y la altura.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra el interfaz cuadrilátero.</p> <p>4. el sistema muestra la interfaz trapecio.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y volumen de una esfera.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña cuerpos redondos.</p> <p>3. dentro de la pestaña elige la subpestaña esfera.</p>	<p>2. el sistema muestra la interfaz cuerpos redondos.</p> <p>4. el sistema muestra la interfaz esfera.</p> <p>6. sistema capta el valor.</p>

	<p>5. ingresa el valor del radio (r).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	--	--

Clase.	Calcular el área y volumen de un cilindro.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña cuerpos redondos.</p> <p>3. dentro de la pestaña elige la subpestaña cilindro.</p> <p>5. ingresa el valor del radio (r) y altura (h).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra la interfaz cuerpos redondos.</p> <p>4. el sistema muestra la interfaz cilindro.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y volumen de un cono.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña cuerpos redondos.</p> <p>3. dentro de la pestaña elige la</p>	<p>2. el sistema muestra la interfaz cuerpos redondos.</p> <p>4. el sistema muestra la interfaz cono.</p>

<p>subpestaña cono.</p> <p>5. ingresa los valores de radio (r), altura (h), y generatriz o lado (g).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
---	---

Clase.	Calcular el área y volumen de un cubo.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña cubo.</p> <p>5. ingresa el valor del lado (a).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz cubo.</p> <p>6. sistema capta el valor.</p> <p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y volumen de un paralelepípedo.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña paralelepípedo.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz paralelepípedo.</p> <p>6. sistema capta los valores.</p>

	<p>5. ingresa los valores de largo (a), ancho (b), altura (h).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	---	--

Clase.	Calcular el área y volumen de una pirámide cuadrada.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña pirámide cuadrada.</p> <p>5. ingresa los valores de acuerdo al caso elegido.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz pirámide cuadrada.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y volumen de una pirámide regular.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña pirámide regular.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz pirámide regular.</p> <p>6. sistema capta los valores.</p>

	<p>5. ingresa los valores de acuerdo a la indicación dada.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	---	--

Clase.	Calcular el área y volumen de un prisma regular.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña prisma regular.</p> <p>5. ingresa los valores del lado (l) y altura (h).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz prisma regular.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y volumen de un tetraedro.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña tetraedro.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz tetraedro.</p> <p>6. sistema capta el valor.</p> <p>8. el sistema muestra el resultado del área</p>

	<p>5. ingresa el valor del lado (a).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	---	--

Clase.	Calcular el área y volumen de un prisma triangular.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña poliedros.</p> <p>3. dentro de la pestaña elige la subpestaña prisma triangular.</p> <p>5. ingresa los valores de acuerdo a la indicación dada.</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra el interfaz poliedro.</p> <p>4. el sistema muestra la interfaz prisma triangular.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y volumen.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área y perímetro de un polígono.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña polígonos.</p> <p>3. el usuario elige el polígono deseado.</p> <p>5. ingresa el valor del lado (l).</p> <p>7. da clic en el botón hallar.</p>	<p>2. el sistema muestra el interfaz polígono.</p> <p>4. el sistema activa el proceso.</p> <p>6. sistema capta el valor.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p>

	<p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	--	--

Clase.	Calcular el área y perímetro de un círculo.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña (cónicas).</p> <p>3. el usuario elige el círculo.</p> <p>5. ingresa el valor del radio (r).</p> <p>7. da clic en el botón hallar.</p> <p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra la interfaz (cónicas).</p> <p>4. el sistema activa el proceso.</p> <p>6. sistema capta el valor.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p> <p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Calcular el área de una elipse.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz geometría, elige la pestaña (cónicas).</p> <p>3. el usuario elige la elipse.</p> <p>5. ingresa los valores del semieje mayor (b) y del semieje menor (a).</p> <p>7. da clic en el botón hallar.</p>	<p>2. el sistema muestra la interfaz (cónicas).</p> <p>4. el sistema activa el proceso.</p> <p>6. sistema capta los valores.</p> <p>8. el sistema muestra el resultado del área y perímetro.</p>

	<p>9 da clic en el botón nuevo.</p> <p>11. el usuario da clic en hallar sin ingresar el valor.</p>	<p>10. el sistema borra los valores anteriores.</p> <p>12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	--	--

Clase.	Ingresar a la interfaz algebra	
	Usuario	Sistema (programa)
Acciones	<p>1. da clic en el botón algebra.</p> <p>3. da clic en el botón inicio.</p>	<p>2. el sistema muestra la interfaz algebra.</p> <p>4. el sistema muestra la interfaz principal.</p>

Clase.	Hallar el valor de x en una ecuación lineal	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz algebra, elige la pestaña ecuación lineal.</p> <p>3 dentro de la pestaña elige la subpestaña una variable.</p> <p>5. dentro de la subpestaña elige una nueva opción.</p> <p>7. ingresa los valores pedidos por el sistema.</p> <p>9. da clic en el botón hallar.</p> <p>11 da clic en el botón proceso.</p> <p>13 da clic en el botón nuevo.</p> <p>15. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra la interfaz ecuación lineal.</p> <p>4. el sistema muestra la interfaz una variable.</p> <p>6. el sistema muestra la nueva interfaz de la opción elegida.</p> <p>8. sistema capta los valores.</p> <p>10. el sistema muestra el valor de “x” y activa el botón proceso.</p> <p>12. el sistema muestra los pasos del cálculo. Activa el botón nuevo.</p> <p>14. el sistema borra los datos anteriores.</p> <p>16. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Hallar el valor de “x” y “y” en una ecuación lineal.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña ecuación lineal. 3 dentro de la pestaña elige la subpestaña dos variables. 5. dentro de la subpestaña elige una nueva opción. 7. ingresa los valores pedidos por el sistema. 9. da clic en el botón hallar. 11 da clic en el botón proceso. 13 da clic en el botón nuevo. 15. el usuario da clic en hallar sin ingresar el valor.	2. el sistema muestra la interfaz ecuación lineal. 4. el sistema muestra la interfaz dos variables. 6. el sistema muestra la nueva interfaz de la opción elegida. 8. sistema capta los valores. 10. el sistema muestra el valor de “x” y “y”, activa el botón proceso. 12. el sistema muestra los pasos del cálculo. Activa el botón nuevo. 14. el sistema borra los datos anteriores. 16. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.

Clase.	Hallar el valor de “x” y “y” y “z” en una ecuación lineal.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña ecuación lineal. 3 dentro de la pestaña elige la subpestaña tres variables. 5. ingresa los valores pedidos por el sistema. 7. da clic en el botón hallar. 9 da clic en el botón nuevo. 11. el usuario da clic en hallar sin	2. el sistema muestra la interfaz ecuación lineal. 4. el sistema muestra la interfaz tres variables. 6. sistema capta los valores. 8. el sistema muestra el valor de “x” y “y”, y “z” activa el botón nuevo. 10. el sistema borra los datos anteriores. 12. el sistema muestra un mensaje de

	ingresar el valor.	error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.
--	--------------------	--

Clase.	Hallar el valor de “x” en una ecuación cuadrática.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña ecuación cuadrática. 3 dentro de la pestaña elige la subpestaña deseada. 5. ingresa los valores pedidos por el sistema. 7. da clic en el botón hallar. 9 da clic en el botón proceso. 11 da clic en el botón nuevo. 13. el usuario da clic en hallar sin ingresar el valor.	2. el sistema muestra la interfaz ecuación cuadrática. 4. el sistema muestra la interfaz de la subpestaña elegida. 6. sistema capta los valores. 8. el sistema muestra los valores de “x”, activa el botón proceso. 10. el sistema muestra los pasos del cálculo. Activa el botón nuevo. 12. el sistema borra los datos anteriores. 14. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.

Clase.	Hallar el valor de determinantes.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña “determinantes”. 3 dentro de la pestaña elige la subpestaña deseada. 5. ingresa los valores pedidos por el sistema. 7. da clic en el botón hallar. 9 da clic en el botón nuevo.	2. el sistema muestra la interfaz (determinantes). 4. el sistema muestra la interfaz de la subpestaña elegida. 6. sistema capta los valores. 8. el sistema muestra el resultado y procesos, activa el botón nuevo. 10. el sistema borra los datos anteriores.

	11. el usuario da clic en hallar sin ingresar el valor.	12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.
--	---	---

Clase.	Hallar los puntos de corte.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña puntos de corte. 3 dentro de la pestaña elige la subpestaña deseada. 5. ingresa los valores pedidos por el sistema. 7. da clic en el botón hallar. 9 da clic en el botón proceso. 11 da clic en el botón nuevo. 13. el usuario da clic en hallar sin ingresar el valor.	2. el sistema muestra la interfaz puntos de corte. 4. el sistema muestra la interfaz de la subpestaña elegida. 6. sistema capta los valores. 8. el sistema muestra los valores de “x”, y “y” activa el botón proceso. 10. el sistema muestra los pasos del cálculo. Activa el botón nuevo. 12. el sistema borra los datos anteriores. 14. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.

Clase.	Hallar la pendiente de una recta.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña pendiente de la recta 3. ingresa los valores pedidos por el sistema (x1, y1, x2, y2). 5. da clic en el botón hallar. 7 da clic en el botón proceso.	2. el sistema muestra la interfaz pendiente de la recta. 4. sistema capta los valores. 6. el sistema muestra el valor de m, activa el botón proceso. 8. el sistema muestra los pasos del cálculo. Activa el botón nuevo.

	9 da clic en el botón nuevo. 11. el usuario da clic en hallar sin ingresar el valor.	10. el sistema borra los datos anteriores. 12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.
--	---	---

Clase.	Hallar la ecuación de una recta.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz algebra, elige la pestaña ecuación de una recta. 3. ingresa los valores pedidos por el sistema (m, x, y). 5. da clic en el botón hallar. 7 da clic en el botón proceso. 9 da clic en el botón nuevo. 11. el usuario da clic en hallar sin ingresar el valor.	2. el sistema muestra la interfaz ecuación de una recta. 4. sistema capta los valores. 6. el sistema muestra el resultado, activa el botón proceso. 8. el sistema muestra los pasos del cálculo. Activa el botón nuevo. 10. el sistema borra los datos anteriores. 12. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.

Clase.	Ingresar a la interfaz trigonometría	
	Usuario	Sistema (programa)
Acciones	1. da clic en el botón trigonometría. 3. da clic en el botón inicio.	2. el sistema muestra la interfaz trigonometría. 4. el sistema muestra la interfaz principal.

Clase.	Manejo del teorema de Pitágoras.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz	2. el sistema muestra la interfaz teorema

	<p>trigonometría, elige la pestaña teorema de Pitágoras.</p> <p>3. selecciona las dos variables conocidas.</p> <p>5. da clic en mostrar.</p> <p>7. ingresa los valores de las variables conocidas.</p> <p>9. da clic en el botón hallar.</p> <p>11 da clic en el botón proceso.</p> <p>13 da clic en el botón nuevo.</p> <p>15. el usuario da clic en hallar sin ingresar el valor.</p>	<p>de Pitágoras.</p> <p>4. el sistema almacena las variables.</p> <p>6. el sistema muestra los campos de las variables, activa el botón hallar.</p> <p>8. sistema capta los valores.</p> <p>10. el sistema muestra el resultado, activa el botón proceso.</p> <p>12. el sistema muestra los pasos del cálculo. Activa el botón nuevo.</p> <p>14. el sistema borra los datos anteriores.</p> <p>16. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	---	--

Clase.	Hallar el valor de las funciones trigonométricas.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz trigonometría, elige la pestaña funciones trigonométricas.</p> <p>3. ingresa los valores de las variables conocidas (x, y, h).</p> <p>5. da clic en el botón hallar.</p> <p>7 da clic en el botón nuevo.</p> <p>9. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra la interfaz funciones trigonométricas.</p> <p>4. sistema capta los valores.</p> <p>6. el sistema muestra el resultado, activa el botón nuevo.</p> <p>8. el sistema borra los datos anteriores.</p> <p>10. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Funciones trigonométricas conociendo 2 lados.	
	Usuario	Sistema (programa)

Acciones	<p>1. estando en la interfaz trigonometría, elige la pestaña solución tipo 1.</p> <p>3. selecciona las dos variables conocidas.</p> <p>5. da clic en mostrar.</p> <p>7. ingresa los valores de las variables conocidas.</p> <p>9. da clic en el botón hallar.</p> <p>11 da clic en el botón nuevo.</p> <p>13. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra la interfaz solución tipo 1.</p> <p>4. el sistema almacena las variables.</p> <p>6. el sistema muestra los campos de las variables, activa el botón hallar.</p> <p>8. sistema capta los valores.</p> <p>10. el sistema muestra el resultado y proceso, activa el botón nuevo.</p> <p>12. el sistema borra los datos anteriores.</p> <p>14. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
-----------------	---	---

Clase.	Funciones trigonométricas conociendo un lado y un ángulo.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz trigonometría, elige la pestaña solución tipo 2.</p> <p>3. selecciona el lado y el ángulo conocido.</p> <p>5. da clic en continuar.</p> <p>7. ingresa los valores de las variables conocidas.</p> <p>9. da clic en el botón hallar.</p> <p>11 da clic en el botón nuevo.</p> <p>13. el usuario da clic en hallar sin ingresar el valor.</p>	<p>2. el sistema muestra la interfaz solución tipo 2.</p> <p>4. el sistema almacena las variables del lado y el ángulo.</p> <p>6. el sistema muestra los campos de las variables, activa el botón hallar.</p> <p>8. sistema capta los valores.</p> <p>10. el sistema muestra el resultado y proceso, activa el botón nuevo.</p> <p>12. el sistema borra los datos anteriores.</p> <p>14. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>

Clase.	Ingresar a la interfaz funciones	
	Usuario	Sistema (programa)
Acciones	1. da clic en el botón funciones. 3. da clic en el botón inicio.	2. el sistema muestra la interfaz funciones. 4. el sistema muestra la interfaz principal

Clase.	Hallar el valor de “y” en una función lineal	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz funciones, elige la pestaña lineal. 3. dentro de la pestaña elige la subpestaña deseada. 5. ingresa los valores pedidos por el sistema. 7. da clic en el botón continuar. 9. ingresa el valor de “x”. 11. da clic en hallar. 13. da clic en el botón nuevo. 15. el usuario da clic en hallar sin ingresar el valor.	2. el sistema muestra la interfaz lineal. 4. el sistema muestra la interfaz de la pestaña seleccionada. 6. el sistema capta los valores iniciales. 8. sistema activa el siguiente proceso, activa el botón hallar. 10. el sistema capta el valor. 12. el sistema muestra el valor de “y”, activa el botón almacenar. 14. el sistema borra los datos anteriores. 16. sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.

Clase.	Hallar el valor de “y” en una función cuadrática.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz funciones, elige la pestaña cuadrática. 3. dentro de la pestaña elige la subpestaña deseada. 5. ingresa los valores pedidos por el	2. el sistema muestra la interfaz cuadrática. 4. el sistema muestra la interfaz de la pestaña seleccionada. 6. el sistema capta los valores iniciales.

	<p>sistema.</p> <p>7. da clic en el botón continuar.</p> <p>9. ingresa el valor de “x”.</p> <p>11. da clic en hallar.</p> <p>13. da clic en el botón nuevo.</p> <p>15. el usuario da clic en hallar sin ingresar el valor.</p>	<p>8. sistema activa el siguiente proceso, activa el botón hallar.</p> <p>10. el sistema capta el valor.</p> <p>12. el sistema muestra el valor de “y”, activa el botón almacenar.</p> <p>14. el sistema borra los datos anteriores.</p> <p>16. el sistema muestra un mensaje de error indicando que debe ingresar un valor antes de iniciar el proceso de cálculo.</p>
--	--	---

Clase.	Ingresar a la interfaz consultas	
	Usuario	Sistema (programa)
Acciones	<p>1. da clic en el botón consultas.</p> <p>2. da clic en el botón inicio.</p>	<p>2. el sistema muestra la interfaz consultas.</p> <p>4. el sistema muestra la interfaz principal</p>

Clase.	Resolver un cuestionario.	
	Usuario	Sistema (programa)
Acciones	<p>1. estando en la interfaz consultas, elige la pestaña actividades.</p> <p>3. dentro de la pestaña elige la subpestaña deseada.</p> <p>5. resuelve el cuestionario, seleccionando la respuesta.</p> <p>7. da clic en el botón ver resultado.</p> <p>9. da clic en el botón nuevo intento.</p>	<p>2. el sistema muestra la interfaz actividades.</p> <p>4. el sistema muestra la interfaz de la pestaña seleccionada.</p> <p>6. el sistema capta las respuestas.</p> <p>8. el sistema muestra el puntaje obtenido y el número de respuestas acertadas.</p> <p>10. el sistema borra los puntos seleccionados.</p>

Clase.	Ver información sobre un tema.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz consultas, elige la pestaña información. 3. dentro de la pestaña, elije el tema y da clic en un texto deseado.	2. el sistema muestra la interfaz información. 4. el sistema muestra el texto elegido, en formato pdf.

Clase.	Ver ejemplos resueltos sobre un tema.	
	Usuario	Sistema (programa)
Acciones	1. estando en la interfaz consultas, elige la pestaña ejemplos. 3. dentro de la pestaña, busca el ejemplo deseado.	2. el sistema muestra la interfaz ejemplos. 4. el sistema muestra el ejemplo elegido, en formato pdf..

Clase.	Ingresar al menú ayuda.	
	Usuario	Sistema (programa)
Acciones	1. da clic en el menú ayuda.	2. el sistema despliega las opciones del menú ayuda.

Clase.	Ver información del producto (software).	
	Usuario	Sistema (programa)
Acciones	1. estando en el menú ayuda, da clic en acerca de. 3. da clic en el botón aceptar.	2. el sistema muestra la interfaz acerca de con la información del producto. 4. el sistema muestra la interfaz principal.

Clase.	Ver manual del producto (software).	
---------------	--	--

	Usuario	Sistema (programa)
Acciones	1. estando en el menú ayuda, da clic en manual de uso. 3. da clic en el botón cerrar.	2. el sistema muestra la información guía en formato de texto. 4. el sistema cierra la interfaz.

Diagramas de secuencias.



Figura 39: *diagrama de secuencia: ingresar al programa y a la interfaz geometría.* Mueses, J. (2017). Fuente: autoría propia

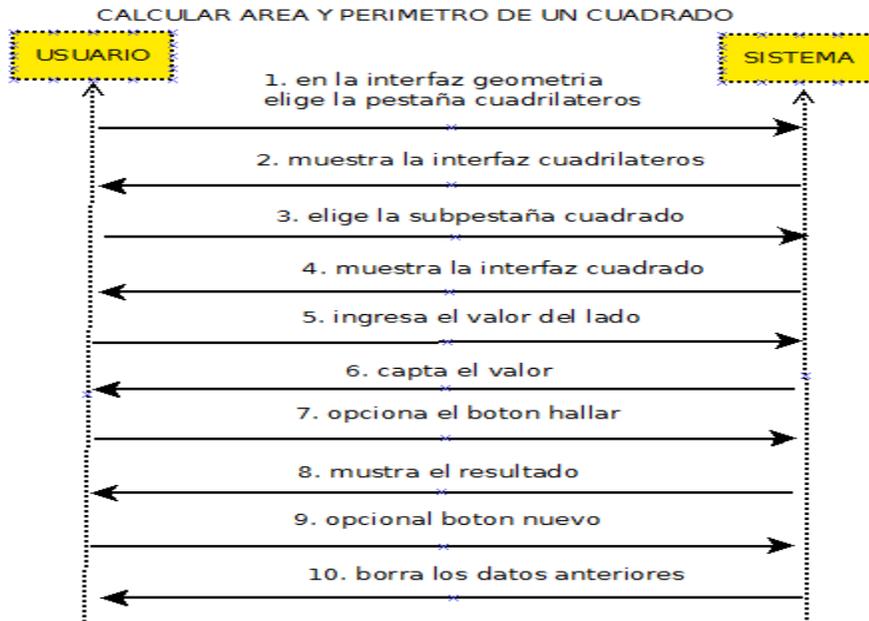


Figura 40: diagrama de secuencia: calcular área y perímetro de un cuadrado. Mueses, J. (2017). Fuente: autoría propia



Figura 41: diagrama de secuencia: calcular área y perímetro de un rombo. Mueses, J. (2017). Fuente: autoría propia

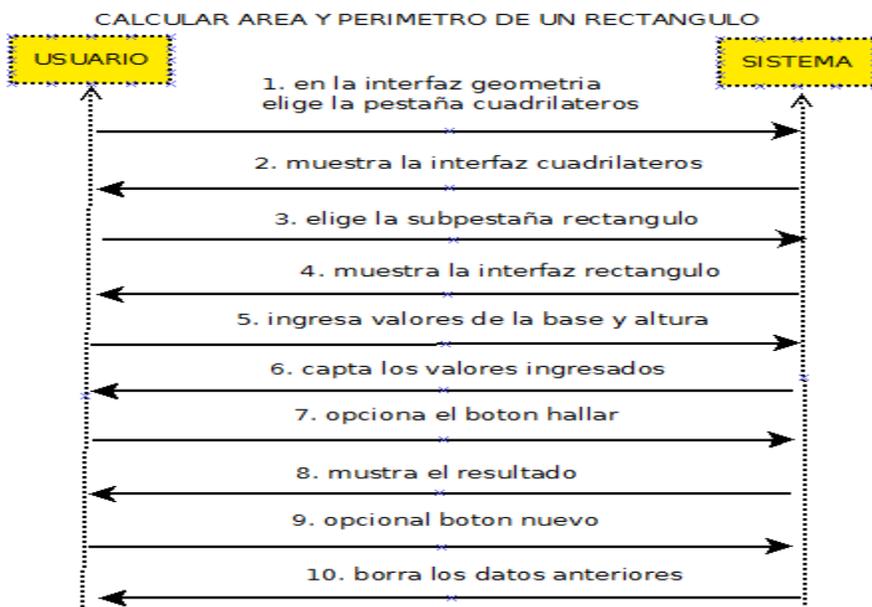


Figura 42: diagrama de secuencia: calcular área y perímetro de un rectángulo. Mueses, J. (2017). Fuente: autoría propia

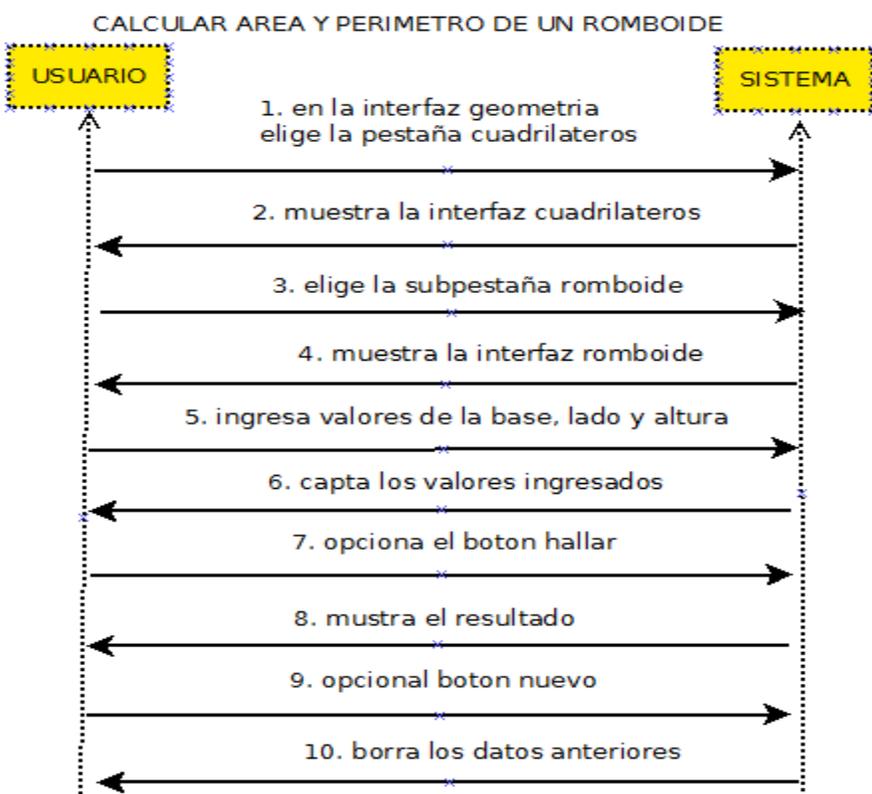


Figura 43: diagrama de secuencia: calcular área y perímetro de un romboide. Mueses, J. (2017). Fuente: autoría propia



Figura 44: diagrama de secuencia: calcular área y perímetro de un trapecio. Mueses, J. (2017). Fuente: autoría propia

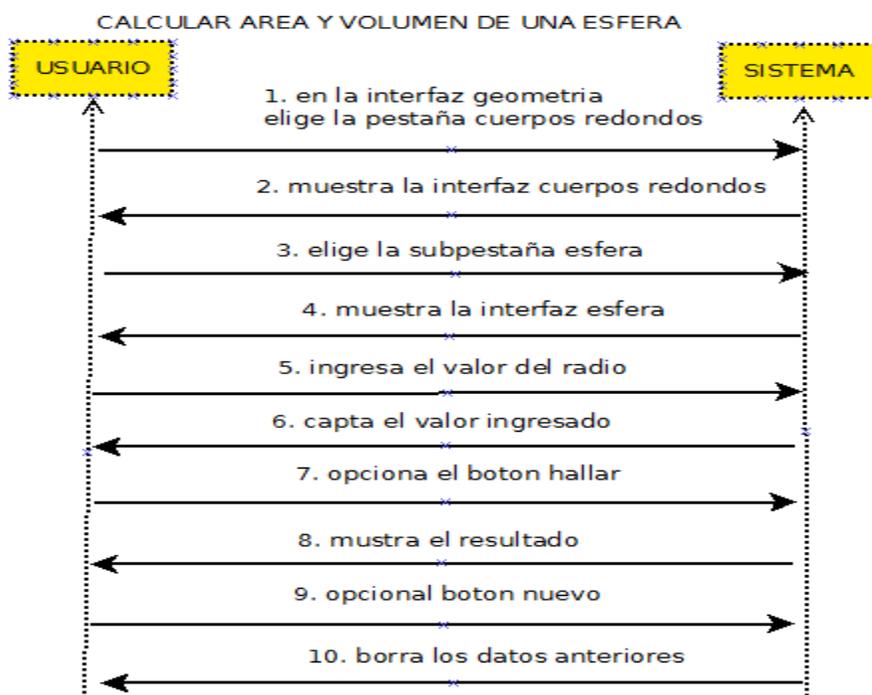


Figura 45: diagrama de secuencia: calcular área y volumen de una esfera. Mueses, J. (2017). Fuente: autoría propia

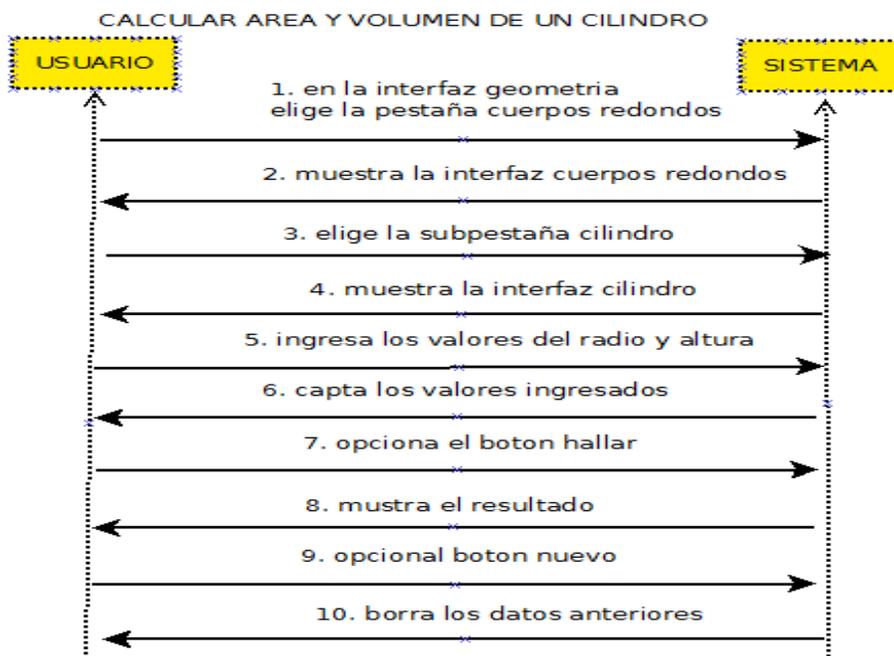


Figura 46: diagrama de secuencia: calcular área y volumen de un cilindro. Mueses, J. (2017). Fuente: autoría propia

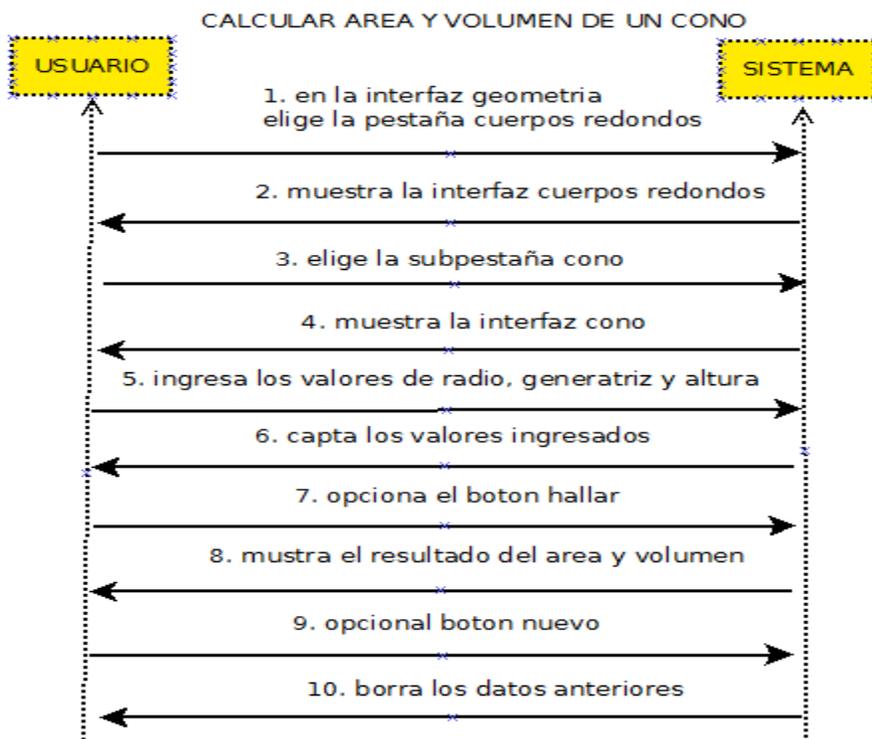


Figura 47: diagrama de secuencia: calcular área y volumen de un cono. Mueses, J. (2017). Fuente: autoría propia

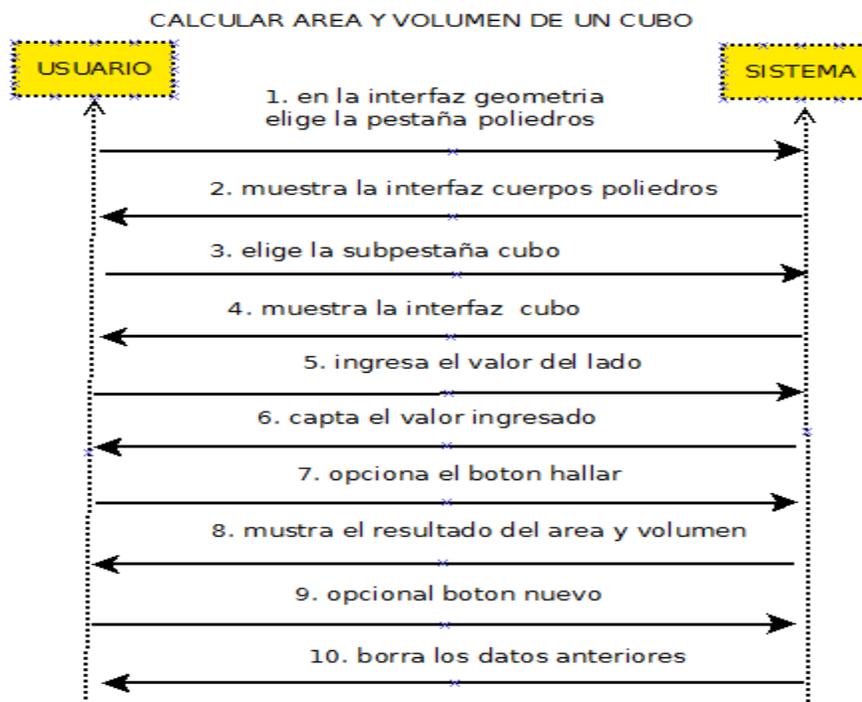


Figura 48: diagrama de secuencia: calcular área y volumen de un cubo. Mueses, J. (2017). Fuente: autoría propia

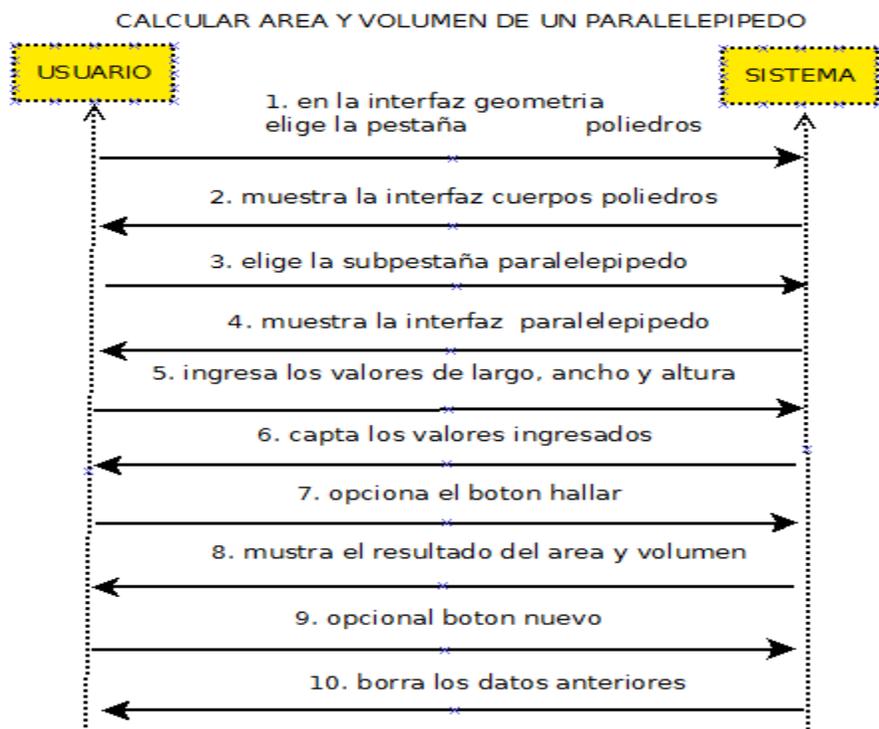


Figura 49: diagrama de secuencia: calcular área y volumen de un paralelepípedo. Mueses, J. (2017). Fuente: autoría propia

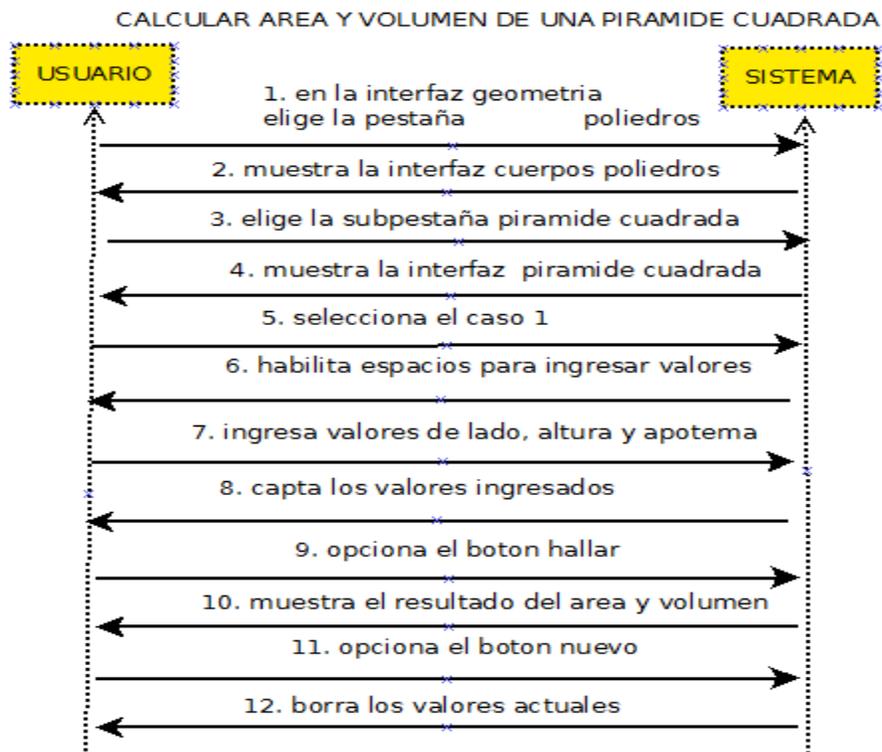


Figura 50: diagrama de secuencia: calcular área y volumen de una pirámide cuadrada. Mueses, J. (2017). Fuente: autoría propia

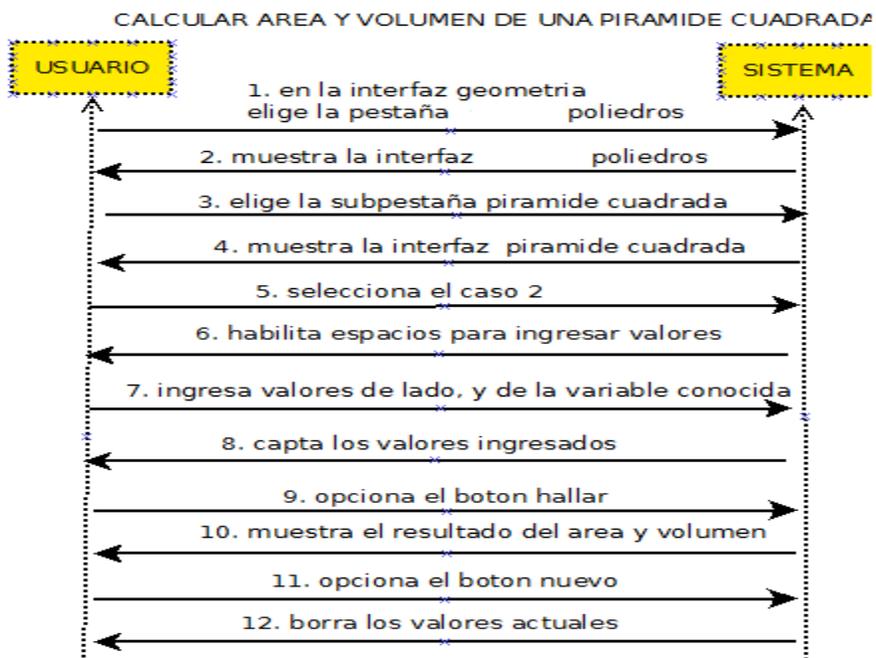


Figura 51: diagrama de secuencia: calcular área y volumen de una pirámide cuadrada, caso 2. Mueses, J. (2017). Fuente: autoría propia

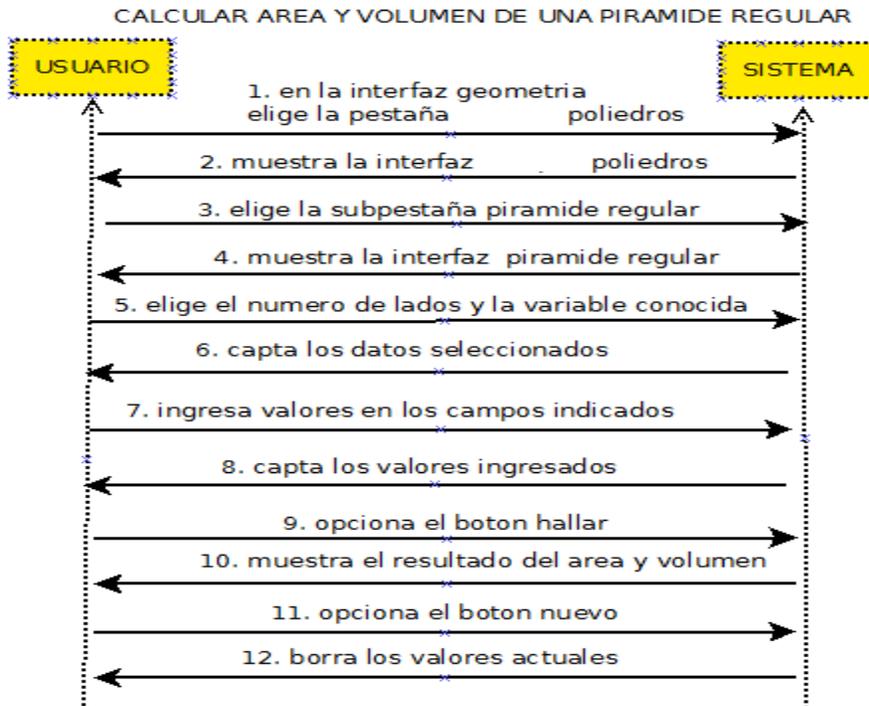


Figura 52: diagrama de secuencia: calcular área y volumen de una pirámide regular. Mueses, J. (2017). Fuente: autoría propia

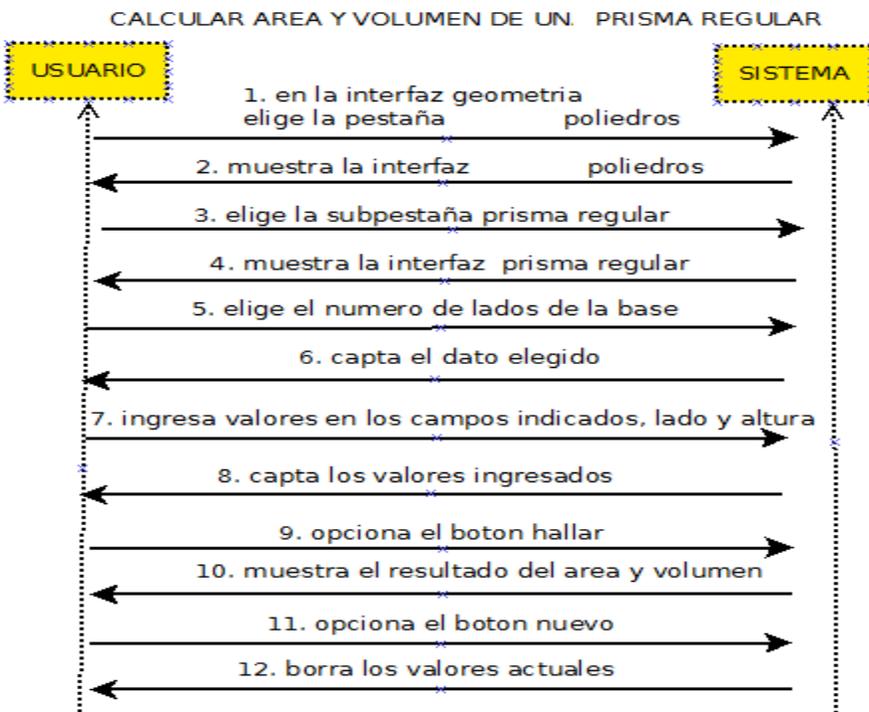


Figura 53: diagrama de secuencia: calcular área y volumen de un prisma regular. Mueses, J. (2017). Fuente: autoría propia

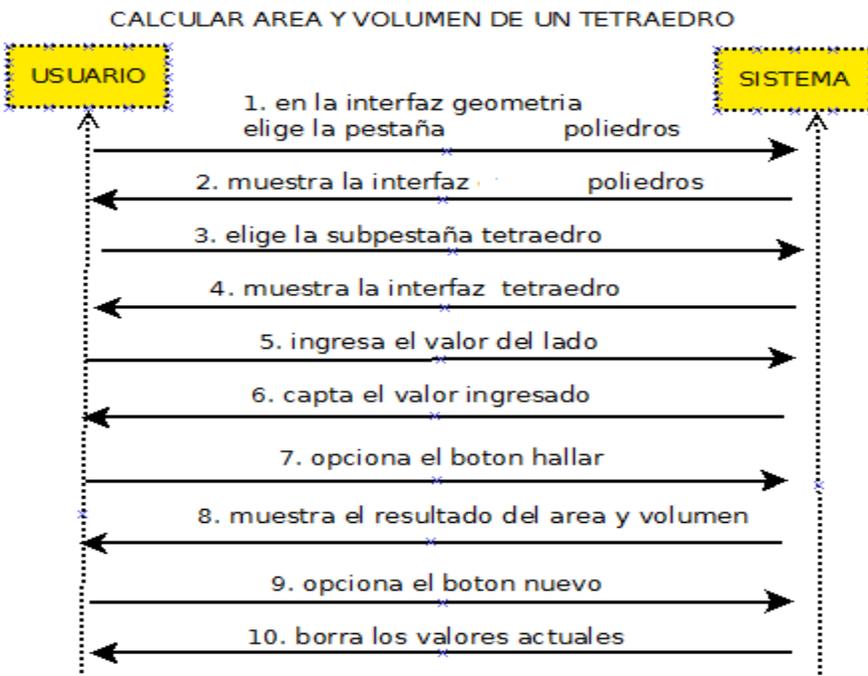


Figura 54: diagrama de secuencia: calcular área y volumen de un tetraedro. Mueses, J. (2017). Fuente: autoría propia

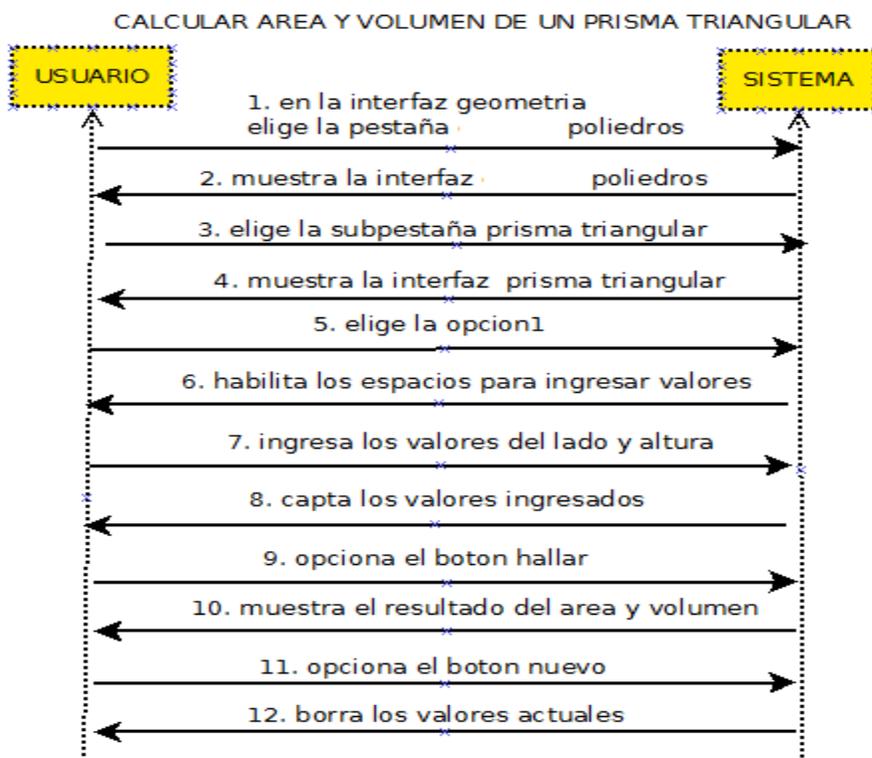


Figura 55: diagrama de secuencia: calcular área y volumen de un prisma triangular. Mueses, J. (2017). Fuente: autoría propia

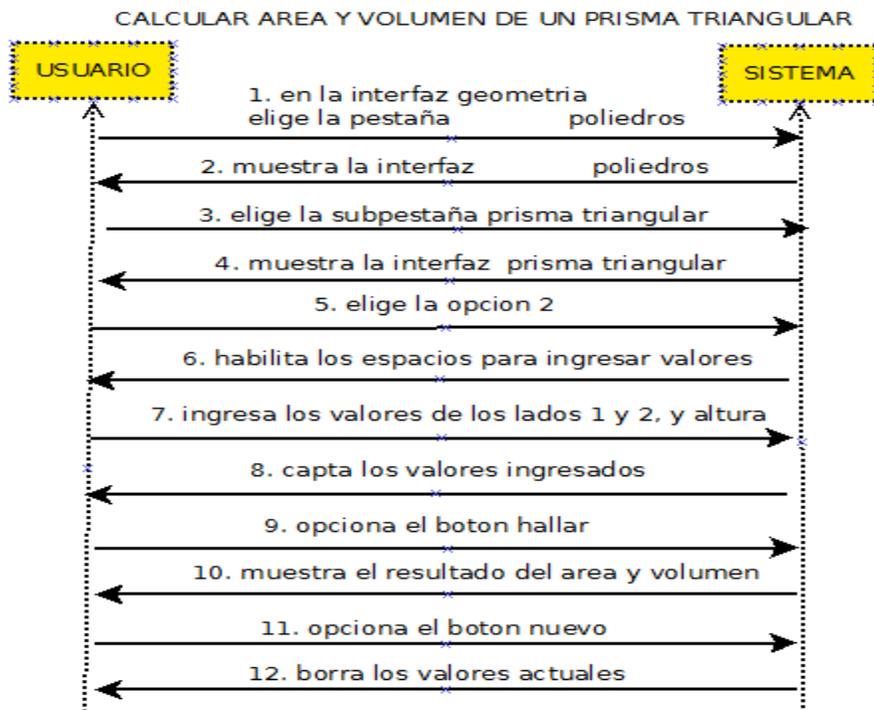


Figura 56: diagrama de secuencia: calcular área y volumen de un prisma triangular, opción 2. Mueses, J. (2017). Fuente: autoría propia

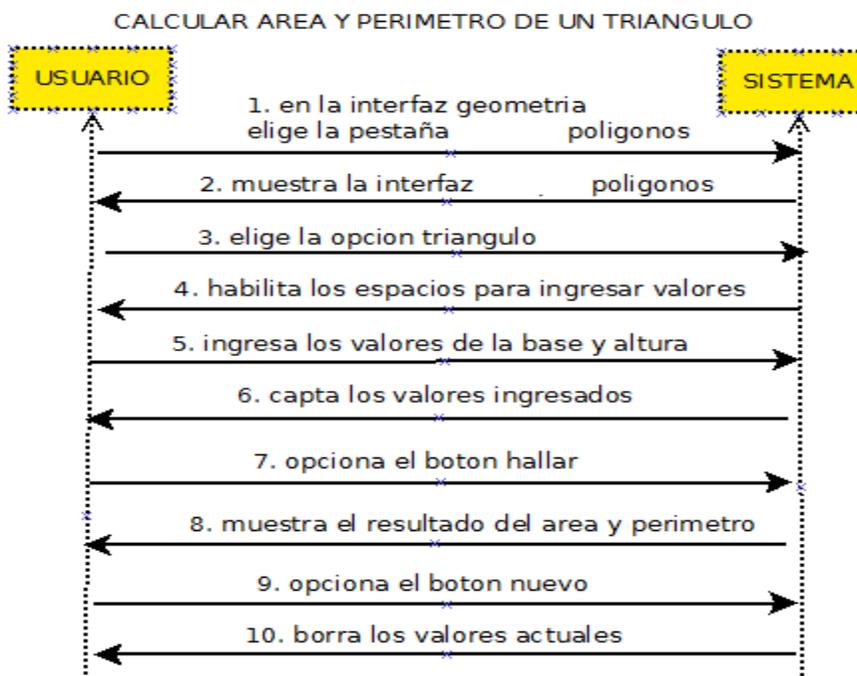


Figura 57: diagrama de secuencia: calcular área y perímetro de un triángulo. Mueses, J. (2017). Fuente: autoría propia

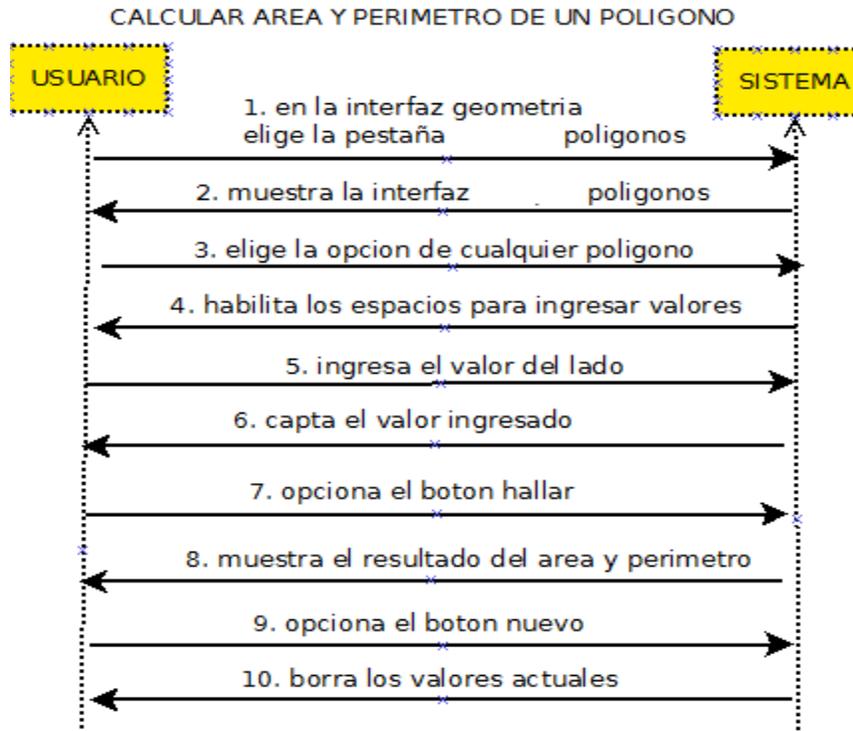


Figura 58: diagrama de secuencia: calcular área y perímetro de un polígono. Mueses, J. (2017). Fuente: autoría propia

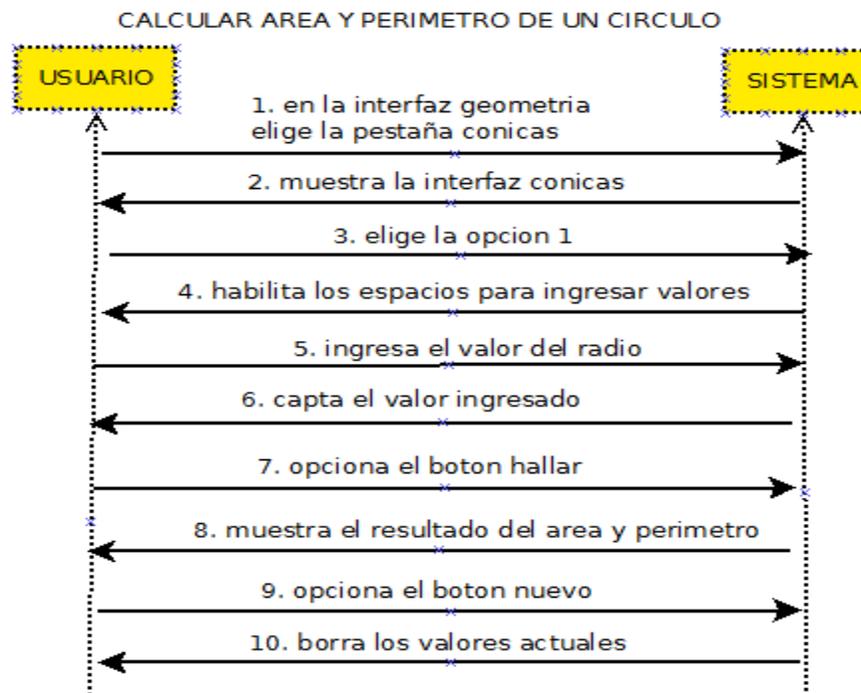


Figura 59: diagrama de secuencia: calcular área y perímetro de un círculo. Mueses, J. (2017). Fuente: autoría propia

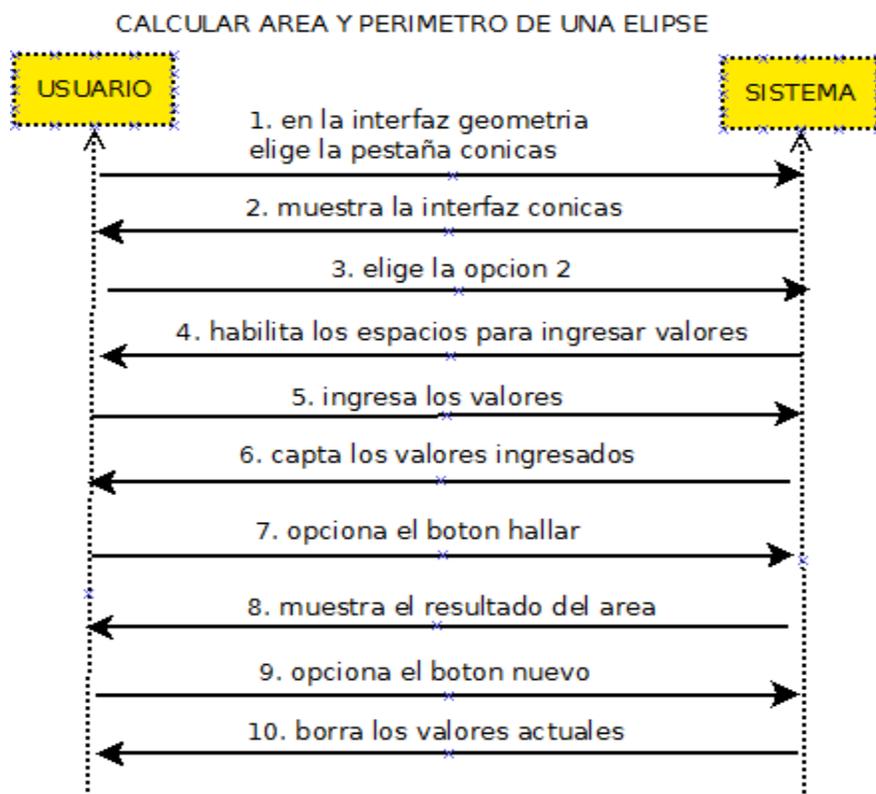


Figura 60: diagrama de secuencia: calcular área y perímetro de una elipse. Mueses, J. (2017). Fuente: autoría propia

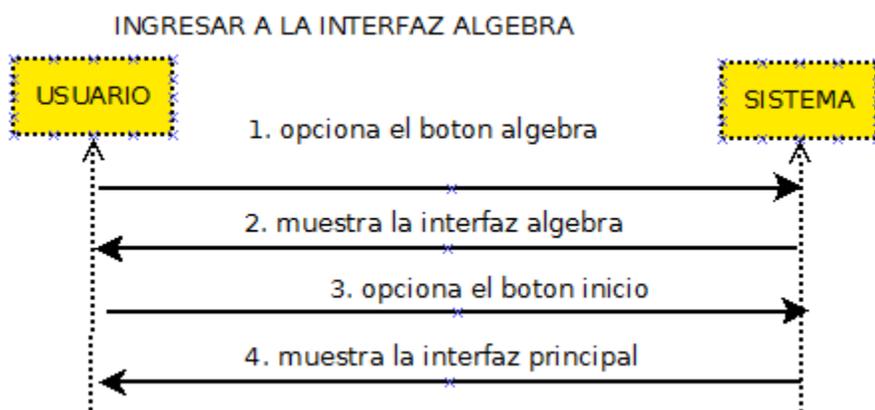


Figura 61: diagrama de secuencia: ingresar a la interfaz algebra. Mueses, J. (2017). Fuente: autoría propia

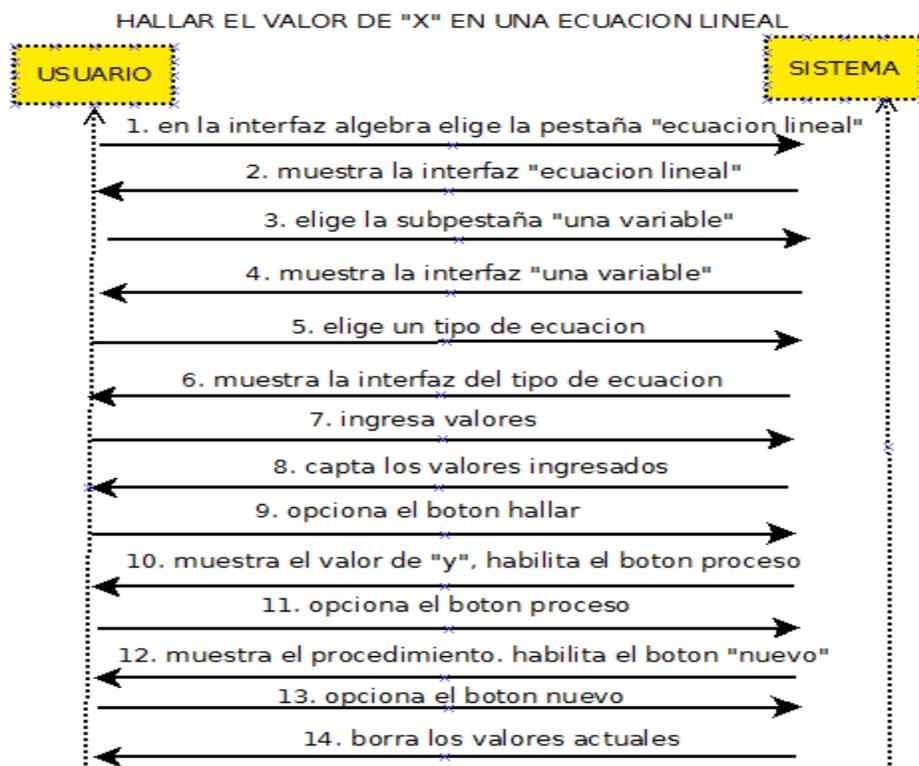


Figura 62: diagrama de secuencia: hallar el valor de "x" en ecuación lineal. Mueses, J. (2017). Fuente: autoría propia

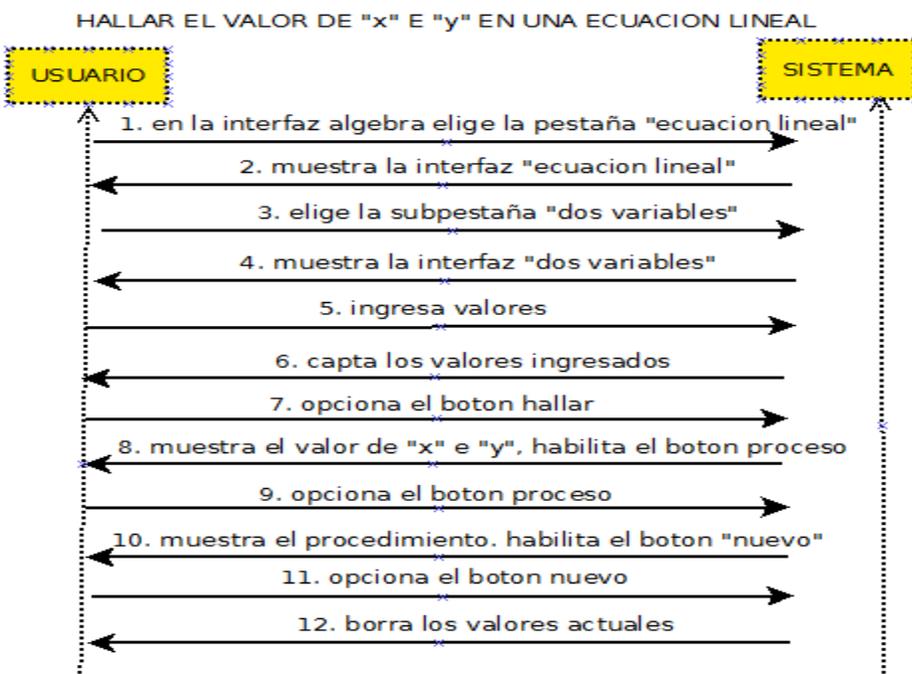


Figura 63: diagrama de secuencia: hallar el valor de "x" e "y" en ecuación lineal. Mueses, J. (2017). Fuente: autoría propia

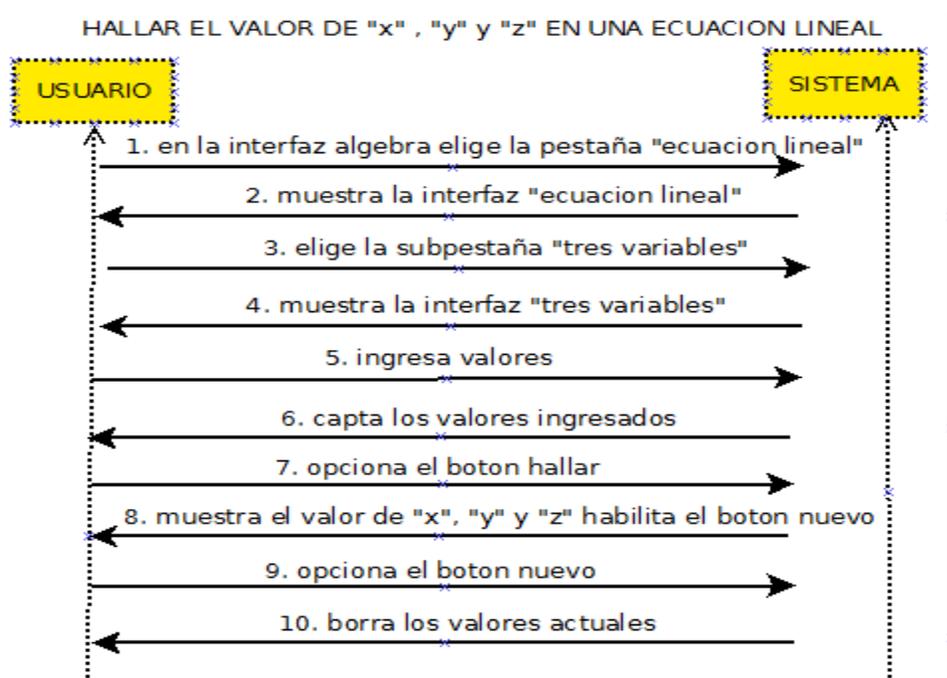


Figura 64: *diagrama de secuencia: hallar el valor de "x", "y" y "z" en ecuación lineal.* Mueses, J. (2017). Fuente: autoría propia

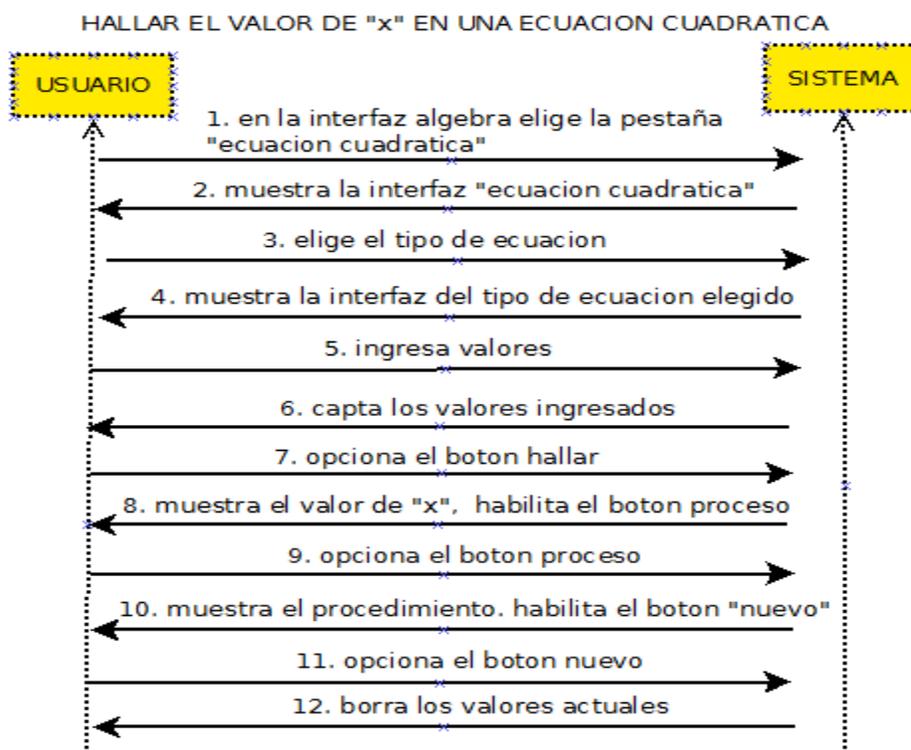


Figura 65: *diagrama de secuencia: hallar el valor de "x" en ecuación cuadrática.* Mueses, J. (2017). Fuente: autoría propia

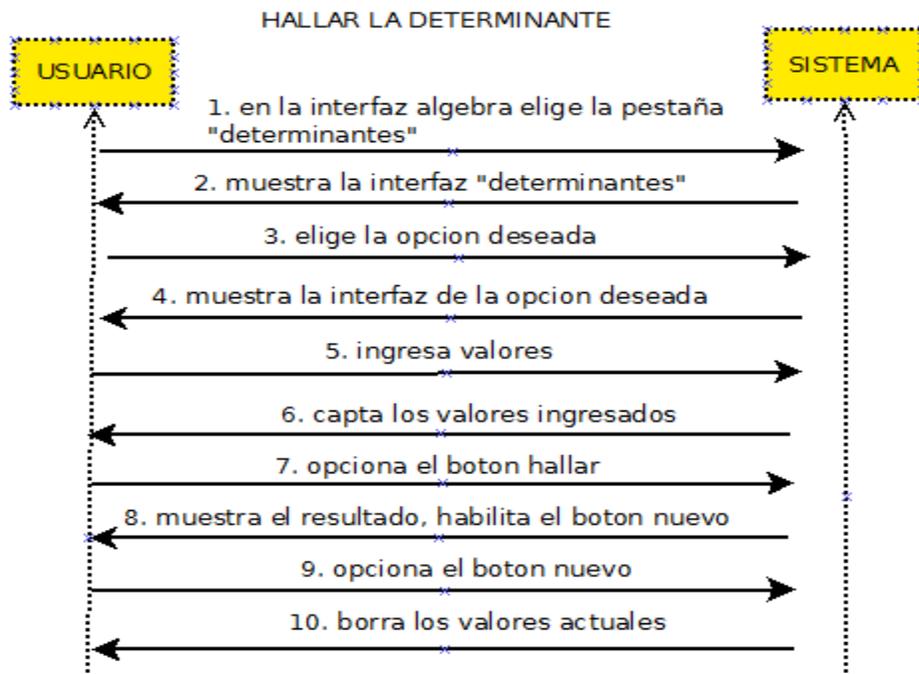


Figura 66: diagrama de secuencia: hallar la determinante de acuerdo a la opción. Mueses, J. (2017). Fuente: autoría propia



Figura 67: diagrama de secuencia: hallar los puntos de corte de acuerdo al caso elegido. Mueses, J. (2017). Fuente: autoría propia

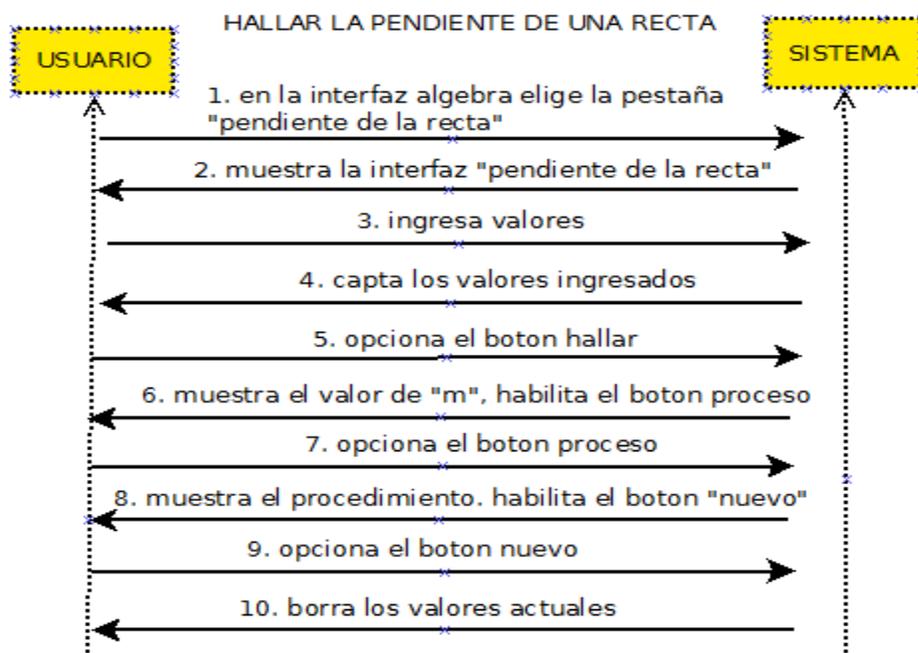


Figura 68: diagrama de secuencia: hallar la pendiente de una recta. Mueses, J. (2017). Fuente: autoría propia



Figura 69: diagrama de secuencia: hallar la ecuación de una recta lineal. Mueses, J. (2017). Fuente: autoría propia

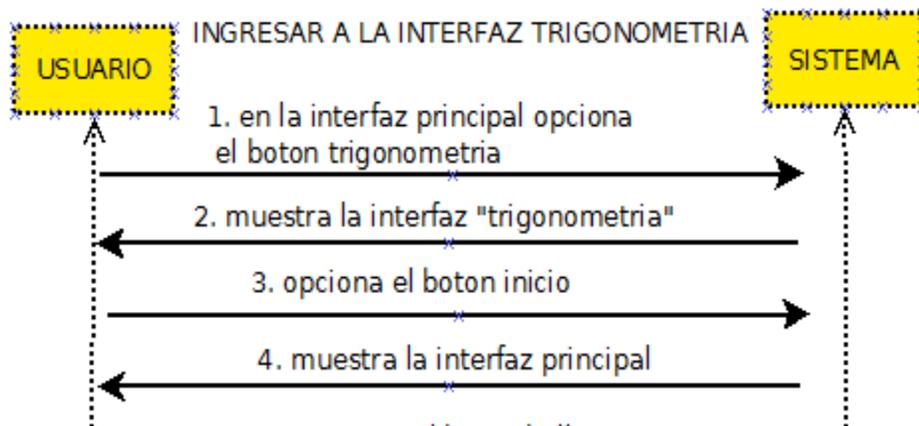


Figura 70: diagrama de secuencia: ingresar a la interfaz trigonometría. Mueses, J. (2017). Fuente: autoría propia

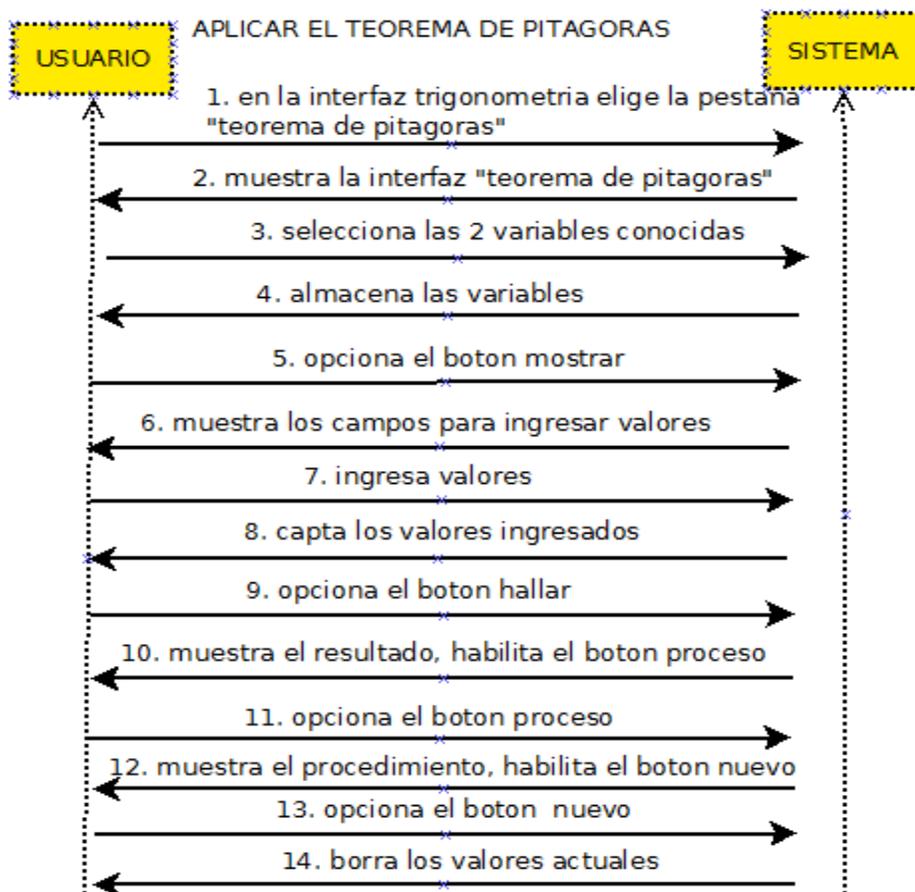


Figura 71: diagrama de secuencia: usar el teorema de Pitágoras. Mueses, J. (2017). Fuente: autoría propia

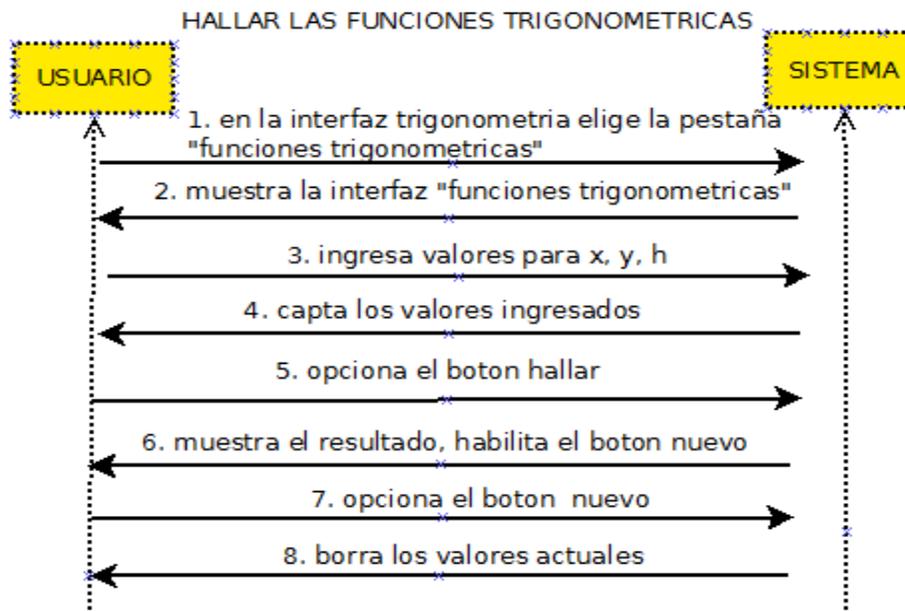


Figura 72: diagrama de secuencia: halla el valor de las 6 funciones trigonométricas. Mueses, J. (2017). Fuente: autoría propia

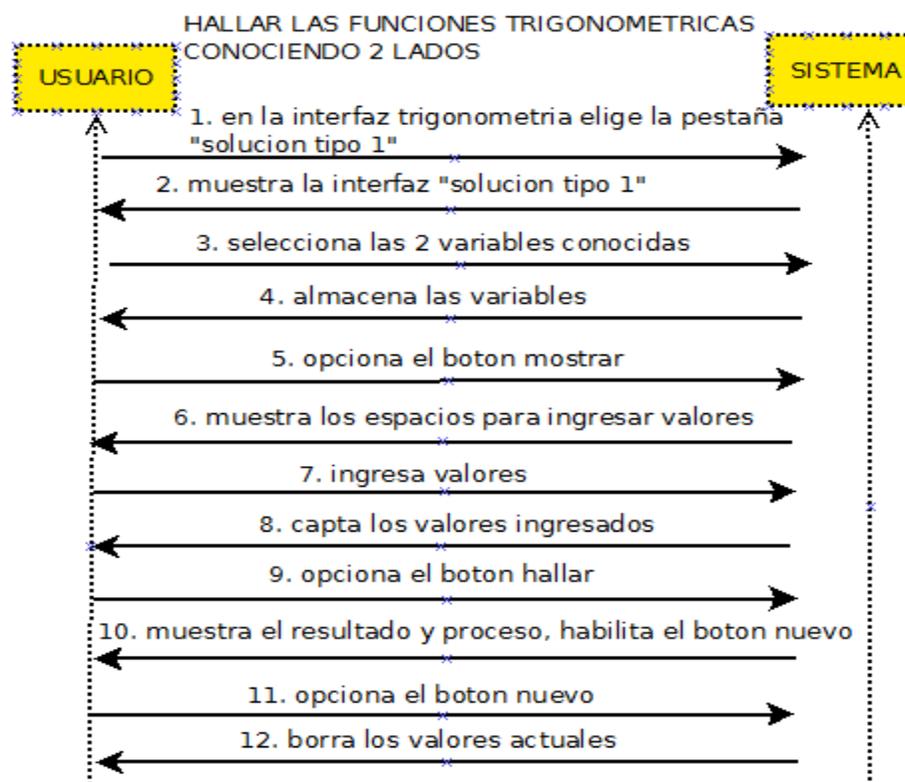


Figura 73: diagrama de secuencia: hallar las funciones trigonométricas conociendo 2 lados. Mueses, J. (2017). Fuente: autoría propia



Figura 74: diagrama de secuencia: hallar las funciones trigonométricas conociendo el valor de 1 lado y 1 ángulo. Mueses, J. (2017). Fuente: autoría propia

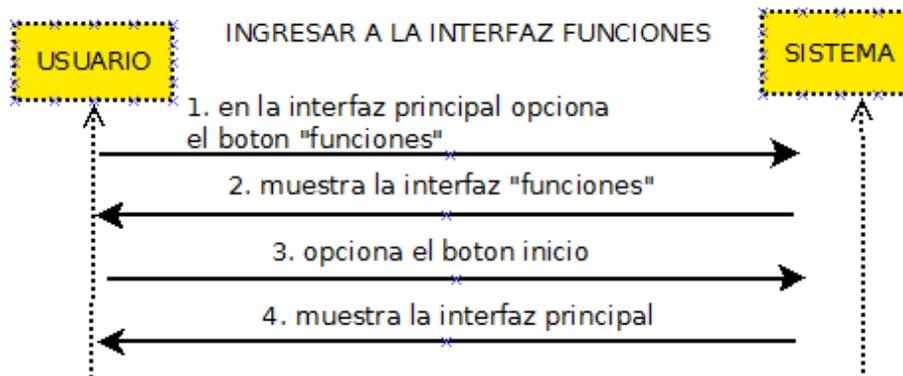


Figura 75: diagrama de secuencia: ingresar a la interfaz funciones. Mueses, J. (2017). Fuente: autoría propia

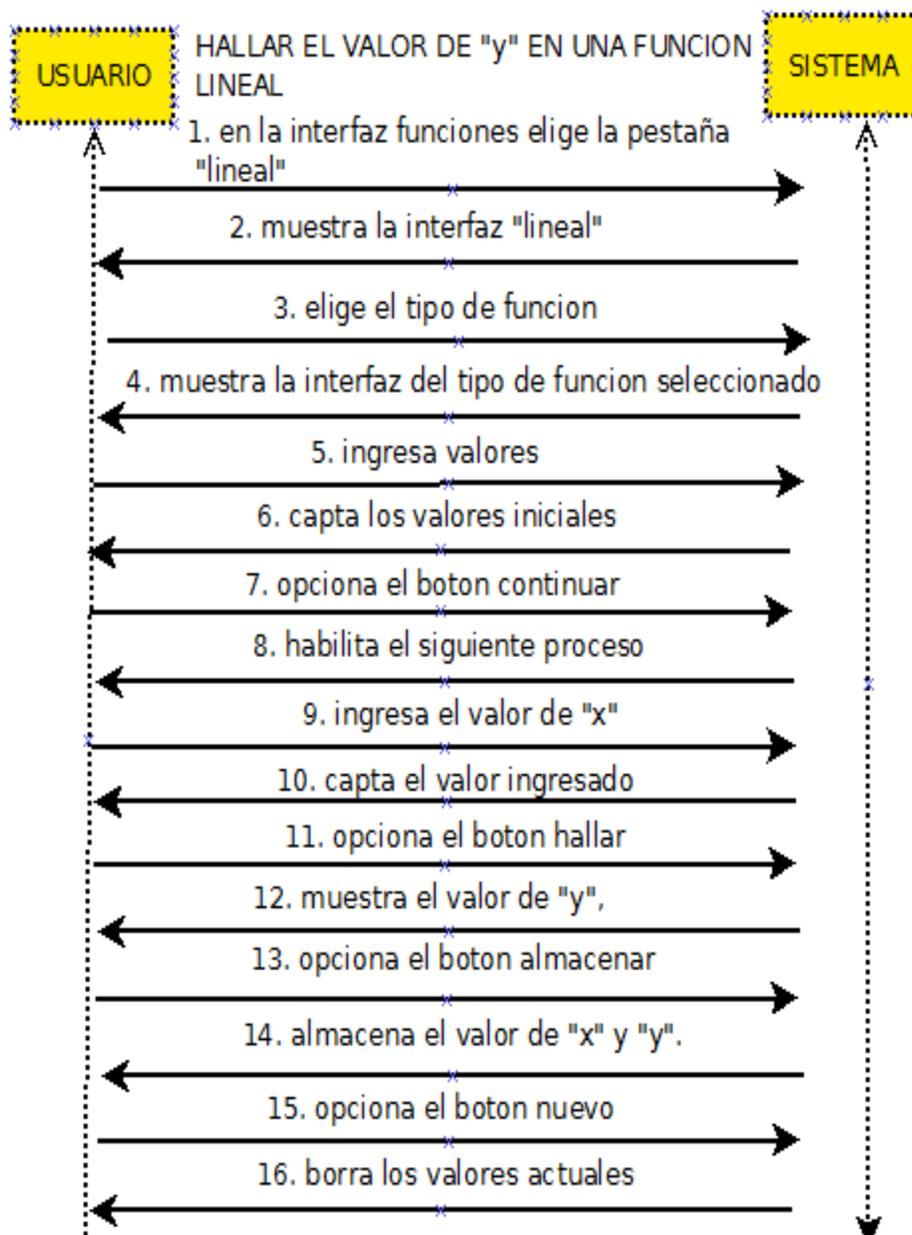


Figura 76: diagrama de secuencia: hallar el valor de "y" en una función lineal. Mueses, J. (2017). Fuente: autoría propia

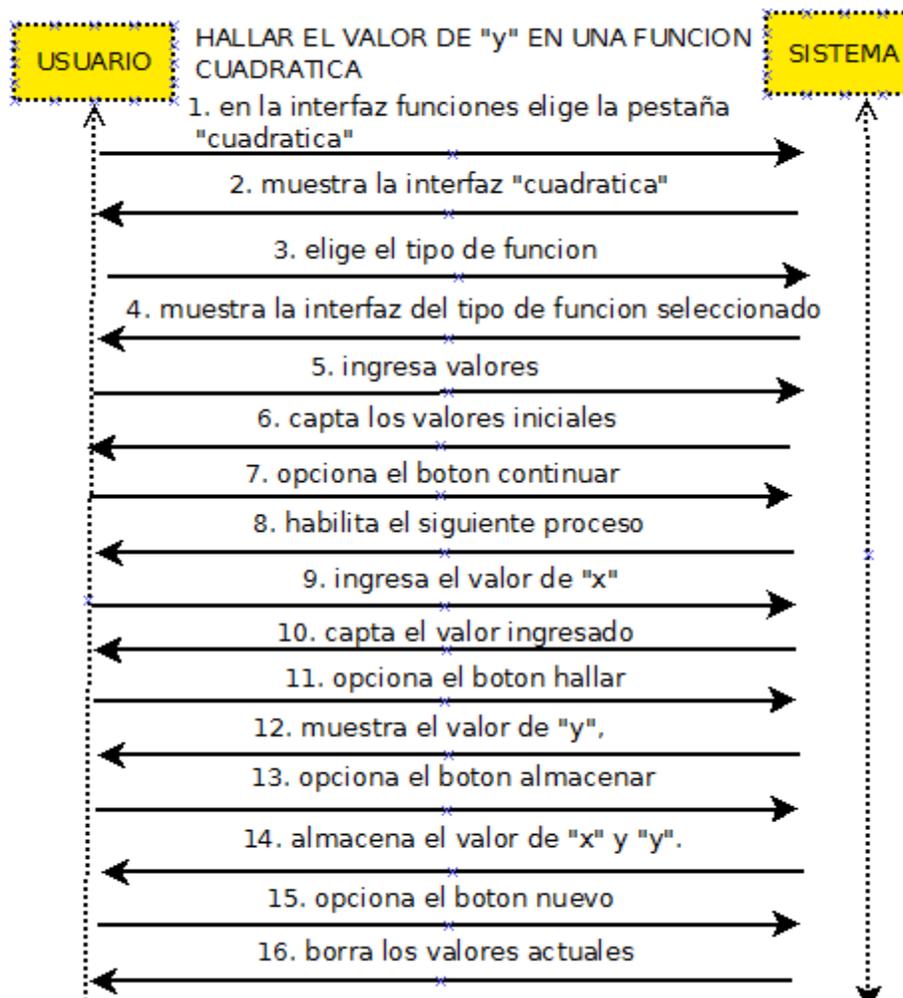


Figura 77: diagrama de secuencia: hallar el valor de "y" en una función cuadrática. Mueses, J. (2017). Fuente: autoría propia

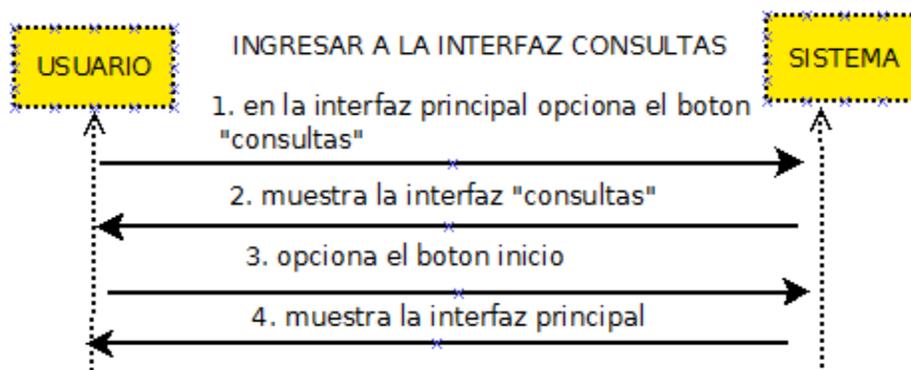


Figura 78: diagrama de secuencia: ingresar a la interfaz consultas. Mueses, J. (2017). Fuente: autoría propia

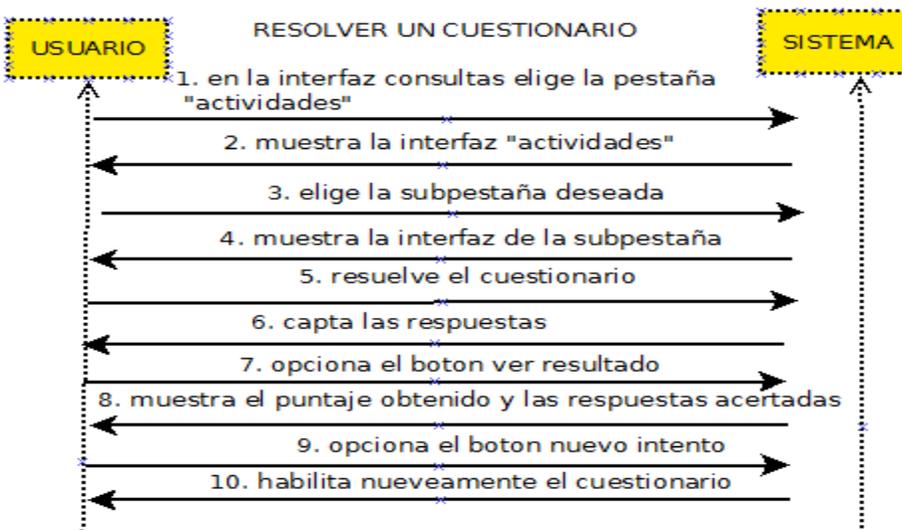


Figura 79: *diagrama de secuencia: resolver un cuestionario.* Mueses, J. (2017). Fuente: autoría propia

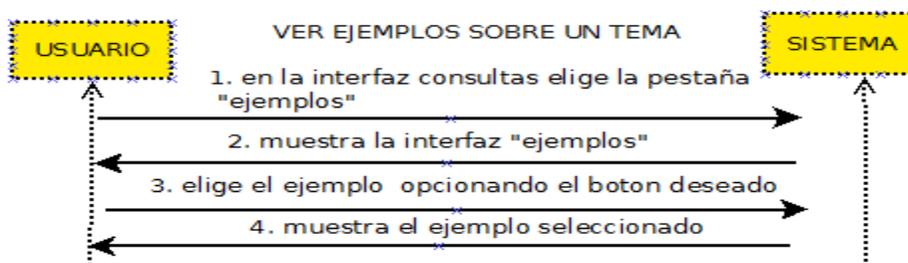


Figura 80: *diagrama de secuencia: ver ejemplos.* Mueses, J. (2017). Fuente: autoría propia

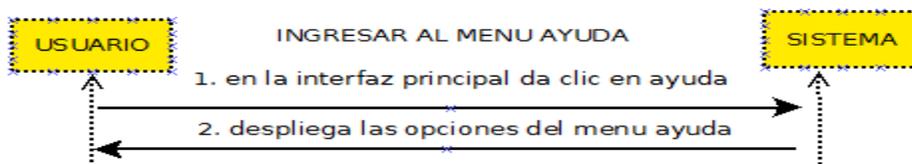


Figura 81: *diagrama de secuencia: ingresar al menú ayuda.* Mueses, J. (2017). Fuente: autoría propia

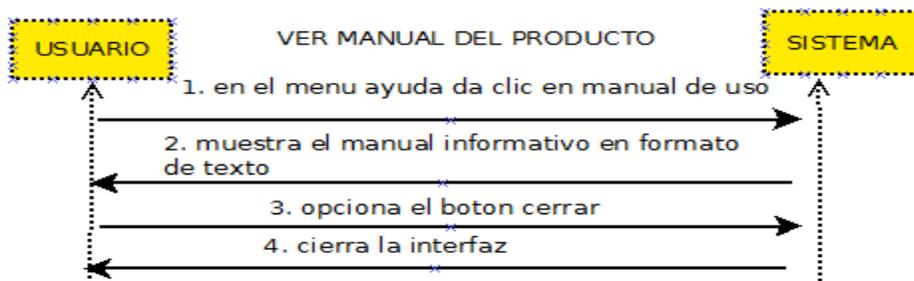


Figura 80: *diagrama de secuencia: ver el manual de uso.* Mueses, J. (2017). Fuente: autoría propia

Diseño de interfaces gráficas.



Figura 81: interfaz gráfica inicio del programa. Mueses, J. (2017). Fuente: autoría propia

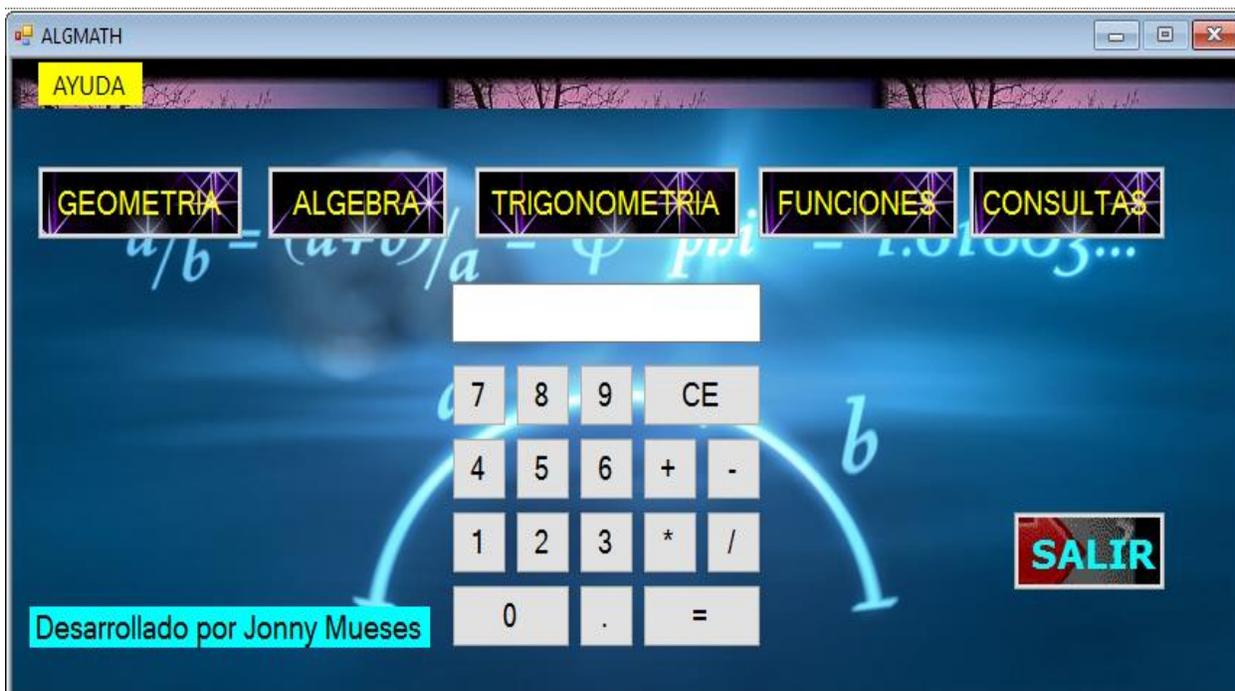


Figura 82: interfaz gráfica ventana principal. Mueses, J. (2017). Fuente: autoría propia

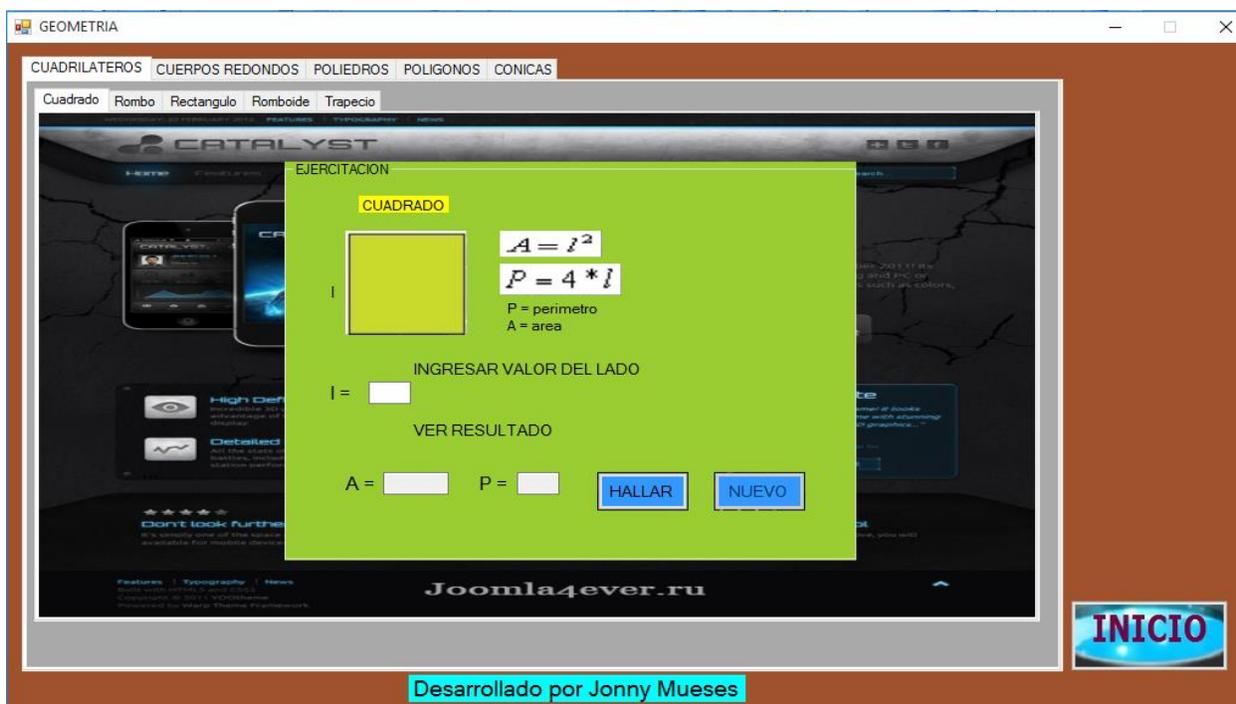


Figura 83: interfaz gráfica geometría, pestaña (cuadrilátero), subpestaña cuadrado. Mueses, J. (2017). Fuente: autoría propia

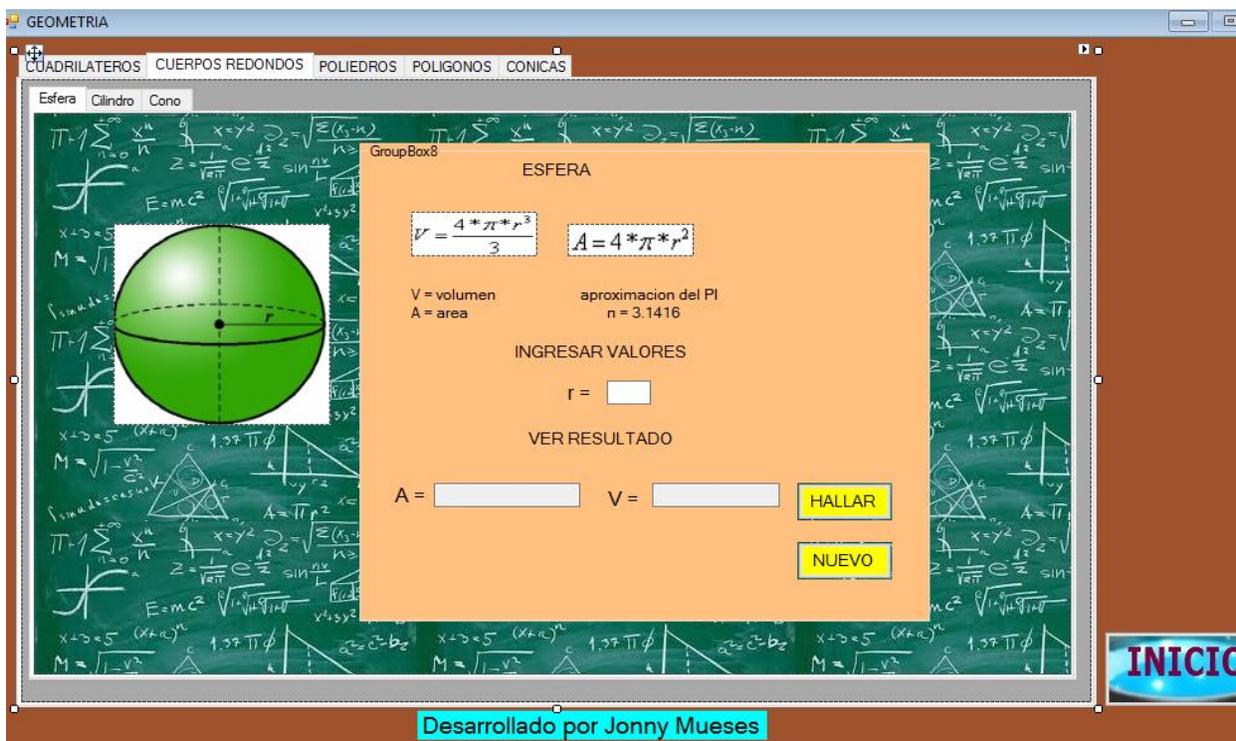


Figura 84: interfaz gráfica geometría, pestaña (cuerpos redondos), subpestaña esfera. Mueses, J. (2017). Fuente: autoría propia

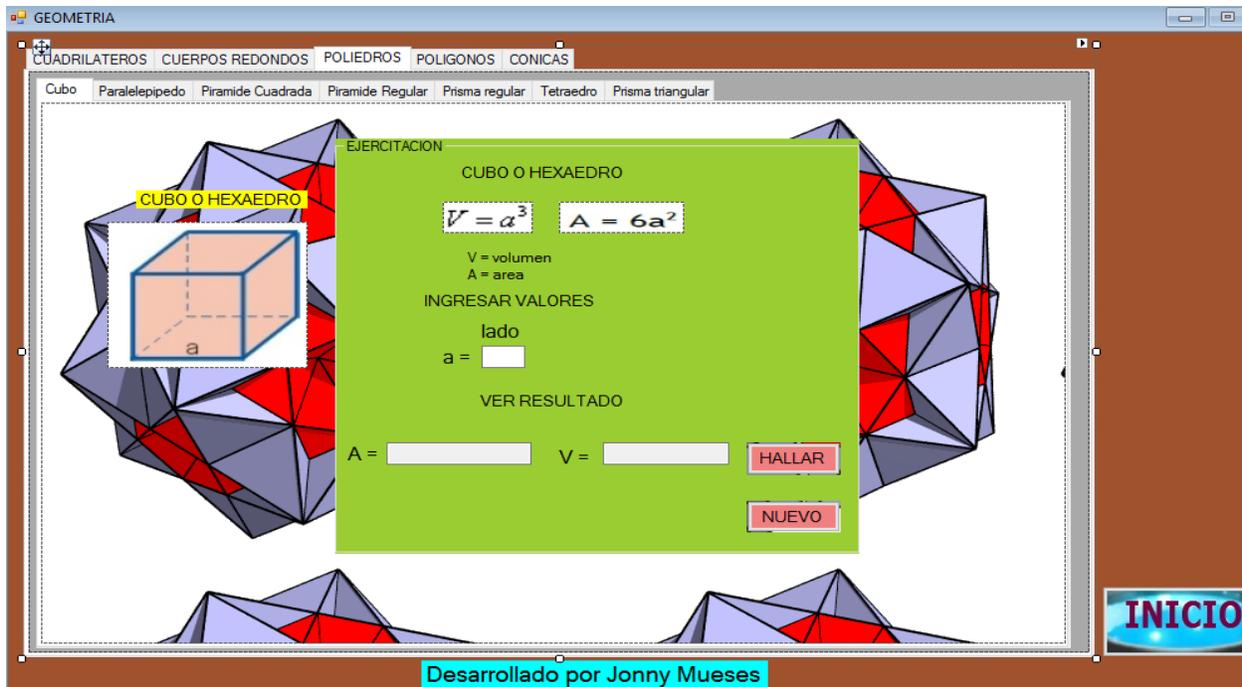


Figura 85: interfaz gráfica geometría, pestaña (poliedros), subpestaña cubo. Mueses, J. (2017). Fuente: autoría propia



Figura 86: interfaz gráfica geometría, pestaña (poligonos). Mueses, J. (2017). Fuente: autoría propia

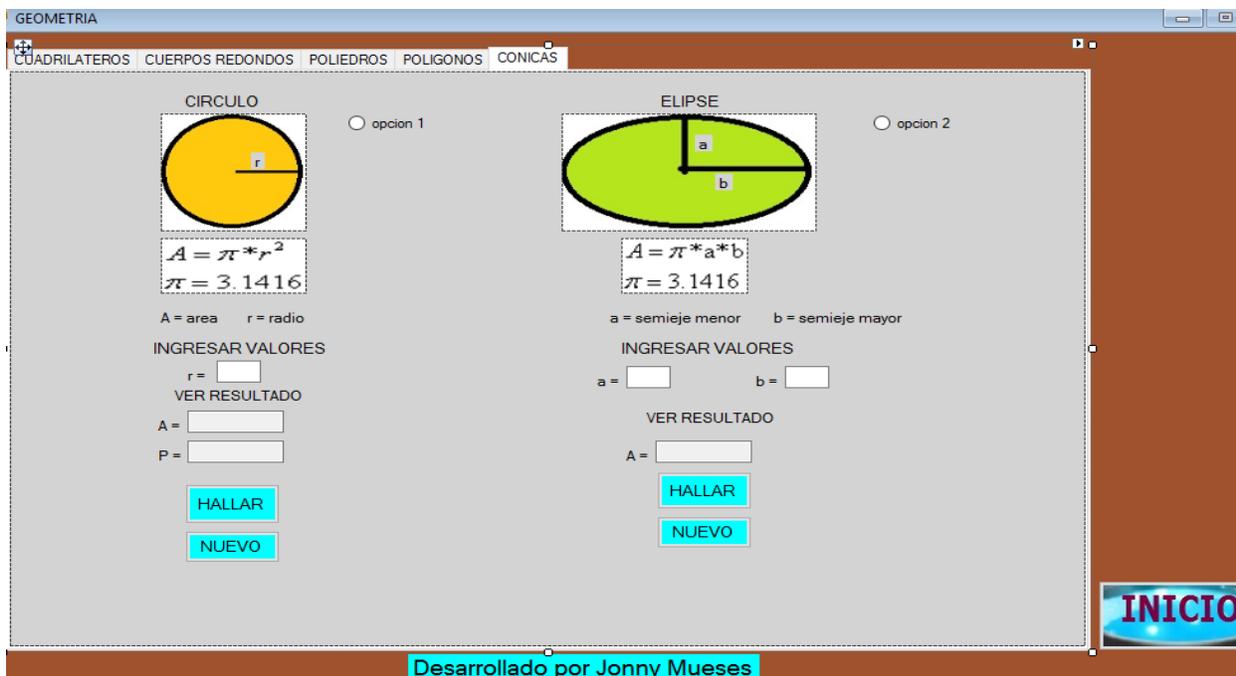


Figura 87: interfaz gráfica geometría, pestaña (cónicas). Mueses, J. (2017). Fuente: autoría propia

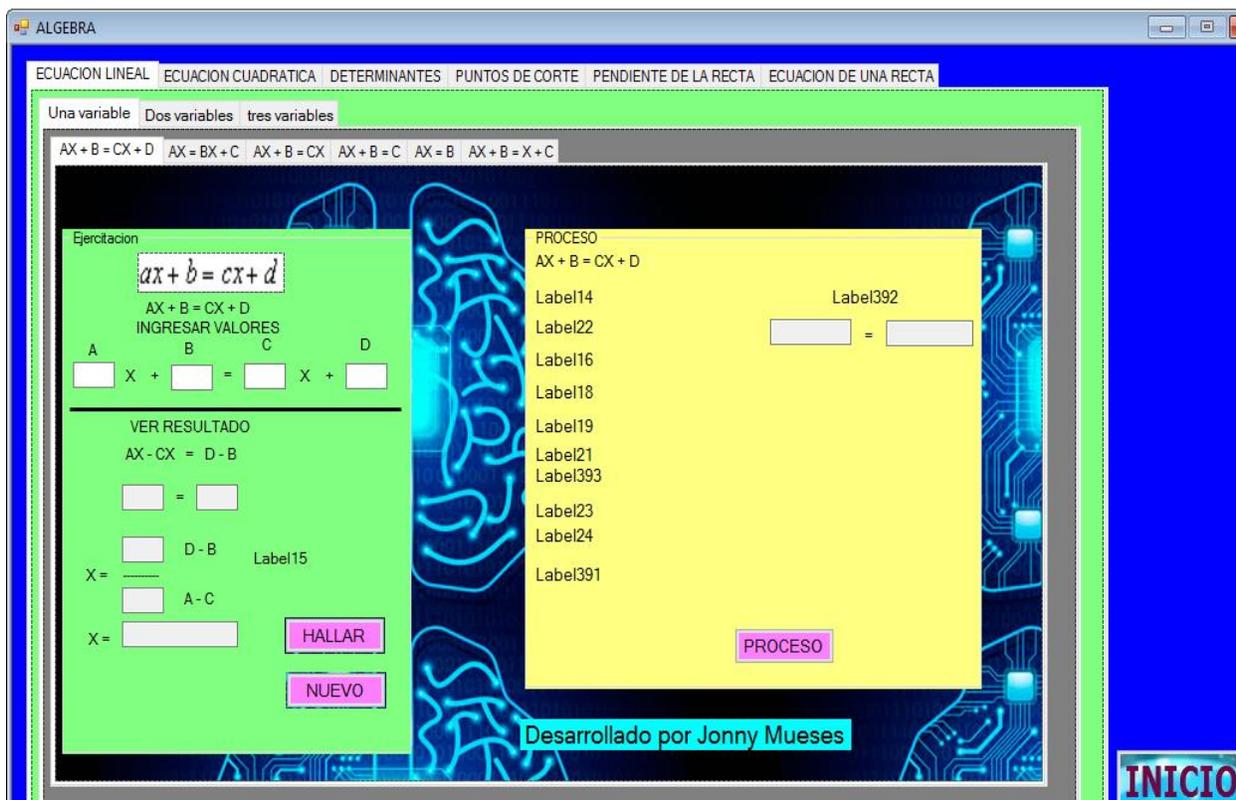


Figura 88: interfaz gráfica algebra, pestaña (ecuación lineal), subpestaña una variable. Mueses, J. (2017). Fuente: autoría propia

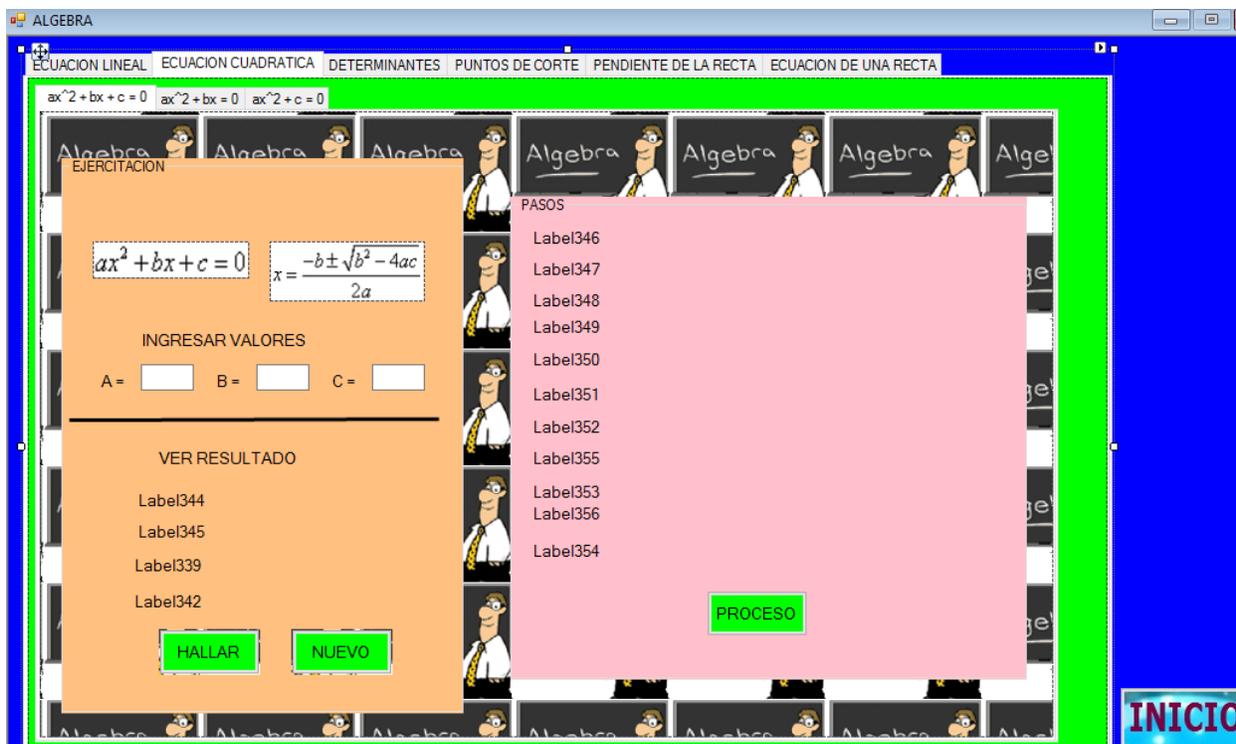


Figura 89: interfaz gráfica algebra, pestaña (ecuación cuadrática). Mueses, J. (2017). Fuente: autoría propia

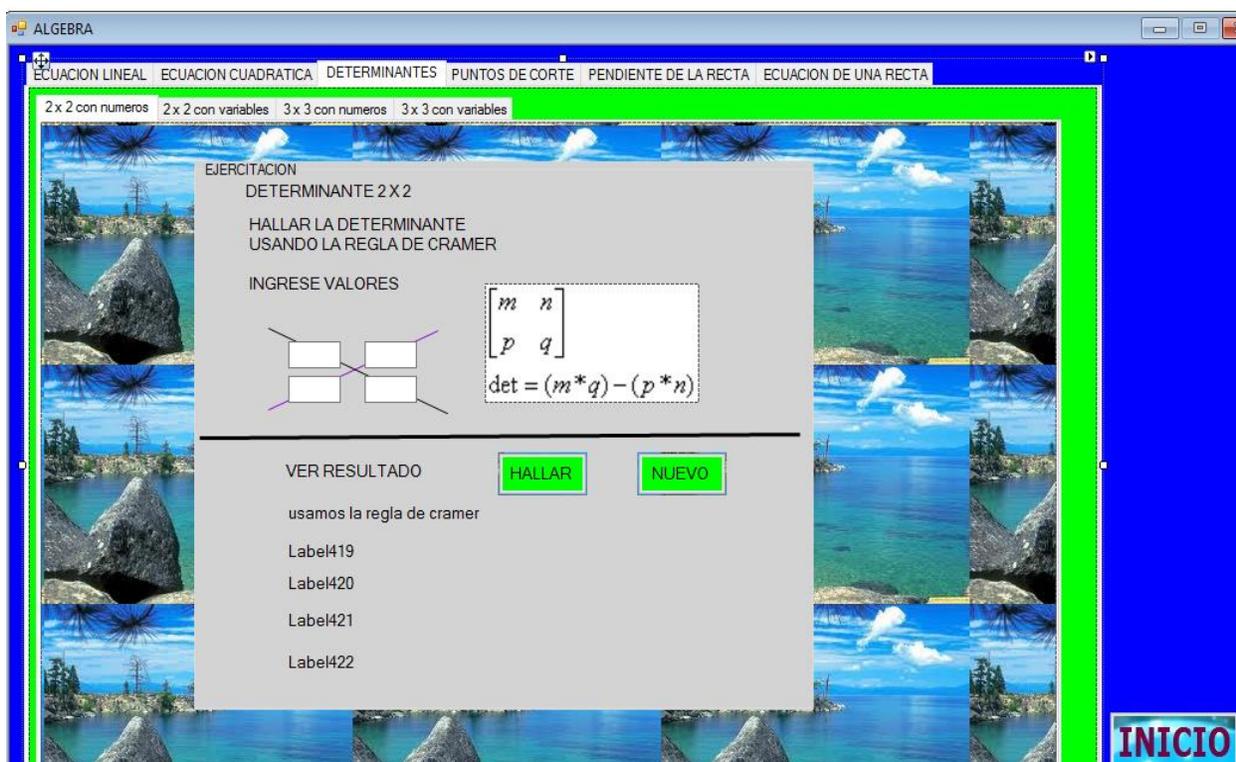


Figura 90: interfaz gráfica algebra, pestaña (determinantes), subpestaña 2x2 con números. Mueses, J. (2017). Fuente: autoría propia

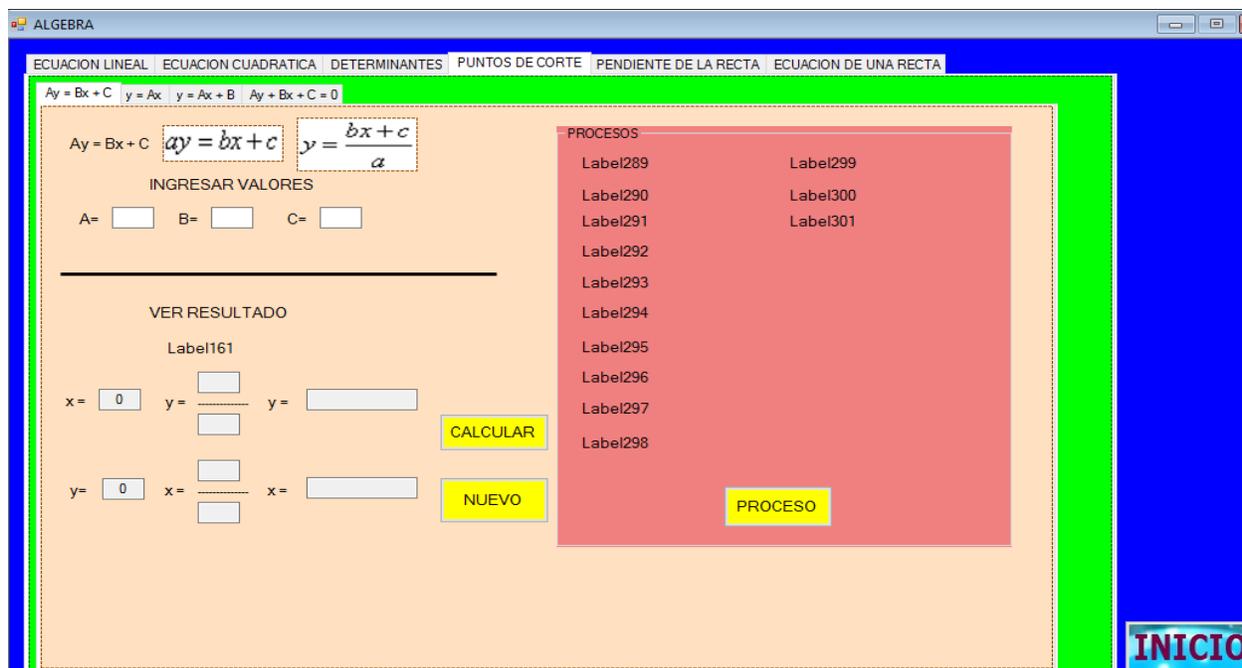


Figura 91: interfaz gráfica algebra, pestaña (puntos de corte). Mueses, J. (2017). Fuente: autoría propia

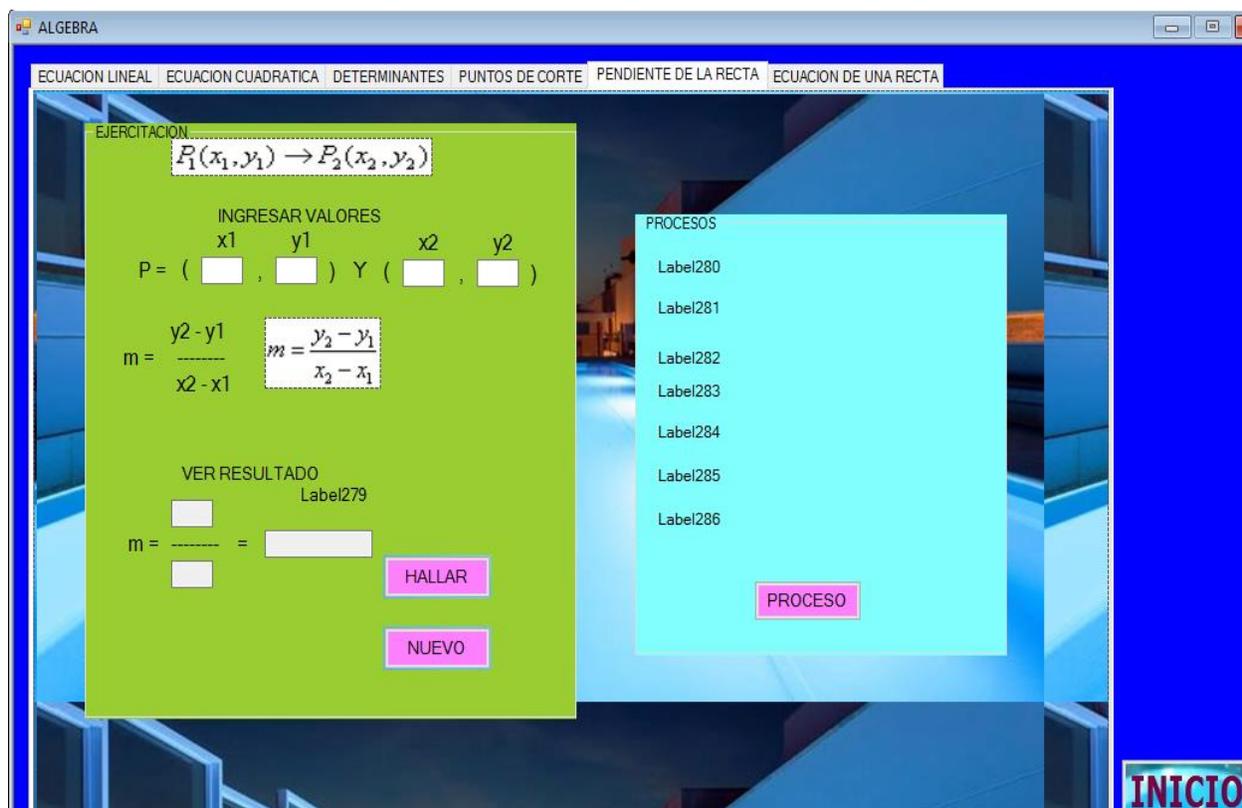


Figura 92: interfaz gráfica algebra, pestaña (pendiente de la recta). Mueses, J. (2017). Fuente: autoría propia

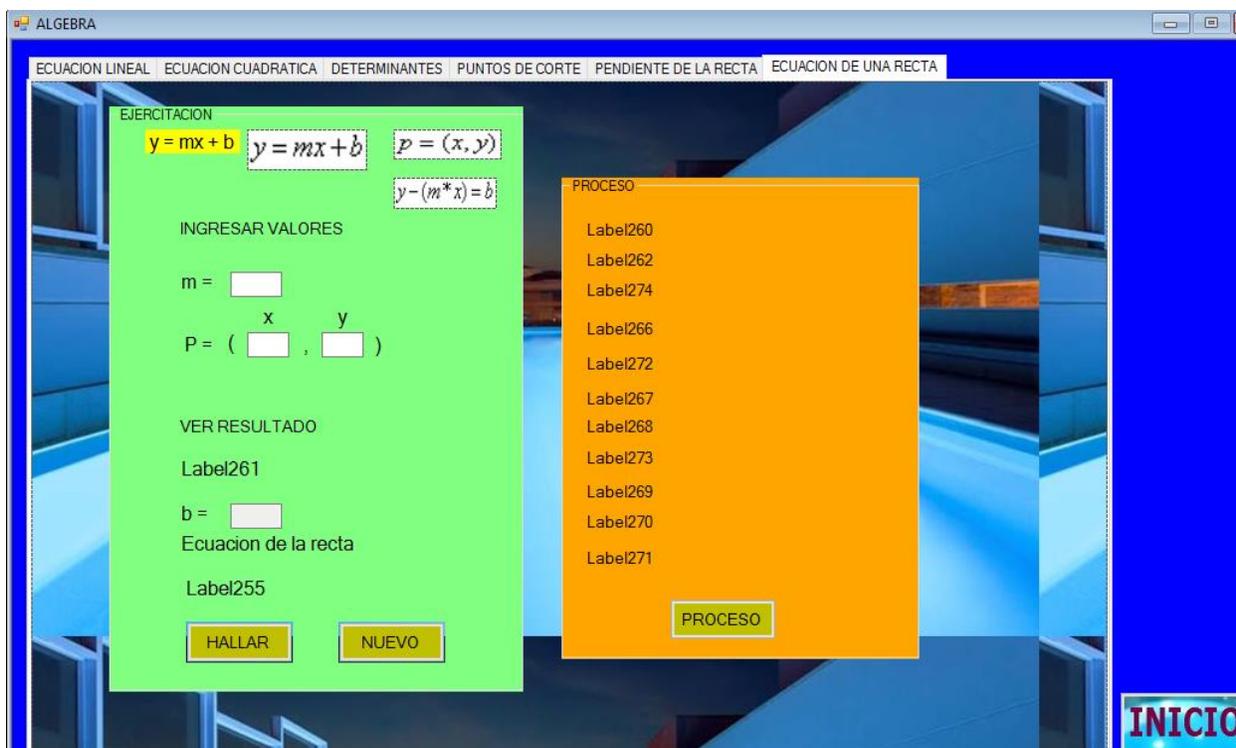


Figura 93: interfaz gráfica algebra, pestaña (ecuación de una recta). Mueses, J. (2017). Fuente: autoría propia

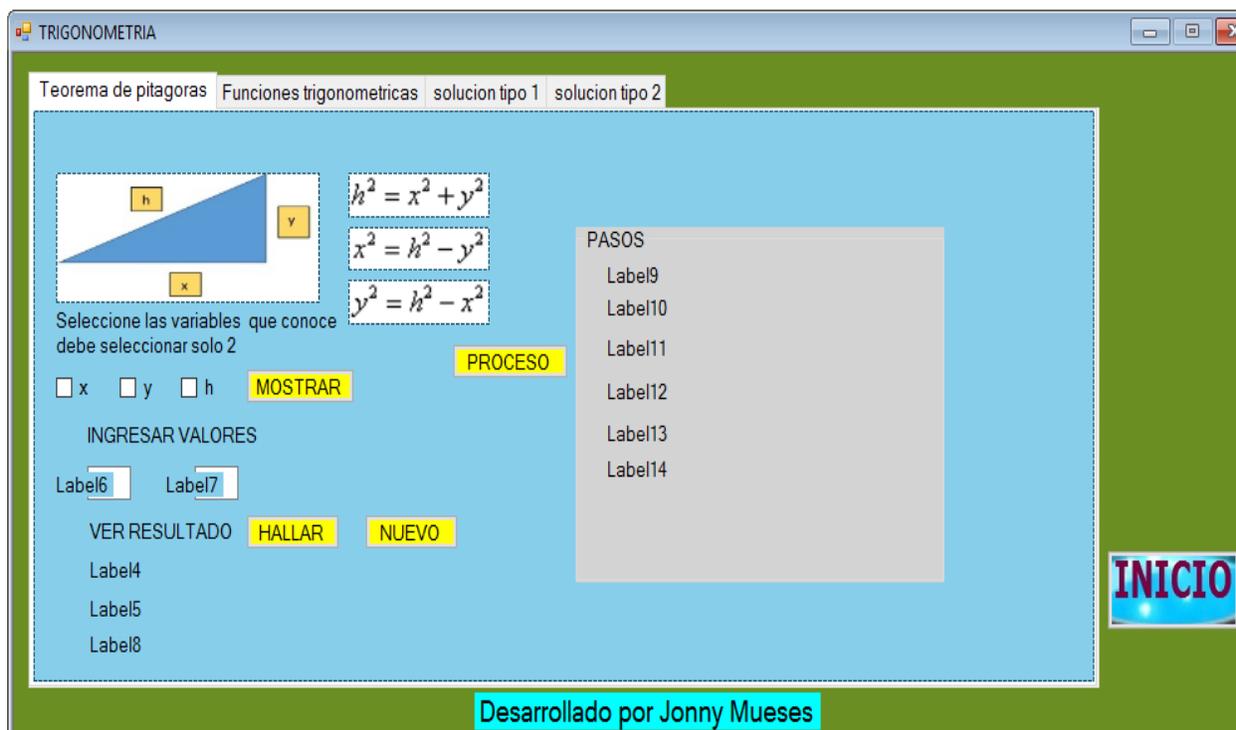


Figura 94: interfaz gráfica trigonometría, pestaña (Teorema de Pitágoras). Mueses, J. (2017). Fuente: autoría propia

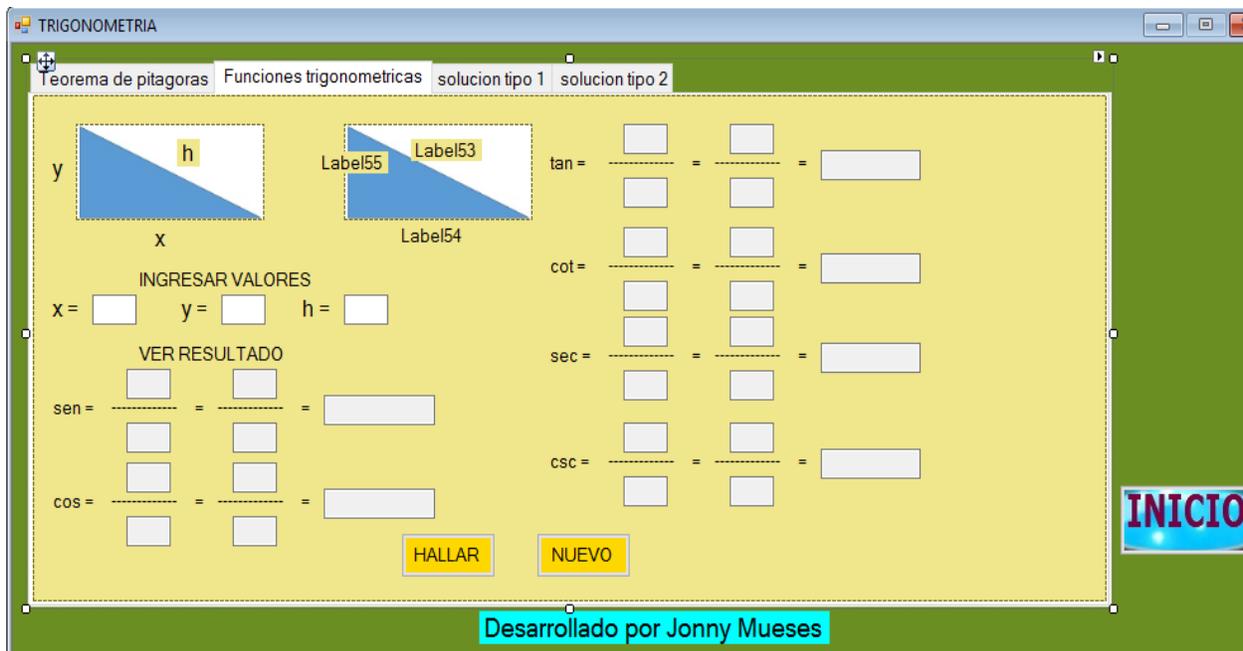


Figura 95: interfaz gráfica trigonometría, pestaña (funciones trigonométricas). Mueses, J. (2017). Fuente: autoría propia

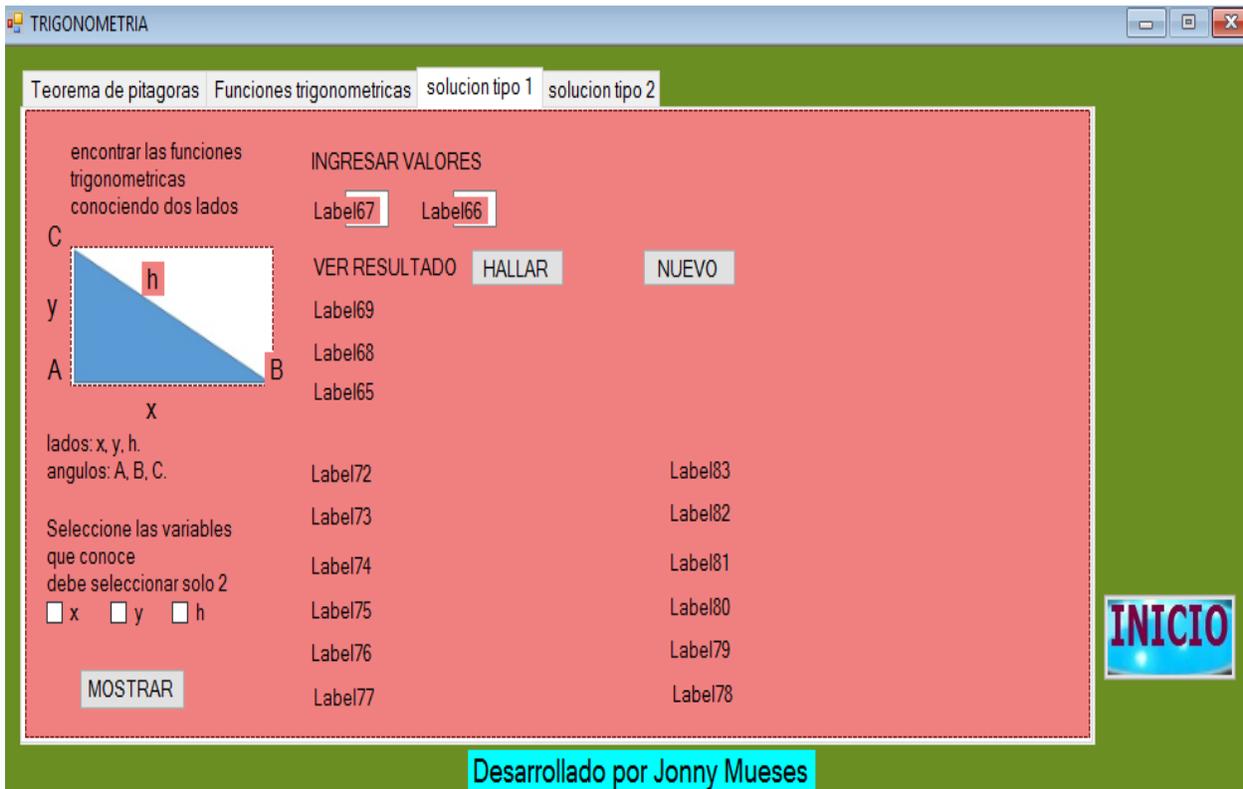


Figura 96: interfaz gráfica trigonometría, pestaña (solución tipo 1). Mueses, J. (2017). Fuente: autoría propia



Figura 97: interfaz gráfica trigonometría, pestaña (solución tipo 2). Mueses, J. (2017). Fuente: autoría propia

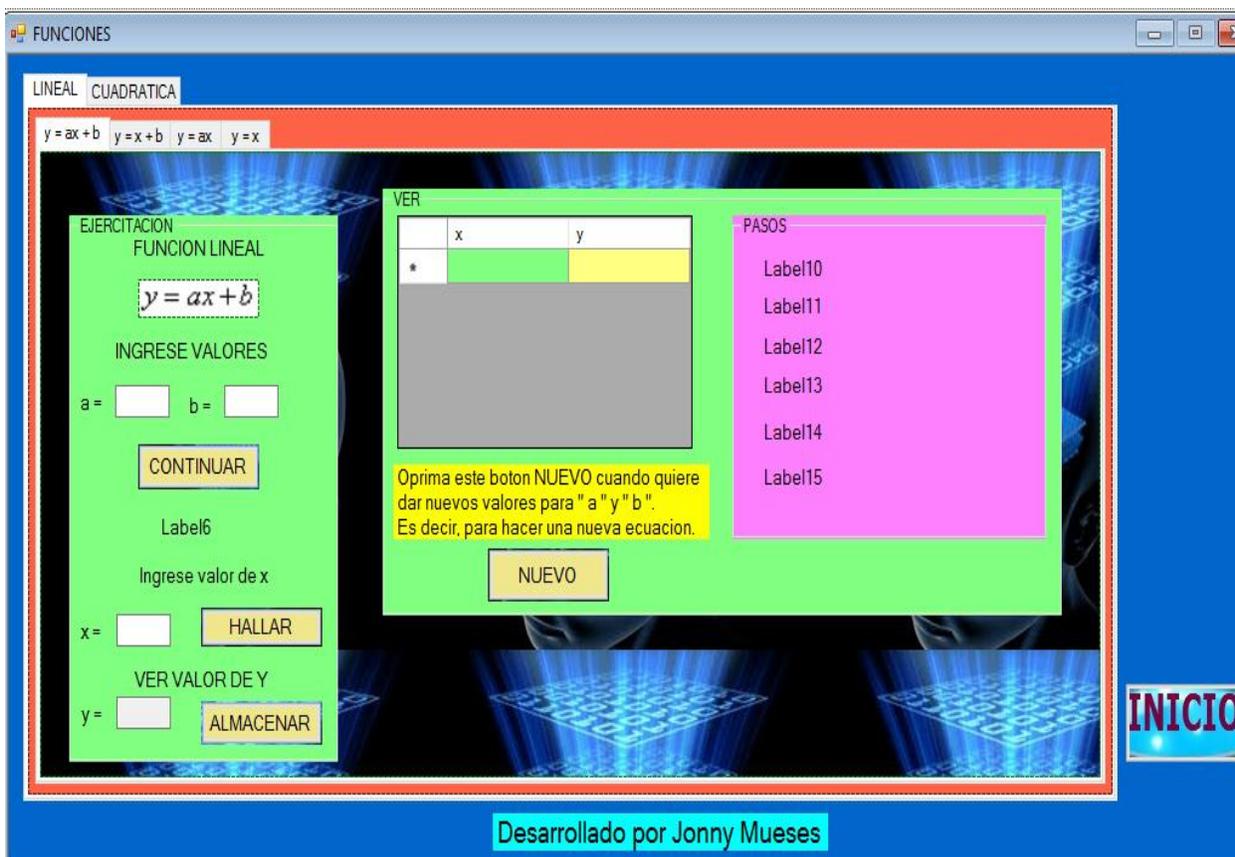


Figura 98: interfaz gráfica funciones, pestaña (lineal), subpestaña $y=ax+b$. Mueses, J. (2017). Fuente: autoría propia

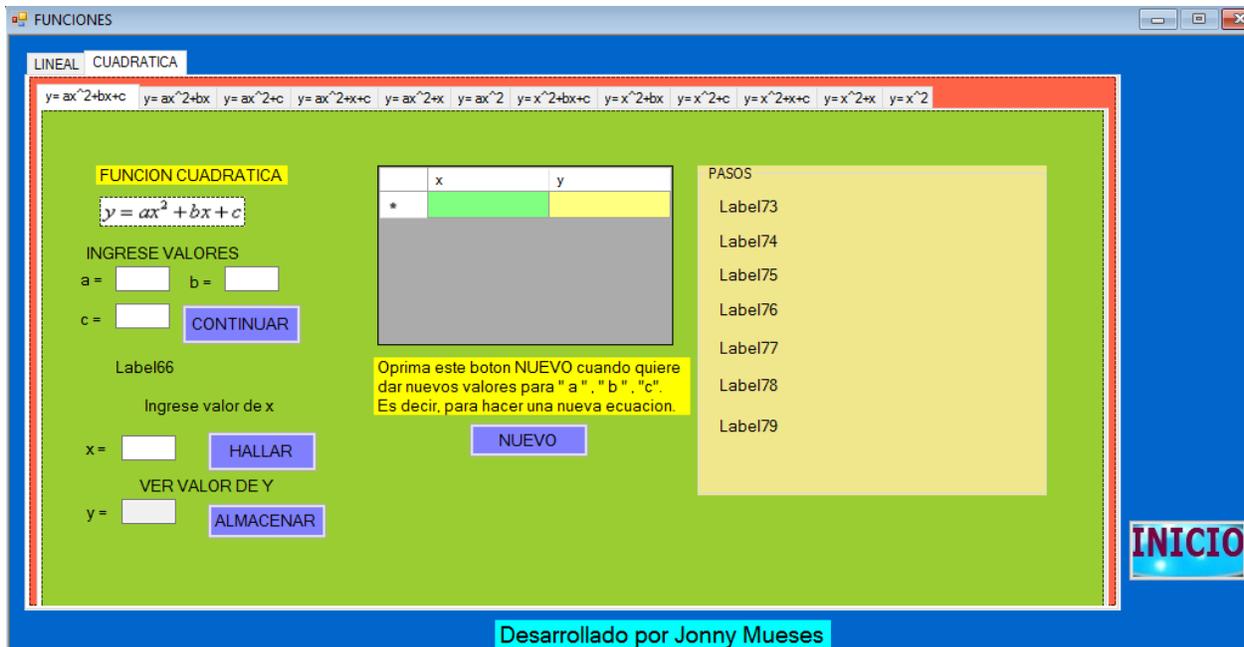


Figura 99: interfaz gráfica funciones, pestaña (cuadrática). Mueses, J. (2017). Fuente: autoría propia

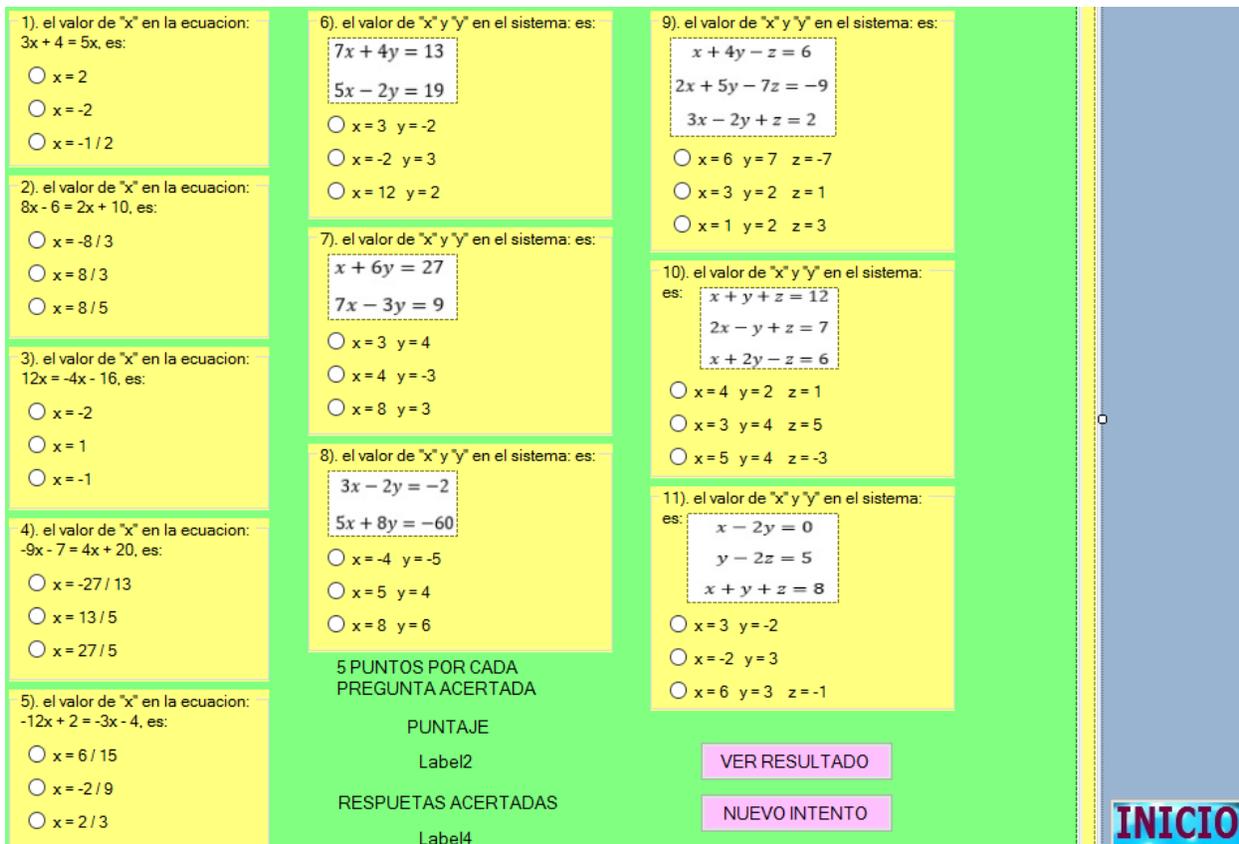


Figura 100: interfaz gráfica consultas, pestaña (actividades). Mueses, J. (2017). Fuente: autoría propia

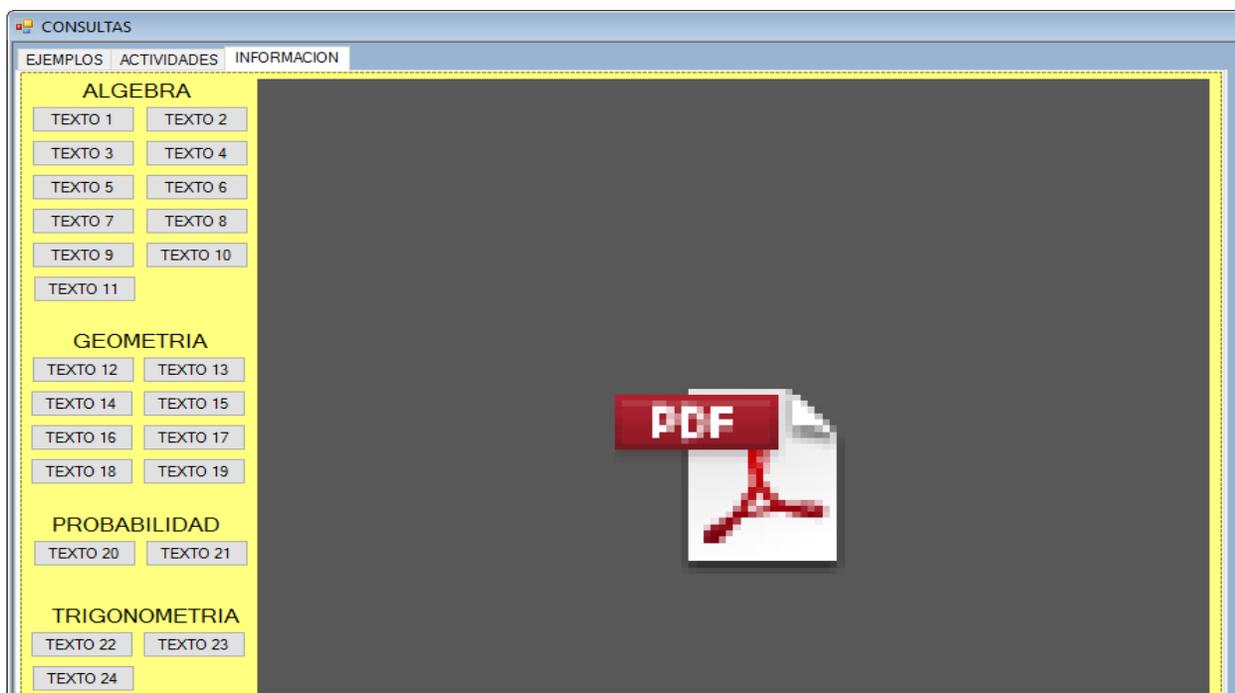


Figura 101: interfaz gráfica consultas, pestaña (información). Mueses, J. (2017). Fuente: autoría propia

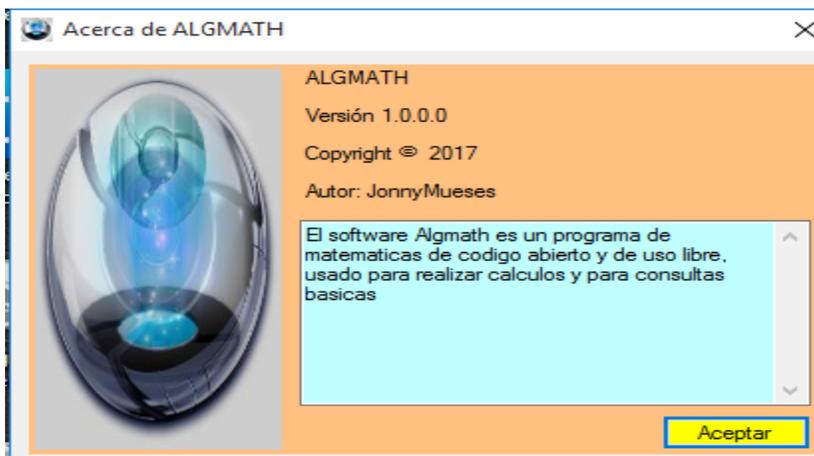


Figura 102: interfaz gráfica acerca de. Mueses, J. (2017). Fuente: autoría propia

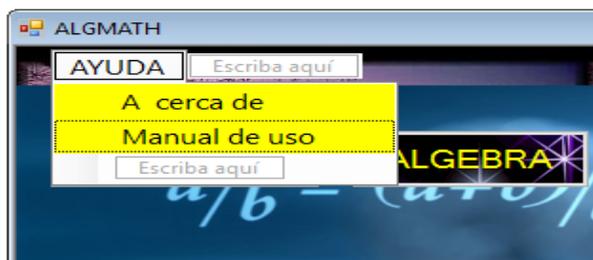


Figura 103: interfaz gráfica menú ayuda, J. (2017). Fuente: autoría propia

Diseño arquitectónico.



Figura 104: *diagrama arquitectónico usuario-sistema*. Mueses, J. (2017). Fuente: autoría propia

Diagrama de relación.

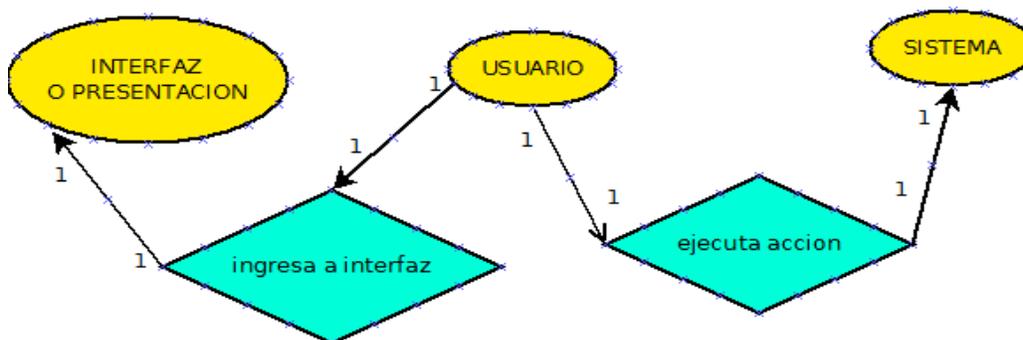


Figura 105: *diagrama de relación interfaz-usuario-sistema*. Mueses, J. (2017). Fuente: autoría propia

De acuerdo al diagrama de relación, entre el usuario y la interfaz hay una relación de 1 a 1, donde el usuario solo puede ingresar a una sola interfaz, y la relación entre usuario y sistema es 1 a 1, donde el usuario solo puede ejecutar una sola acción en el sistema, y el sistema muestra un solo resultado, de acuerdo a la acción ejecutada por el usuario.

11.8 Etapa de codificación

Luego de haber diseñado las interfaces graficas del software, y de especificar como debe funcionar de acuerdo a la interfaz gráfica, se procedió a la generación del código (**ver anexo**) de

manera correcta, teniendo en cuenta la estructura del diseño. Cada interfaz gráfica cuenta con su escritura de código y para que funciones de acuerdo a lo establecido, se hacían depuraciones en tiempo real.

11. 9 Etapa de pruebas

En esta etapa se lleva a cabo la depuración y verificación del código escrito para la funcionalidad de cada interfaz y del programa en general, se comprueba que no haya errores ni los más mínimos tanto en el código como en las interfaces, y módulos.

Se compone de 2 pruebas, las pruebas iniciales, donde se analiza el funcionamiento del software bajo la plataforma del lenguaje usado para su desarrollo (visual Basic 2010), las pruebas finales, donde se comprueba la funcionalidad del software luego de generarse el instalador (aplicación).

Pruebas iniciales (bajo la plataforma de Visual Basic 2010):



Figura 106: ejecutando el programa. Mueses, J. (2017). Fuente: autoría propia



Figura 107: interfaz principal del programa. Mueses, J. (2017). Fuente: autoría propia

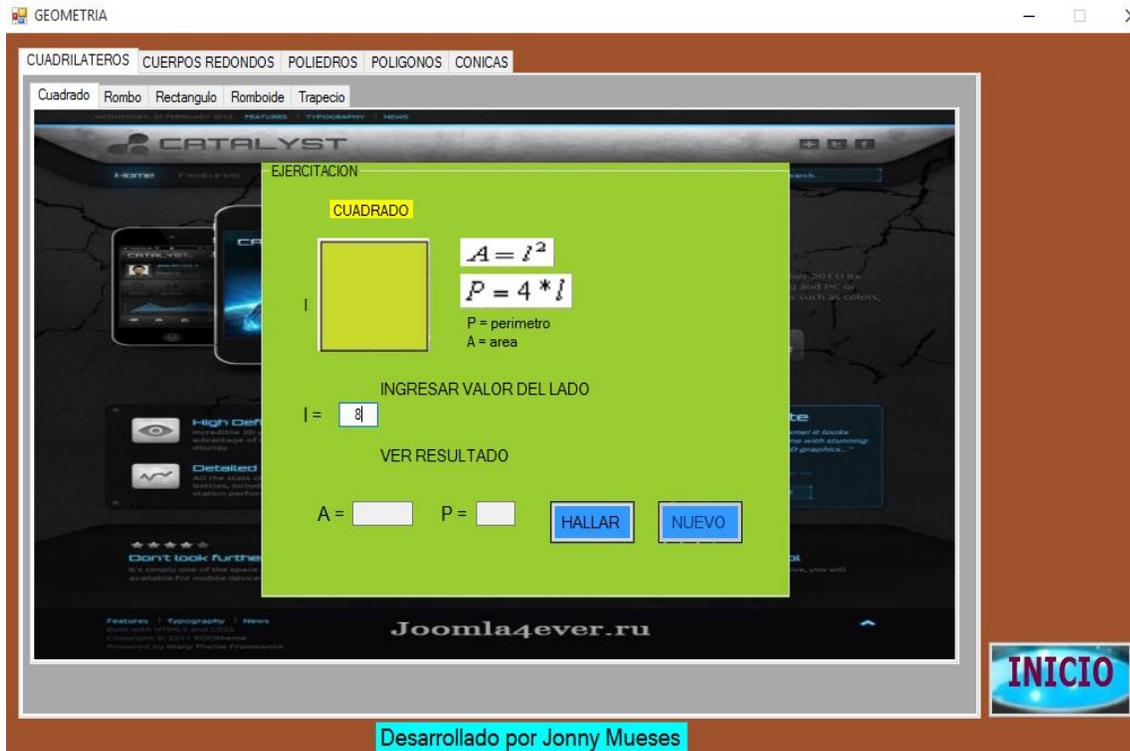


Figura 108: ingresando valor para hallar el área y perímetro del cuadrado. Mueses, J. (2017). Fuente: autoría propia

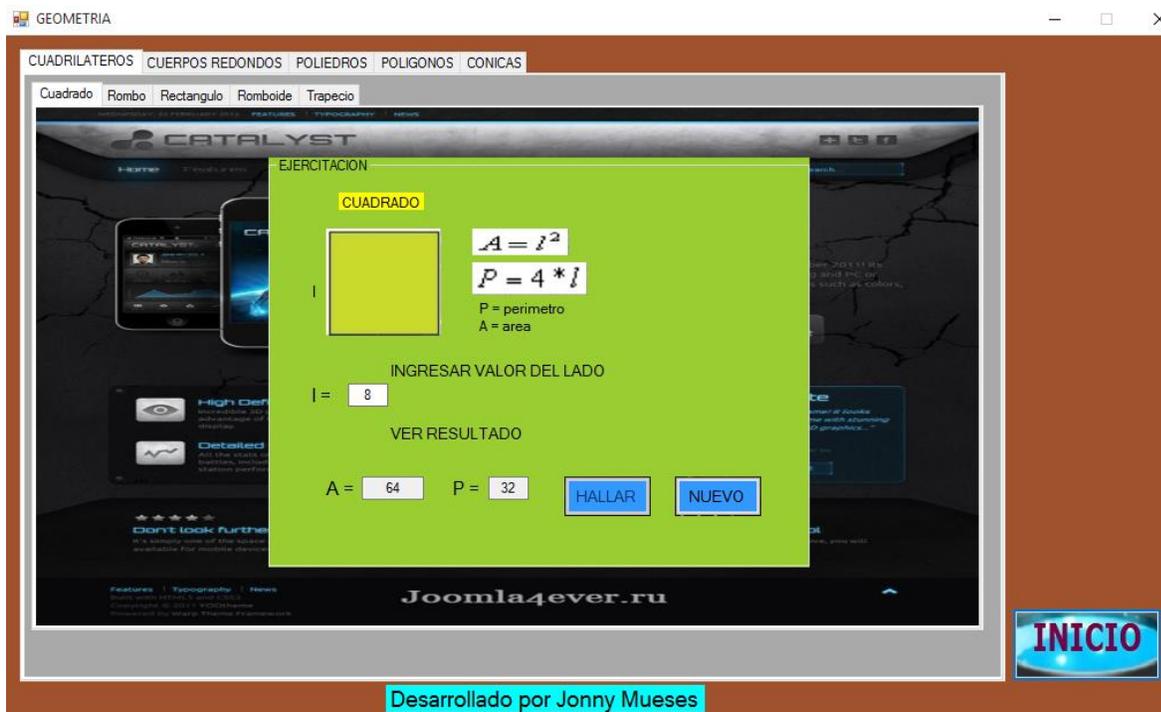


Figura 109: muestra del resultado del área y perímetro. Mueses, J. (2017). Fuente: autoría propia

EJERCITACION

CUADRADO



$$A = l^2$$

$$P = 4 * l$$

P = perimetro
A = area

INGRESAR VALOR DEL LADO

l =

VER RESULTADO

A = P =

HALLAR **NUEVO**

Figura 110: sistema borra los datos para ingresar nuevos valores. Mueses, J. (2017). Fuente: autoría propia

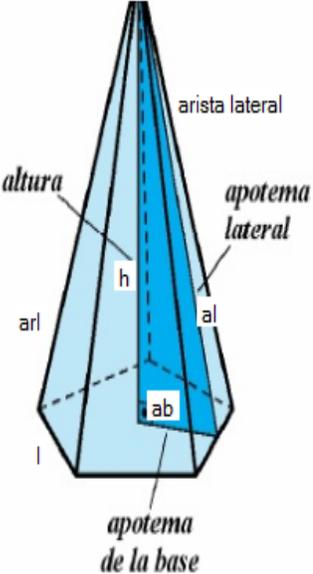
GEOMETRIA

CUADRILATEROS CUERPOS REDONDOS POLIEDROS POLIGONOS CONICAS

Cubo Paralelepipedo Piramide Cuadrada Piramide Regular Prisma regular Tetraedro Prisma triangular

PIRAMIDE REGULAR

Piramide pentagonal.
base de 5 lados



arista lateral

apotema lateral

altura

arl

h

al

ab

apotema de la base

V = volumen
A = area

$$A = \frac{p * (a_1 + a_2)}{2}$$

$$V = \frac{p * a_b * h}{3}$$

INGRESAR VALORES

5

h

l = h =

VER RESULTADO

A =

V =

HALLAR **NUEVO**

l = lado
al = apotema lateral
ab = apotema de la base
h = altura
arl = arista lateral

Figura 111: ingresando valores para hallar el área y volumen de una pirámide regular. Mueses, J. (2017). Fuente: autoría propia

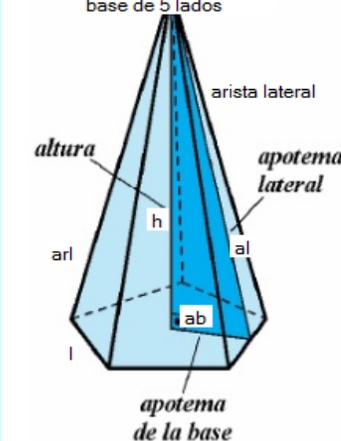
GEOMETRIA

CUADRILATEROS CUERPOS REDONDOS POLIEDROS POLIGONOS CONICAS

Cubo Paralelepipedo Piramide Cuadrada Piramide Regular Prisma regular Tetraedro Prisma triangular

PIRAMIDE REGULAR

Piramide pentagonal.
base de 5 lados



V = volumen
A = area

$$A = \frac{p * (a_1 + a_2)}{2}$$

$$V = \frac{p * a_2 * h}{3}$$

INGRESAR VALORES

5

h

l = h =

VER RESULTADO

debe ingresar valores en todos los campos

Aceptar

l = lado
al = apotema lateral
ab = apotema de la base
h = altura
ar = arista lateral

Figura 112: mensaje de error cuando no se ingresan valores en los campos indicados. Mueses, J. (2017). Fuente: autoría propia

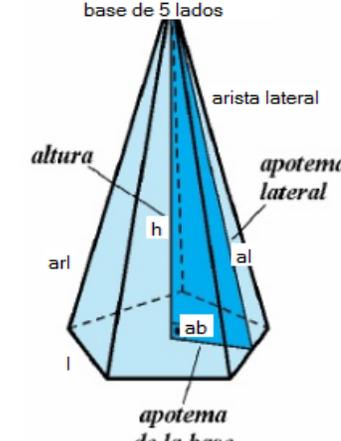
GEOMETRIA

CUADRILATEROS CUERPOS REDONDOS POLIEDROS POLIGONOS CONICAS

Cubo Paralelepipedo Piramide Cuadrada Piramide Regular Prisma regular Tetraedro Prisma triangular

PIRAMIDE REGULAR

Piramide pentagonal.
base de 5 lados



V = volumen
A = area

$$A = \frac{p * (a_1 + a_2)}{2}$$

$$V = \frac{p * a_2 * h}{3}$$

INGRESAR VALORES

5

h

l = 4 h = 8

VER RESULTADO

piramide pentagonal
base de 5 lados
A = 121.818995022191
V = 92.3760430703401

HALLAR NUEVO

l = lado
al = apotema lateral
ab = apotema de la base
h = altura
ar = arista lateral

Figura 113: ingreso de valores y muestra de resultados. Mueses, J. (2017). Fuente: autoría propia

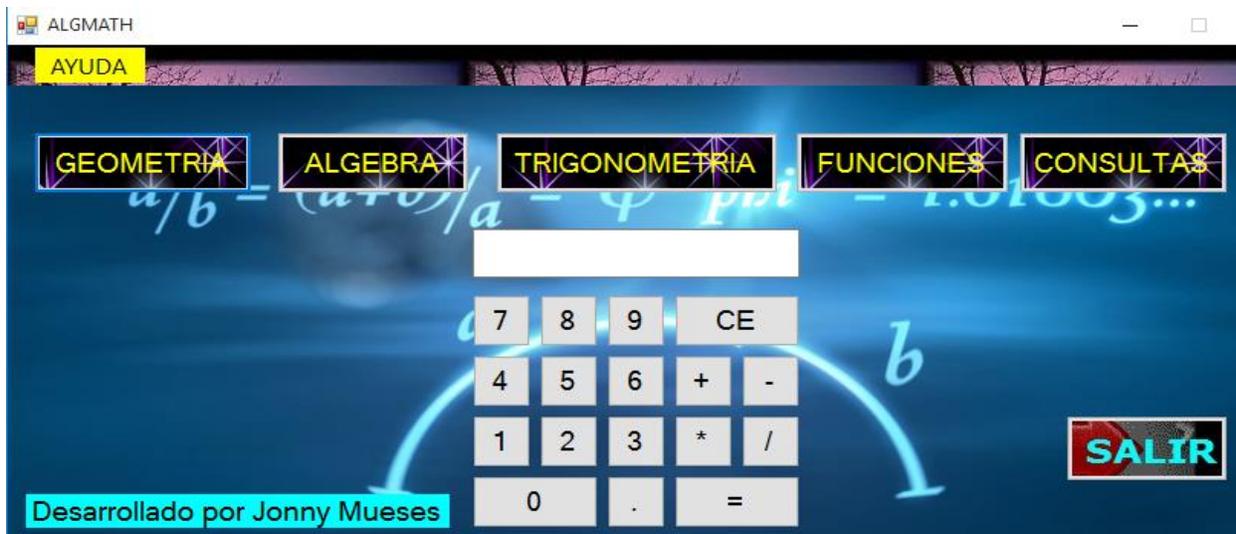


Figura 114: interfaz principal cuando se da en la opción inicio. Mueses, J. (2017). Fuente: autoría propia

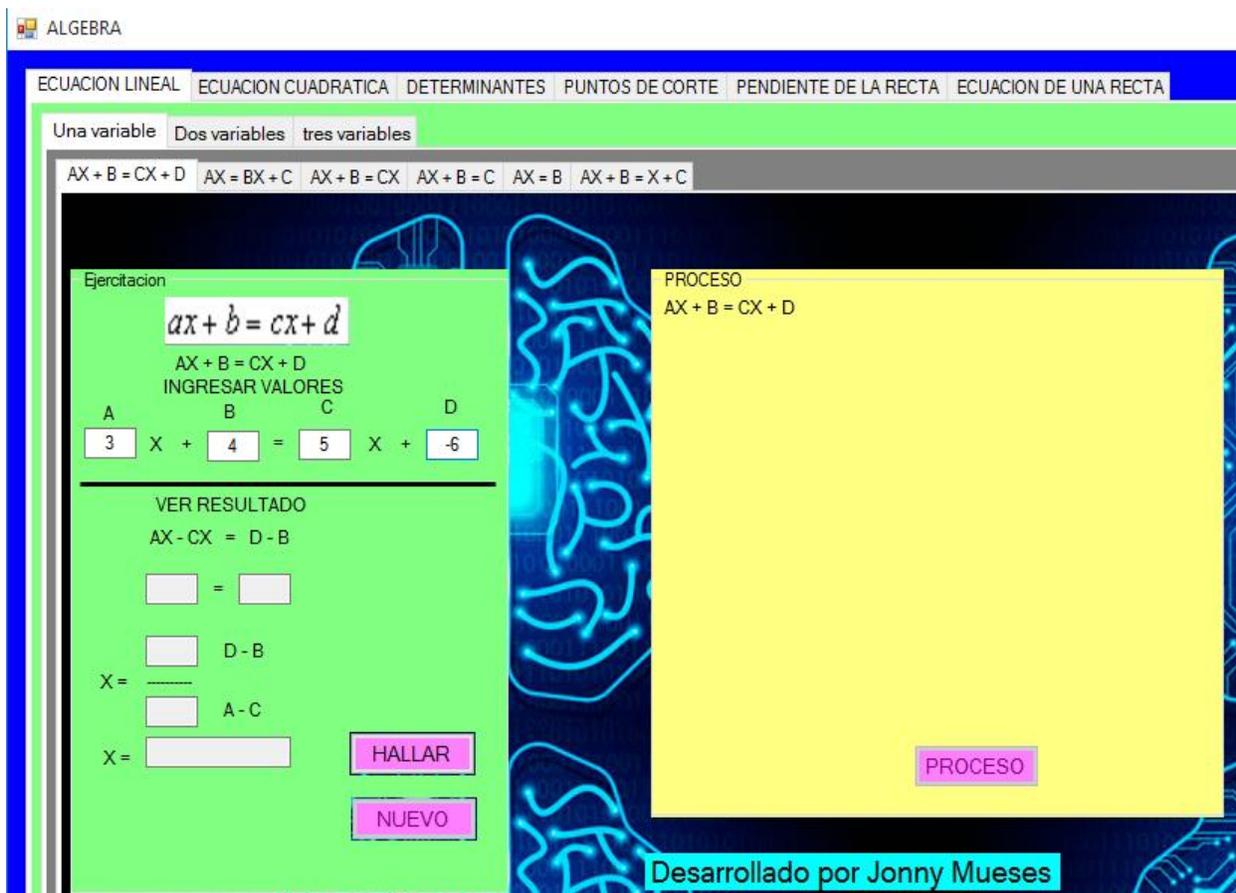


Figura 115: ingresando valores para hallar el valor de "x" en una ecuación lineal. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

Una variable Dos variables tres variables

AX + B = CX + D AX = BX + C AX + B = CX AX + B = C AX = B AX + B = X + C

Ejercitacion

$$ax + b = cx + d$$

AX + B = CX + D
INGRESAR VALORES

A	B	C	D
3	4	5	-6

VER RESULTADO

$$AX - CX = D - B$$

$$-2 = -10$$

X = $\frac{-10}{-2} = \frac{D - B}{A - C}$

X = 5

LA ECUACION TIENE SOLUCION

HALLAR

NUEVO

PROCESO

AX + B = CX + D

Desarrollado por Jonny Mueses

Figura 116: muestra de resultados del valor de "x". Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

Una variable Dos variables tres variables

AX + B = CX + D AX = BX + C AX + B = CX AX + B = C AX = B AX + B = X + C

Ejercitacion

$$ax + b = cx + d$$

AX + B = CX + D
INGRESAR VALORES

A	B	C	D
3	4	5	-6

VER RESULTADO

$$AX - CX = D - B$$

$$-2 = -10$$

X = $\frac{-10}{-2} = \frac{D - B}{A - C}$

X = 5

LA ECUACION TIENE SOLUCION

HALLAR

NUEVO

PROCESO

AX + B = CX + D

$$3x + 4 = 5x - 6$$

$$3x - 5x = -6 - 4$$

$$-2x = -10$$

$$x = -10 / -2$$

$$x = 5$$

Prueba

$$3x + 4 = 5x - 6$$

$$(3 * (-10 / -2)) + 4 = (5 * (-10 / -2)) - 6$$

$$(-30 / -2) + 4 = (-50 / -2) - 6$$

$$(-30 + (-2 * 4)) / -2 = (-50 - (-2 * 6)) / -2$$

$$-38 / -2 = -38 / -2$$

$$19 = 19$$

Desarrollado por Jonny Mueses

Figura 117: muestra del proceso y prueba de cómo se resolvió el problema. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

$ax^2 + bx + c = 0$ $ax^2 + bx = 0$ $ax^2 + c = 0$

EJERCITACION

$ax^2 + bx + c = 0$ $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

INGRESAR VALORES

A = B = C =

VER RESULTADO

$3x^2 - 7x + 2 = 0$

SOLUCIONES REALES

$x_1 = 2$

$x_2 = 0.3333333333333333$

HALLAR NUEVO

PASOS

$3x^2 - 7x + 2 = 0$

$x = \frac{-b \pm \sqrt{b^2 - (4 * a * c)}}{(2 * a)}$

$x^2 + bx + c = 0$

$x = \frac{-(-7) \pm \sqrt{(-7)^2 - (4 * 3 * 2)}}{(2 * 3)}$

$x = 7 \pm \sqrt{(49 - 24)} / 6$

$x = 7 \pm \sqrt{(25)} / 6$

SOLUCIONES REALES

$x_1 = 7 + \sqrt{(25)} / 6$

$x_1 = 2$

$x_2 = 7 - \sqrt{(25)} / 6$

$x_2 = 0.3333333333333333$

PROCESO

Figura 118: ingreso de valores y muestra de resultado y proceso para hallar el valor de “x” en ecuación cuadrática. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

2 x 2 con numeros 2 x 2 con variables 3 x 3 con numeros 3 x 3 con variables

DETERMINANTE 3 X 3
HALLAR LA DETERMINANTE USANDO LA REGLA DE SARRUS

INGRESE VALORES

<input type="text" value="2"/>	<input type="text" value="3"/>	<input type="text" value="4"/>
<input type="text" value="3"/>	<input type="text" value="5"/>	<input type="text" value="6"/>
<input type="text" value="2"/>	<input type="text" value="4"/>	<input type="text" value="3"/>

VER RESULTADO

$a = (m * r * v) + (q * u * p) + (t * n * s)$

$b = (p * r * t) + (s * u * m) + (v * n * q)$

$\det = a - b$

m	n	p
q	r	s
t	u	v

m	n	p
q	r	s
t	u	v

HALLAR NUEVO

Figura 119: ingreso de valores para hallar la determinante de una matriz 3x3. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

2 x 2 con numeros 2 x 2 con variables 3 x 3 con numeros 3 x 3 con variables

DETERMINANTE 3 X 3
HALLAR LA DETERMINANTE USANDO LA REGLA DE SARRUS

INGRESE VALORES

2	3	4
3	5	6
2	4	3

VER RESULTADO
usamos la regla de Sarrus agregamos las 2 primeras filas despues de la tercera fila

2	3	4		
3	5	6		
2	4	3	2	3
2	3	4		
3	5	6		

multiplicamos en diagonal de izquierda a derecha de arriba hacia abajo

$$a = (2 * 5 * 3) + (3 * 4 * 4) + (2 * 3 * 6)$$

$$a = 30 + 48 + 36$$

$$a = 114$$

multiplicamos en diagonal de derecha a izquierda de arriba hacia abajo

$$b = (4 * (5) * (2)) + (6 * (4) * (2)) + (3 * (3) * (3))$$

$$b = 40 + 48 + 27$$

$$b = 115$$

hallamos la determinante

$$DET = a - b$$

$$DET = 114 - 115$$

$$DET = -1$$

HALLAR NUEVO

Figura 120: muestra de resultado del valor de la determinante. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

EJERCITACION

$y = mx + b$ $y = mx + b$ $P = (x, y)$

$y - (m * x) = b$

INGRESAR VALORES

m =

P = (,)

VER RESULTADO

b =

Ecuacion de la recta

HALLAR NUEVO

PROCESO

PROCESO

Figura 121: ingreso de valores para hallar la ecuación de una recta. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

EJERCITACION

$y = mx + b$ $y = mx + b$ $P = (x, y)$
 $y - (m \cdot x) = b$

INGRESAR VALORES

m =

P = (,)

VER RESULTADO

$6 = 4 \cdot (2) + b$

b =

Ecuacion de la recta

$y = 4x - 2$

HALLAR NUEVO

PROCESO

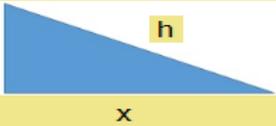
$P = (x, y)$
 $P = (2, 6)$
 $m = 4$
 $y = mx + b$
 Reemplazamos valores
 $6 = 4 \cdot (2) + b$
 $6 = 8 + b$
 despejamos b
 $6 - 8 = b$
 $-2 = b$
 $y = 4x - 2$

PROCESO

Figura 122: muestra del resultado y proceso de la ecuación de la recta obtenida. Mueses, J. (2017). Fuente: autoría propia

TRIGONOMETRIA

Teorema de pitagoras Funciones trigonometricas solucion tipo 1 solucion tipo 2



INGRESAR VALORES

x = y = h =

VER RESULTADO

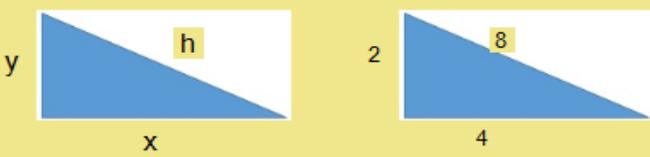
HALLAR NUEVO

Desarrollado por Jonny Mueses

Figura 123: ingreso de valores para hallar el valor de las funciones trigonométricas. Mueses, J. (2017). Fuente: autoría propia

TRIGONOMETRIA

Teorema de pitagoras Funciones trigonometricas solucion tipo 1 solucion tipo 2



INGRESAR VALORES
 $x = 4$ $y = 2$ $h = 8$

VER RESULTADO

sen = $\frac{y}{h} = \frac{2}{8} = 0,25$

cos = $\frac{x}{h} = \frac{4}{8} = 0,5$

tan = $\frac{y}{x} = \frac{2}{4} = 0,5$

cot = $\frac{x}{y} = \frac{4}{2} = 2$

sec = $\frac{h}{x} = \frac{8}{4} = 2$

csc = $\frac{h}{y} = \frac{8}{2} = 4$

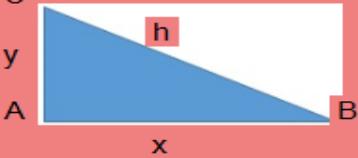
HALLAR NUEVO

Desarrollado por Jonny Mueses

Figura 124: muestra del resultado del valor de las funciones trigonométricas. Mueses, J. (2017). Fuente: autoría propia

TRIGONOMETRIA

Teorema de pitagoras Funciones trigonometricas solucion tipo 1 solucion tipo 2



INGRESAR VALORES
 $x = 4$ $y = 3$

VER RESULTADO HALLAR NUEVO

Seleccione las variables que conoce debe seleccionar solo 2
 x y h

MOSTRAR

Desarrollado por Jonny Mueses

Figura 125: ingreso de valores para hallar las funciones trigonométricas conociendo dos variables. Mueses, J. (2017). Fuente: autoría propia

TRIGONOMETRIA

Teorema de pitagoras Funciones trigonometricas solucion tipo 1 solucion tipo 2

INGRESAR VALORES

x = y =

VER RESULTADO HALLAR NUEVO

$h^2 = x^2 + y^2$
 $h^2 = 4^2 + 3^2$
 $h = 5$

$\text{sen } B = y/h = 3/5 = 0,6$
 $\text{cos } B = x/h = 4/5 = 0,8$
 $\text{tan } B = y/x = 3/4 = 0,75$
 $\text{cot } B = x/y = 4/3 = 1,3333333333333333$
 $\text{sec } B = h/x = 5/4 = 1,25$
 $\text{csc } B = h/y = 5/3 = 1,6666666666666667$

$\text{sen } C = x/h = 4/5 = 0,8$
 $\text{cos } C = y/h = 3/5 = 0,6$
 $\text{tan } C = x/y = 4/3 = 1,3333333333333333$
 $\text{cot } C = y/x = 3/4 = 0,75$
 $\text{sec } C = h/y = 5/3 = 1,6666666666666667$
 $\text{csc } C = h/x = 5/4 = 1,25$

Seleccione las variables que conoce debe seleccionar solo 2
 x y h

MOSTRAR

Desarrollado por Jonny Mueses

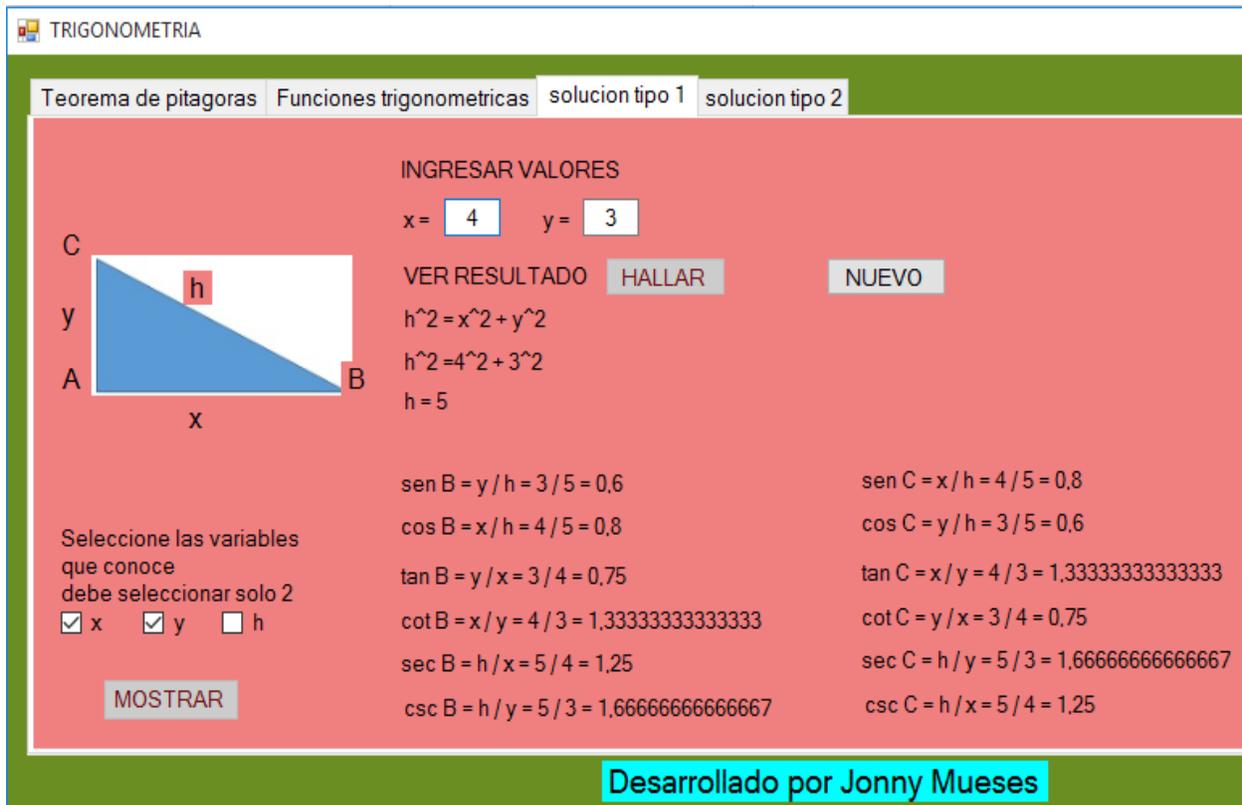


Figura 126: muestra del resultado de los valores obtenidos. Mueses, J. (2017). Fuente: autoría propia

FUNCIONES

LINEAL CUADRATICA

$y = ax + b$ $y = x + b$ $y = ax$ $y = x$

EJERCITACION FUNCION LINEAL

$y = ax + b$

INGRESE VALORES

a = b =

CONTINUAR

Ingrese valor de x

x = HALLAR

VER VALOR DE Y

y = ALMACENAR

VER

	x	y
*		

PASOS

Oprima este boton NUEVO cuando quiere dar nuevos valores para "a" y "b". Es decir, para hacer una nueva ecuacion.

NUEVO

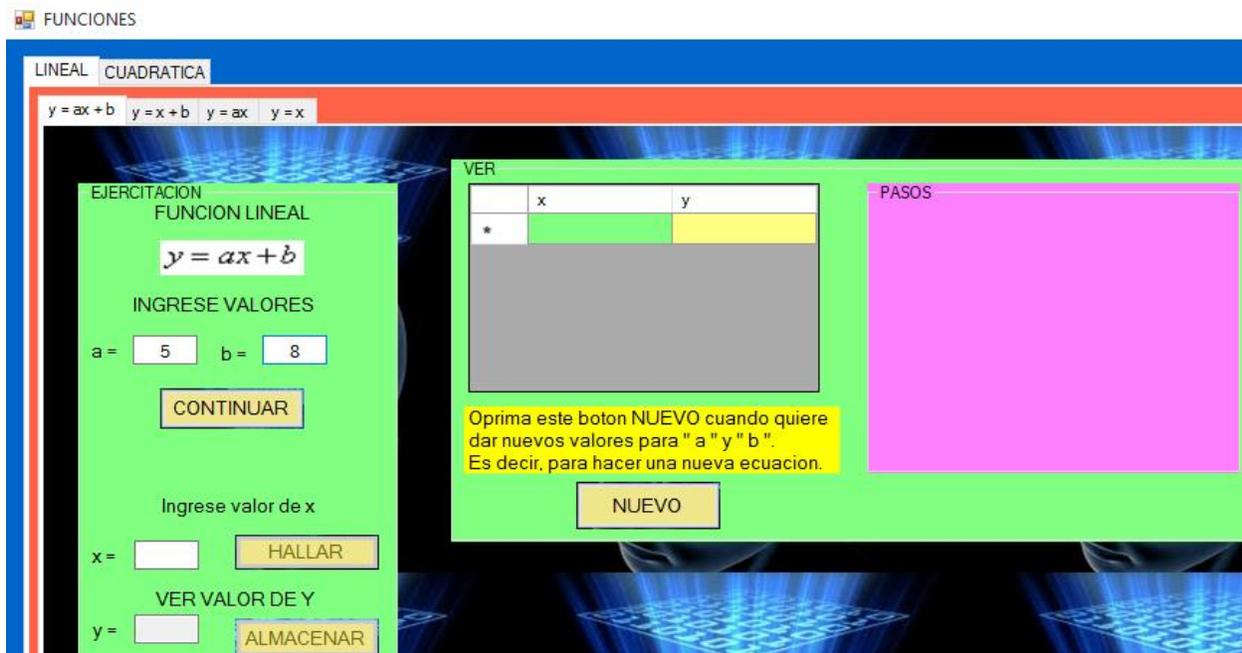


Figura 127: ingreso de valores para hallar el valor de "y" en una función lineal. Mueses, J. (2017). Fuente: autoría propia

FUNCIONES

LINEAL CUADRATICA

$y = ax + b$ $y = x + b$ $y = ax$ $y = x$

EJERCITACION
FUNCION LINEAL

$y = ax + b$

INGRESE VALORES

a = 5 b = 8

CONTINUAR

$y = 5x + 8$

Ingrese valor de x

x = 3 HALLAR

VER VALOR DE Y

y = ALMACENAR

VER

x	y
2	18
3	23
*	

Oprensa este boton NUEVO cuando quiere dar nuevos valores para "a" y "b". Es decir, para hacer una nueva ecuacion.

NUEVO

PASOS

$y = 5x + 8$
damos valores a x
cuando $x = 3$
 $y = (5 * (3)) + 8$
 $y = 15 + 8$
 $y = 23$

Figura 128: muestra del resultado del valor de "y" y los pasos llevados a cabo. Mueses, J. (2017). Fuente: autoría propia

CONSULTAS

EJEMPLOS ACTIVIDADES INFORMACION

Ecuacion lineal Ecuacion cuadratica Determinantes Puntos de corte Pendiente y ecuacion de la recta Trigonometria Geometria

1). de acuerdo al teorema de Pitagoras, si "h = 10" y "x = 6", el valor de "y" es:

a) $y = 8$ $h^2 = x^2 + y^2$

b) $y = 4$

c) $y = 64$

2). de acuerdo al teorema de Pitagoras, si "x = 4" y "y = 3", el valor de "h" es:

a) $y = 25$ $h^2 = x^2 + y^2$

b) $y = 5$

c) $y = 1$

3). de acuerdo al grafico, el sec B es igual a:

a) $\sec B = h / y$

b) $\sec B = x / h$

c) $\sec B = h / x$

4). de acuerdo al grafico, el sen a es igual a:

a) $\sen a = 1 / 2$

b) $\sen a = 3 / 4$

c) $\sen a = 2$

5). de acuerdo al grafico, el cos a es igual a:

a) $\cos a = 2$

b) $\cos a = 1 / 2$

c) $\cos a = 1 / 4$

6). de acuerdo al grafico, "y" es igual a:

a) $y = 2$

b) $y = 8$

c) $y = 3.464$

7). de acuerdo al grafico, "x" es igual a:

a) $x = 11.547$

b) $x = 20$

c) $x = 5$

8). de acuerdo al grafico, cos B es igual a:

a) $\cos B = 18 / 5$

b) $\cos B = 3 / 5$

c) $\cos B = 32 / 5$

9). de acuerdo al grafico, tan B es igual a:

a) $\tan B = 34 / 3$

b) $\tan B = 4 / 3$

c) $\tan B = 16 / 3$

5 PUNTOS POR CADA PREGUNTA ACERTADA

VER RESULTADO

NUEVO INTENTO

Figura 129: resolviendo un cuestionario de selección única. Mueses, J. (2017). Fuente: autoría propia

CONSULTAS

EJEMPLOS ACTIVIDADES INFORMACION

Ecuacion lineal Ecuacion cuadratica Determinantes Puntos de corte Pendiente y ecuacion de la recta Trigonometria Geometria

1). de acuerdo al teorema de Pitagoras, si "h = 10" y "x = 6", el valor de "y" es:

$h^2 = x^2 + y^2$

a) y = 8
 b) y = 4
 c) y = 64

2). de acuerdo al teorema de Pitagoras, si "x = 4" y "y = 3", el valor de "h" es:

$h^2 = x^2 + y^2$

a) y = 25
 b) y = 5
 c) y = 1

3). de acuerdo al grafico, el sec B es igual a:

a) $\sec B = h / y$
 b) $\sec B = x / h$
 c) $\sec B = h / x$

4). de acuerdo al grafico, el sen a es igual a:

a) $\sen a = 1 / 2$
 b) $\sen a = 3 / 4$
 c) $\sen a = 2$

5). de acuerdo al grafico, el cos a es igual a:

a) $\cos a = 2$
 b) $\cos a = 1 / 2$
 c) $\cos a = 1 / 4$

6). de acuerdo al grafico, "y" es igual a:

a) y = 2
 b) y = 8
 c) y = 3.464

7). de acuerdo al grafico, "x" es igual a:

a) x = 11.547
 b) x = 20
 c) x = 5

8). de acuerdo al grafico, cos B es igual a:

a) $\cos B = 18 / 5$
 b) $\cos B = 3 / 5$
 c) $\cos B = 32 / 5$

9). de acuerdo al grafico, tan B es igual a:

a) $\tan B = 34 / 3$
 b) $\tan B = 4 / 3$
 c) $\tan B = 16 / 3$

5 PUNTOS POR CADA PREGUNTA ACERTADA

PUNTAJE
 su puntaje es: 40 de 45

RESPUESTAS ACERTADAS
 su acertacion es: 8 de 9

VER RESULTADO

NUEVO INTENTO

Figura 130: muestra del puntaje obtenido y respuestas acertadas. Mueses, J. (2017). Fuente: autoría propia

CONSULTAS

EJEMPLOS ACTIVIDADES INFORMACION

ALGEBRA

TEXTO 1 TEXTO 2

TEXTO 3 TEXTO 4

TEXTO 5 TEXTO 6

TEXTO 7 TEXTO 8

GEOMETRIA

TEXTO 9 TEXTO 10

TEXTO 11 TEXTO 12

TEXTO 13 TEXTO 14

Figura 131: ingreso a la interfaz información. Mueses, J. (2017). Fuente: autoría propia

CONSULTAS

EJEMPLOS ACTIVIDADES INFORMACION

ALGEBRA

TEXTO 1 TEXTO 2

TEXTO 3 TEXTO 4

TEXTO 5 TEXTO 6

TEXTO 7 TEXTO 8

GEOMETRIA

TEXTO 9 TEXTO 10

TEXTO 11 TEXTO 12

TEXTO 13 TEXTO 14

PROBABILIDAD

TEXTO 15 TEXTO 16

TRIGONOMETRIA

TEXTO 17 TEXTO 18

TEXTO 19

3

Ecuaciones de segundo grado

Objetivos

En esta quincena aprenderás a:

- Identificar las soluciones de una ecuación.
- Reconocer y obtener ecuaciones equivalentes.
- Resolver ecuaciones de primer grado
- Resolver ecuaciones de segundo grado tanto completas como incompletas.
- Utilizar el lenguaje algebraico y las ecuaciones para resolver problemas.

Antes de empezar.

1. Expresiones Algebraicas pág. 42
Identidad y ecuación
Solución de una ecuación

2. Ecuaciones de primer grado pág. 44
Definición
Método de resolución
Resolución de problemas

3. Ecuaciones de segundo grado pág. 46
Definición. Tipos
Resolución de $ax^2+bx=0$
Resolución de $ax^2+c=0$
Resolución de $ax^2+bx+c=0$
Suma y producto de las raíces
Discriminante de una ecuación
Ecuación $(x-a)\cdot(x-b)=0$
Resolución de problemas

Ejercicios para practicar

Para saber más

Resumen

Actividades para enviar al tutor

1 / 16

Figura 132: selección del texto 1 y muestra del texto en formato pdf. Mueses, J. (2017). Fuente: autoría propia

Acerca de ALGMATH

ALGMATH

Versión 1.0.0.0

Copyright © 2017

Autor: JonnyMueses

El software Algmath es un programa de matemáticas de código abierto y de uso libre, usado para realizar cálculos y para consultas básicas

Aceptar

Figura 133: ingreso a la interfaz a cerca de. Mueses, J. (2017). Fuente: autoría propia

Pruebas finales:

En este paso compilamos el programa y lo convertimos en archivo ejecutable o instalador (**ver anexo**). Ya realizado las pruebas iniciales y constar que el programa funciona bien, se procede a compilar el programa y hacerlo un programa ejecutable, realizando los procesos adecuados y que se pueda instalar en computadoras con sistema operativo más recientes de Windows.

Realizada la compilación y generación del ejecutable, se procedió con la instalación de la aplicación creada, donde se hizo de manera eficiente y su funcionamiento fue correcto. Antes de iniciar con la siguiente etapa, se hicieron los ajustes necesarios de manera minuciosas analizando los detalles más pequeños.

11.10 Etapa de utilización.

En esta fase o etapa, se lleva cabo lo que es el uso del programa, por parte del docente y de los estudiantes del grado noveno, en el área de matemáticas. El programa como tal se instalará en los computadores que están el área de informática y en aquellos que tengan los estudiantes en sus casas.

Previo a esto se tendrán en cuenta ciertos requisitos importantes antes de realizar la instalación para que no se presenten problemas.

Requisitos:**Software:**

Sistemas Operativos: windows 7 / Windows 8 y 8.1 / Windows 10.

Net frameworks: v. 4.0 (para tener un mejor funcionamiento del programa, si el computador no lo tiene, se lo puede descargar e instalar antes de la instalación del software, o conjuntamente con la instalación)

Hardware mínimo

600 MHz CPU

500 MB RAM

40 MB de espacio en disco para instalación.

Ficha técnica.

Nombre: ALGMATH

Versión: 1.0.0.0

Tamaño: 26 MB

Idioma: español

El programa se instaló en los computadores de la sala de informática (**ver anexo**), previo a esto se hizo una revisión detallada a nivel hardware y software para ver si cumplían con los requisitos mínimos para que el programa funcione a la perfección, dando visto bueno a la revisión, se hizo la instalación de net frameworks 4. 0 y de ALGMATH. Se hizo una explicación al docente y a los estudiantes sobre la instalación del software y el funcionamiento del mismo.

Los estudiantes practicaron y realizaron la instalación del software ellos mismos con la supervisión y orientación de mi persona, de igual forma lo hizo el docente. También realizaron la exploración del programa en general, observando cada interfaz, su funcionamiento, y se

familiarizaron con el programa, realizaron cálculos sobre diferentes temas y se ejercitaron realizando las actividades del cuestionario para evaluarse en su aprendizaje.

Para que el programa tenga una aceptación y manejo eficaz, se emplearon 10 hora distribuidos en 2 horas por semana, para que los usuarios tengan un manejo correcto del programa.

Gran parte de los estudiantes y de igual manera el docente, llevaron el software para instalarlo en sus computadores de sus casas. la guía sobre el manejo del software les ayudó mucho y así pudieron usar con potencialidad el programa y aprender sobre los diferentes temas.

11.11 Etapa de mantenimiento.

En esta etapa se manejarán lo que son posibles cambios, a lo largo de la vida útil y a futuro, en el funcionamiento del software, donde se tienen factores como: posibles cambios en algunos de los componentes del sistema, modificaciones funcionales no contempladas durante el diseño y desarrollo del software. Implementación de más funciones que abarquen nuevos temas, mejoramiento en general y de cada interfaz del programa para hacer de este más potente.

Lo anterior con el fin de que sea usado por un periodo largo.

12 Conclusiones.

El desarrollo del presente proyecto y, por consiguiente, del presente software, permitió aplicar los conocimientos adquiridos en la formación del programa Tecnología de Sistemas, realizar investigaciones y consultas sobre lo que se está planteando hacer, y sobre todo aportar algo a la sociedad, en este caso brindar una ayuda a estudiantes del grado noveno de la I.E.R.J.A.S., en el área de matemáticas a través de la implementación de un software, usado como complemento en el aprendizaje, además de ser parte de la continuidad del desarrollo del software educativo libre.

De igual forma se tuvo conciencia de la importancia que tiene el uso del software educativo en el sistema de la educación, y que falta mucho por hacer en este tema, ya que muchas instituciones no cuentan con herramientas informáticas educativas ni aun gratuitas o de forma libre.

Así como vemos un avance sorprendente de la tecnología y de las TIC's, vemos un proceso lento en el uso del software educativo ya sea online u offline, sobre todo en áreas rurales o apartadas, y que incluso en las mismas ciudades se presenta el caso, y cuyas causas pueden ser varias.

Por tal motivo se ve la importancia de que se implemente con más fuerza el uso del software educativo en áreas de mayor interés como las matemáticas, programas flexibles e interactivos que complementen el aprendizaje del estudiante, y que con ello no se está aislando al docente o prescindiendo de sus servicios, sino que en conjunto se afianza más el sistema de enseñanza, se hace del estudiante una personas más activa, más responsable en sus deberes académicos, más eficiente y con mayores conocimientos al momento de seguir con la formación profesional.

El desarrollo de este software ALGMATH es una muestra de que se puede implementar este tipo de herramienta para el aprendizaje y despertar nuevas formas de aprender, que produjo grandes efectos positivos tanto para mi persona como para los usuarios finales, se logró la aceptación del programa, el funcionamiento y muestra de resultados del aplicativo fue de gran ayuda para los estudiantes. Esto fue una experiencia significativa donde se hizo un aporte al sistema educativo, y sobre todo dar a conocer los beneficios que ofrece el desarrollo del software libre educativo.

13 Referencias

Somerville, I. (2011). Ingeniería de software. México: Editorial Pearson Educación S.A.

Aparicio, A., Pilar, A. M. (2012). Módulo de Ingeniería de Software. UNAD.

Campderrich Falgueras, B. (2003). Ingeniería del software. Barcelona: Editorial UOC.

Moreno N., J. J. arquitecturas de software. Recuperado de:

<http://pegasus.javeriana.edu.co/~mad/Arquitecturas%20de%20SW.pdf>

Murillo, J.L. (2009). Software educativo libre o gratuito para utilizar en el aula. Recuperado de:

<http://www.edulibre.info/software-educativo-libre-o>

Noragueda, C. (2015). Programas de software libre para labores educativas. Recuperado de:

<https://hipertextual.com/2015/06/interesantes-programas-de-software-libre-educativo>

Pimiento E, L.E. (2009). La Evolución del Software. Recuperado de:

<http://www.monografias.com/trabajos73/evolucion-software/evolucion-software.shtml>

Omatos S., A. (). Software libre y Educación. Recuperado de:

http://legalidad.aomatos.com/software_libre_y_educacin.html

Díaz, J. (). Software libre y Matemáticas. Recuperado de:

<http://www.educacontic.es/blog/software-libre-y-matematicas>

GALVIS, A., (2000) Ingeniería de software educativo. Universidad de Los Andes. Ediciones

UNIANDÉS. Colombia.

SÁNCHEZ, Jaime. (1993). Informática Educativa. Santiago de Chile: Editorial. Universitaria.

Cataldi, Z. (2000). Metodología de diseño, desarrollo y evaluación de software educativo.

Recuperado de: <http://www.iidia.com.ar/rgm/tesistas/cataldi-tesisdemagistereninformatica.pdf>

Quintero, H., Portillo, L., Luque, R. Gonzales, M. (2005). Desarrollo de software educativo: una propuesta metodológica. Recuperado de:

<http://publicaciones.urbe.edu/index.php/telos/article/viewFile/3609/4501>

García P, Carlos I. (). Software de Libre Disponibilidad. Su utilización en el Estado. Recuperado de: <http://www.monografias.com/trabajos18/software-libre-estado/software-libre-estado.shtml>

Cobo, J. C. (2009). Conocimiento, creatividad y software libre: una oportunidad para la educación en la sociedad actual. Recuperado de:

<http://www.uoc.edu/uocpapers/8/dt/esp/cobo.pdf>

Caro P., M. F., Toscazo M., R. E., Hernandez R., F. M., David L., M. E. (2009). DISEÑO DE SOFTWARE EDUCATIVO BASADO EN COMPETENCIAS. Recuperado de:

<http://www.scielo.org.co/pdf/cein/v19n1/v19n1a05.pdf>

Castro D., R., (2008). El software educativo en el entorno de los medios de enseñanza.

Recuperado de: <http://www.redalyc.org/pdf/1815/181518069007.pdf>

Baldor P., A. A. (1941). ALGEBRA. México. Editorial: Publicaciones Cultural, Códice América, S.A.

Culebro J., M., Gómez H., W. G., Torres S., S. (2006). Software libre vs software propietario.

Ventajas y desventajas. Recuperado de: <http://www.rebellion.org/docs/32693.pdf>

Martin G., F. (2015). La laboriosa historia del software libre. Recuperado de:

<https://hipertextual.com/presentado-por/bbva/historia-del-software-libre>

Martínez, M. L. (2011). VENTAJAS DE EMPLEAR SOFTWARE EDUCATIVO. Recuperado de: <http://www.wangelmartz.blogspot.com.co/p/ventajas-de-emplear-software-educativo.html>

Betsabe A., D. (2011). VENTAJAS DE EMPLEAR SOFTWARE EDUCATIVO. Recuperado de: <http://betsabipul.blogspot.com.co/2011/05/ventajas-de-emplear-software-educativo.html>

Montes, D., Duque, V. (2009). CLASIFICACION DEL SOFTWARE EDUCATIVO. Recuperado de: <http://ucosweducativo.blogspot.com.co/2009/06/clasificacion-del-software-educativo.html>

Gonzales P., D. (2004). Software libre en los institutos. Recuperado de: https://www.cs.upc.edu/~tonis/daniel_gonzalez_pinyero.pdf

Stallman, R. M. (2004). Software libre para una sociedad libre. Recuperado de: https://www.gnu.org/philosophy/fsfs/free_software2.es.pdf

14 Anexos.

14.1 Desarrollo de una encuesta tipo cuestionario para análisis de resultados.

1. ¿Cómo resuelves los ejercicios de matemáticas?

Con mucha facilidad.

Con poca facilidad.

De forma regular.

2. ¿te gusta el área de matemáticas?

Bastante.

Poco.

Nada.

3. ¿Cómo adquiere la mayoría de aprendizaje de matemáticas?

Con el docente.

Con los compañeros.

Con alguien que sepa de matemáticas.

4. ¿Cómo la va en los exámenes?

Excelente.

Bien.

Regular.

5. ¿ha oído hablar sobre el software educativo?

Si.

No.

6. ¿Sabe que es un software educativo?

Si.

Algo de idea.

No.

7. ¿ha usado algún software de matemáticas?

Si.

No.

8. ¿crees que el uso de algún software de matemáticas en este nivel, podría complementar tu aprendizaje?

Si.

Un poco.

No.

9. ¿si te facilitan un software de matemáticas lo usarías continuamente?

Si.

No.

10. ¿conoces sobre algunos nombres de software educativos de matemáticas?

Si.

No.

14.2 Código del programa.

Código de la interfaz cargar programa.

```
Public Class CARGAR

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        ProgressBar1.Increment(1)
        If ProgressBar1.Value = ProgressBar1.Maximum Then
            Me.Visible = False
            Form1.Show()
            Timer1.Stop()
            ProgressBar1.Value = 0

        End If
    End Sub

    Private Sub CARGAR_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Timer1.Start()
    End Sub
End Class
```

Código de la interfaz principal.

```
Public Class Form1
    Dim OP As String
    Dim a, b, resultado As Double
    Dim c As Double
    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "5"
        c = False
    End Sub

    Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
        a = TextBox1.Text
        TextBox1.Text = ""
        OP = "DIVISION"
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If c = True Then
            TextBox1.Text = ""
        End If
    End Sub
```

```
        TextBox1.Text = TextBox1.Text + "7"
        c = False
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button9.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "8"
        c = False
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button8.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "9"
        c = False
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button7.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "4"
        c = False
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "6"
        c = False
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "1"
        c = False
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        If c = True Then
            TextBox1.Text = ""
        End If
        TextBox1.Text = TextBox1.Text + "2"
        c = False
    End Sub
```

```

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
    If c = True Then
        TextBox1.Text = ""
    End If
    TextBox1.Text = TextBox1.Text + "3"
    c = False
End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    If c = True Then
        TextBox1.Text = ""
    End If
    TextBox1.Text = TextBox1.Text + "0"
    c = False
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    If c = True Then
        TextBox1.Text = ""
    End If
    TextBox1.Text = TextBox1.Text + "."
    c = False
End Sub

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
    a = TextBox1.Text
    TextBox1.Text = ""
    OP = "SUMA"
End Sub

Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
    a = TextBox1.Text
    TextBox1.Text = ""
    OP = "RESTA"
End Sub

Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
    a = TextBox1.Text
    TextBox1.Text = ""
    OP = "MULTIPLICACION"
End Sub

Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
    b = Val(TextBox1.Text)
    Select Case OP
        Case "SUMA"
            resultado = a + b
            TextBox1.Text = resultado
        Case "RESTA"
            resultado = a - b
            TextBox1.Text = resultado
    End Select
End Sub

```

```

        Case "DIVISION"
            resultado = a / b
            TextBox1.Text = resultado
        Case "MULTIPLICACION"
            resultado = a * b
            TextBox1.Text = resultado
    End Select
End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
    TextBox1.Text = ""
End Sub

Private Sub TextBox1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
    If e.KeyCode = Keys.Divide Then
        Button17_Click(Nothing, Nothing)
    ElseIf e.KeyCode = Keys.Subtract Then
        Button17_Click(Nothing, Nothing)
    ElseIf e.KeyCode = Keys.Add Then
        Button13_Click(Nothing, Nothing)
    ElseIf e.KeyCode = Keys.Multiply Then
        Button16_Click(Nothing, Nothing)
    ElseIf e.KeyCode = Keys.Decimal Then
        Button3_Click(Nothing, Nothing)
    ElseIf e.KeyCode = Keys.Enter Then
        Button15_Click(Nothing, Nothing)
    ElseIf e.KeyCode = Keys.Back Then
        Button12_Click(Nothing, Nothing)
        e.SuppressKeyPress = True
    End If
End Sub

Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    If Char.IsDigit(e.KeyChar) Then
        e.Handled = False
    ElseIf Char.IsControl(e.KeyChar) Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged

End Sub

Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button19.Click
    GEOMETRIA.Show()
    Me.Hide()
End Sub

```

```

    Private Sub Button21_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button21.Click
        TRIGONOMETRIA.Show()
        Me.Hide()
    End Sub

    Private Sub Button20_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button20.Click
        ALGEBRA.Show()
        Me.Hide()
    End Sub

    Private Sub Button22_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button22.Click
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub ACercaDeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ACercaDeToolStripMenuItem.Click
        AboutBox1.Show()
        Me.Hide()
    End Sub

    Private Sub Button23_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button23.Click
        CONSULTAS.Show()
        Me.Hide()
    End Sub

    Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
        End
    End Sub
End Class

```

Código de la interfaz geometría (solo parte del código por motivo de extensión).

```

Public Class GEOMETRIA
    'pestaña cuadrilateros'
    'subpestaña cuadrado'
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        'boton hallar'
        If TextBox3.Text = "" Then
            MessageBox.Show("debe ingresar un valor")
        Else
            If TextBox3.Text <= 0 Then
                MessageBox.Show("debe ingresar un valor mayor que 0")
            Else
                'área del cuadrado'
                TextBox1.Text = TextBox3.Text * TextBox3.Text
                'perímetro del cuadrado'
                TextBox2.Text = 4 * TextBox3.Text
            End If
        End If
    End Sub
End Class

```

```

        Button1.Enabled = False 'se inhabilita el boton hallar'
        Button2.Enabled = True 'se habilita el boton nuevo'

    End If
End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    'boton nuevo; borra los datos ingresados'
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    Button1.Enabled = True 'se habilita el boton hallar'
    Button2.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'subpestaña rombo'
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    'boton hallar'
    If TextBox6.Text = "" Or TextBox7.Text = "" Or TextBox8.Text = "" Then
        MessageBox.Show("debe ingresar valores en todos los campos")
    Else
        If TextBox6.Text <= 0 Or TextBox7.Text <= 0 Or TextBox8.Text <= 0 Then
            MessageBox.Show("debe ingresar valores mayores que 0")
        Else
            If TextBox7.Text < TextBox8.Text Then
                MessageBox.Show("el valor de D debe ser mayor que el valor de d")
            Else
                'area del rombo'
                TextBox5.Text = (TextBox7.Text * TextBox8.Text) / 2
                'perímetro del rombo'
                TextBox4.Text = 4 * TextBox6.Text
                Button4.Enabled = False 'se inhabilita el boton hallar'
                Button3.Enabled = True 'se habilita el boton nuevo'

            End If
        End If
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    'boton nuevo: borra los datos'
    TextBox5.Text = ""
    TextBox4.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""

    Button4.Enabled = True 'se habilita el boton hallar'
    Button3.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'subpestaña rectangulo'
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
    'boton hallar'
    If TextBox11.Text = "" Or TextBox12.Text = "" Then

```

```

        MessageBox.Show("debe ingresar valores en todos los campos")
Else
    If TextBox11.Text <= 0 Or TextBox12.Text <= 0 Then
        MessageBox.Show("debe ingresar valores mayores que 0")
    Else
        'area del rectangulo'
        TextBox10.Text = TextBox11.Text * TextBox12.Text
        'perímetro del rectangulo'
        TextBox9.Text = (2 * TextBox11.Text) + (2 * TextBox12.Text)

        Button5.Enabled = True 'se habilita el boton hallar'
        Button6.Enabled = False 'se inhabilita el boton nuevo'
    End If
End If
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    'boton nuevo: borra los datos'
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""
    TextBox12.Text = ""

    Button6.Enabled = True 'se habilita el boton hallar'
    Button5.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'subpestaña romboide'
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button8.Click
    'boton hallar'
    If TextBox16.Text = "" Or TextBox13.Text = "" Or TextBox17.Text = "" Then
        MessageBox.Show("debe ingresar valores en todos los campos")
    Else
        If TextBox16.Text <= 0 Or TextBox13.Text <= 0 Or TextBox17.Text <= 0 Then
            MessageBox.Show("debe ingresar valores mayores que 0")
        Else
            'area del romboide'
            TextBox15.Text = TextBox16.Text * TextBox17.Text
            'perímetro del romboide'
            TextBox14.Text = 2 * (TextBox16.Text * TextBox13.Text)

            Button7.Enabled = True 'se inhabilita el boton hallar'
            Button8.Enabled = False 'se habilita el boton nuevo'
        End If
    End If
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button7.Click
    'boton nuevo: borra los datos'
    TextBox16.Text = ""
    TextBox13.Text = ""
    TextBox17.Text = ""
    TextBox15.Text = ""
    TextBox14.Text = ""

```

```

        Button8.Enabled = True 'se habilita el boton hallar'
        Button7.Enabled = False 'se inhabilita el boton nuevo'
    End Sub
    'subpestaña trapecio'
    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
        'boton hallar'
        If TextBox22.Text = "" Or TextBox19.Text = "" Or TextBox18.Text = "" Or
TextBox23.Text = "" Or TextBox24.Text = "" Then
            MessageBox.Show("debe ingresar valores en todos los campos")
        Else
            If TextBox22.Text <= 0 Or TextBox19.Text <= 0 Or TextBox18.Text <= 0 Or
TextBox23.Text <= 0 Or TextBox24.Text <= 0 Then
                MessageBox.Show("debe ingresar valores mayores que 0")
            Else
                If TextBox22.Text < TextBox19.Text Then
                    MessageBox.Show("el valor de B debe ser mayor que el valor de b")
                Else
                    'area del trapecio'
                    TextBox21.Text = ((TextBox22.Text + TextBox19.Text) / 2) *
TextBox24.Text
                    'perimetro del trapecio'
                    TextBox20.Text = TextBox22.Text + TextBox19.Text + TextBox18.Text +
TextBox23.Text
                    Button9.Enabled = True 'se inhabilita el boton hallar'
                    Button10.Enabled = False 'se habilita el boton nuevo'
                End If
            End If
        End If
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button9.Click
        'boton nuevo: borra los datos'
        TextBox22.Text = ""
        TextBox19.Text = ""
        TextBox18.Text = ""
        TextBox23.Text = ""
        TextBox24.Text = ""
        TextBox21.Text = ""
        TextBox20.Text = ""

        Button10.Enabled = True 'se habilita el boton hallar'
        Button9.Enabled = False 'se inhabilita el boton nuevo'
    End Sub
    'pestaña cuerpos redondos'
    'subpestaña esfera'
    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
        If TextBox28.Text = "" Then
            MessageBox.Show("debe ingresar un valor")
        Else
            If TextBox28.Text <= 0 Then
                MessageBox.Show("debe ingresar valores mayores que 0")
            Else
                'área de la esfera'
                TextBox27.Text = 4 * 3.1416 * (TextBox28.Text * TextBox28.Text)
            End If
        End If
    End Sub

```

```

        'volumen de la esfera'
        TextBox26.Text = (4 * 3.1416 * (TextBox28.Text * TextBox28.Text *
TextBox28.Text)) / 3
        Button11.Enabled = True 'se inhabilita el boton hallar'
        Button12.Enabled = False 'se habilita el boton nuevo'
    End If
End If
End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    'boton nuevo: borra los datos'
    TextBox28.Text = ""
    TextBox27.Text = ""
    TextBox26.Text = ""

    Button12.Enabled = True
    Button11.Enabled = False
End Sub
'subpestaña cilindro'
Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
    'boton hallar'
    If TextBox30.Text = "" Or TextBox31.Text = "" Then
        MessageBox.Show("debe ingresar valores en los dos campos")
    Else
        If TextBox30.Text <= 0 Or TextBox31.Text <= 0 Then
            MessageBox.Show("debe ingresar valores mayores que 0")
        Else
            'area del cilindro'
            TextBox29.Text = (2 * 3.1416 * (TextBox30.Text * TextBox30.Text)) + (2 *
3.1416 * TextBox30.Text * TextBox31.Text)
            'volumen del cilindro'
            TextBox25.Text = 3.1416 * (TextBox30.Text * TextBox30.Text) *
TextBox31.Text

            Button13.Enabled = True 'se inhabilita el boton hallar'
            Button14.Enabled = False 'se habilita el boton nuevo'
        End If
    End If
End Sub

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
    'boton nuevo: borra los datos'
    TextBox30.Text = ""
    TextBox31.Text = ""
    TextBox29.Text = ""
    TextBox25.Text = ""

    Button14.Enabled = True 'se habilita el boton hallar'
    Button13.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'subpestaña cono'
Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
    'boton hallar'

```

```

If TextBox35.Text = "" Or TextBox32.Text = "" Or TextBox36.Text = "" Then
    MessageBox.Show("debe ingresar valores en los dos campos")
Else
    If TextBox35.Text <= 0 Or TextBox32.Text <= 0 Or TextBox36.Text <= 0 Then
        MessageBox.Show("debe ingresar valores mayores que 0")
    Else
        'area del cono'
        TextBox34.Text = (3.1416 * TextBox35.Text * TextBox36.Text) + (3.1416 *
(TextBox35.Text * TextBox35.Text))
        'volumen del cono'
        TextBox33.Text = (3.1416 * (TextBox35.Text * TextBox35.Text) *
TextBox32.Text) / 3
        Button15.Enabled = True 'se inhabilita el boton hallar'
        Button16.Enabled = False 'se habilita el boton nuevo'

        End If
    End If
End Sub

Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
    'boton nuevo: borra los datos'
    TextBox35.Text = ""
    TextBox32.Text = ""
    TextBox36.Text = ""
    TextBox34.Text = ""
    TextBox33.Text = ""
    Button16.Enabled = True 'se habilita el boton hallar'
    Button15.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'pestaña poliedros'
'subpestaña cubo'
Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
    'boton hallar'
    If TextBox41.Text = "" Then
        MessageBox.Show("debe ingresar valores en los dos campos")
    Else
        If TextBox41.Text <= 0 Then
            MessageBox.Show("debe ingresar valores mayores que 0")
        Else
            'area del cubo'
            TextBox40.Text = 6 * (TextBox41.Text * TextBox41.Text * TextBox41.Text)
            'volumen del cubo'
            TextBox39.Text = TextBox41.Text * TextBox41.Text * TextBox41.Text
            Button17.Enabled = True 'se inhabilita el boton hallar'
            Button18.Enabled = False 'se habilita el boton nuevo'

            End If
        End If
    End Sub

Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
    'boton nuevo: borra los datos'
    TextBox41.Text = ""
    TextBox40.Text = ""
    TextBox39.Text = ""

```

```

    Button18.Enabled = True 'se habilita el boton hallar'
    Button17.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'subpestaña paralelepipedo'
Private Sub Button20_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button20.Click
    'boton hallar'
    If TextBox42.Text = "" Or TextBox43.Text = "" Or TextBox44.Text = "" Then
        MessageBox.Show("debe ingresar valores en los dos campos")
    Else
        If TextBox42.Text <= 0 Or TextBox43.Text <= 0 Or TextBox44.Text <= 0 Then
            MessageBox.Show("debe ingresar valores mayores que 0")
        Else
            'area'
            TextBox38.Text = 2 * ((TextBox42.Text * TextBox43.Text) + (TextBox42.Text
* TextBox44.Text) + (TextBox43.Text * TextBox44.Text))
            'volumen'
            TextBox37.Text = TextBox42.Text * TextBox43.Text * TextBox44.Text
            Button19.Enabled = True 'se inhabilita el boton hallar'
            Button20.Enabled = False 'se habilita el boton nuevo'
        End If
    End If
End Sub

Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button19.Click
    'boton nuevo: borra los datos'
    TextBox42.Text = ""
    TextBox43.Text = ""
    TextBox44.Text = ""
    TextBox38.Text = ""
    TextBox37.Text = ""
    Button20.Enabled = True 'se habilita el boton hallar'
    Button19.Enabled = False 'se inhabilita el boton nuevo'
End Sub
'subpestaña piramide cuadrada'
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton1.CheckedChanged
    'caso 1'
    TextBox45.Enabled = True
    TextBox46.Enabled = True
    TextBox49.Enabled = True
    TextBox48.Enabled = True
    TextBox47.Enabled = True

    Button22.Enabled = True
    Button21.Enabled = False

    TextBox50.Enabled = False
    TextBox52.Enabled = False
    TextBox53.Enabled = False
    ComboBox1.Enabled = False

    Button24.Enabled = False
    Button23.Enabled = False

End Sub

```

```

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton2.CheckedChanged
    'caso 2'
    TextBox50.Enabled = True
    TextBox52.Enabled = True
    TextBox53.Enabled = True
    ComboBox1.Enabled = True

    Button24.Enabled = True
    Button23.Enabled = False

    TextBox45.Enabled = False
    TextBox46.Enabled = False
    TextBox49.Enabled = False
    TextBox48.Enabled = False
    TextBox47.Enabled = False

    Button22.Enabled = False
    Button21.Enabled = False

End Sub

Private Sub Button22_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button22.Click
    'caso 1'
    'boton hallar'
    If TextBox45.Text = "" Or TextBox46.Text = "" Or TextBox49.Text = "" Then
        MessageBox.Show("debe ingresar valores en todos los campos")
    Else
        If TextBox45.Text <= 0 Or TextBox46.Text <= 0 Or TextBox49.Text <= 0 Then
            MessageBox.Show("debe ingresar un valor mayor a 0")
        Else
            'area piramide cuadrada'
            TextBox48.Text = (((4 * TextBox49.Text) * TextBox45.Text) / 2) +
(TextBox49.Text * TextBox49.Text)
            TextBox47.Text = ((TextBox49.Text * TextBox49.Text) * TextBox46.Text) / 3
            'volumen piramide cuadrada'
            Button22.Enabled = False 'se inhabilita el boton hallar'
            Button21.Enabled = True 'se habilita el boton nuevo'
        End If
    End If

End Sub

Private Sub Button21_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button21.Click
    'boton nuevo'
    TextBox45.Text = ""
    TextBox46.Text = ""
    TextBox49.Text = ""
    TextBox48.Text = ""
    TextBox47.Text = ""
    Button22.Enabled = True 'se habilita el boton hallar'
    Button21.Enabled = False 'se inhabilita el boton nuevo'

End Sub

```

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    'combobox 1'
    If ComboBox1.Text = "a" Then
        Label147.Text = "a ="

        Label147.Visible = True
        TextBox51.Visible = True
    End If
    If ComboBox1.Text = "h" Then
        Label147.Text = "h ="

        Label147.Visible = True
        TextBox51.Visible = True
    End If

End Sub

Private Sub Button24_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button24.Click
    'caso 2'
    'boton hallar'
    Dim mp1, mp2, mp3 As Double

    If TextBox50.Text = "" Or TextBox51.Text = "" Then
        MessageBox.Show("debe ingresar valores en todos los campos")
    Else
        If TextBox50.Text <= 0 Or TextBox51.Text <= 0 Then
            MessageBox.Show("debe ingresar un valor mayor a 0")
        Else
            If ComboBox1.Text = "a" And TextBox50.Text > TextBox51.Text Then
                MessageBox.Show("a debe ser mayor que l, lado")
            Else
                If ComboBox1.Text = "a" Then
                    mp1 = TextBox50.Text / 2
                    mp2 = (TextBox51.Text * TextBox51.Text) - (mp1 * mp1) 'h'
                    mp3 = Math.Sqrt(mp2)

                    TextBox53.Text = (((4 * TextBox50.Text) * TextBox51.Text) / 2) +
(TextBox50.Text * TextBox50.Text)
                    TextBox52.Text = ((TextBox50.Text * TextBox50.Text) * mp3) / 3
                End If
                If ComboBox1.Text = "h" Then
                    mp1 = TextBox50.Text / 2
                    mp2 = (TextBox51.Text * TextBox51.Text) + (mp1 * mp1) 'a'
                    mp3 = Math.Sqrt(mp2)
                    TextBox53.Text = (((4 * TextBox50.Text) * mp3) / 2) +
(TextBox50.Text * TextBox50.Text)
                    TextBox52.Text = ((TextBox50.Text * TextBox50.Text) *
TextBox51.Text) / 3
                End If
                Button23.Enabled = True
                Button24.Enabled = False
            End If
        End If
    End If
End Sub

```

```

End Sub

Private Sub Button23_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button23.Click
    'boton nuevo'
    TextBox50.Text = ""
    TextBox53.Text = ""
    TextBox52.Text = ""
    TextBox51.Text = ""
    ComboBox1.Text = "elija la variable conocida"
    TextBox51.Visible = False
    Label147.Visible = False
    Button24.Enabled = True
    Button23.Enabled = False
End Sub
'subpestaña piramide regular'
Private Sub ComboBox3_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox3.SelectedIndexChanged
    'combobox 3'
    If ComboBox3.Text = "h" Then
        Label178.Text = "h ="

        Label178.Visible = True
        TextBox57.Visible = True
    End If
    If ComboBox3.Text = "ar1" Then
        Label178.Text = "ar1 ="

        Label178.Visible = True
        TextBox57.Visible = True
    End If
End Sub

Private Sub Button26_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button26.Click
    'boton hallar'
    Dim mp7, mp8, mp9, mp10, mp11, mp12, mp13, mp14, mp15, mp16, mp17, mp18, mp19,
mp20, mp21, mp22, mp23, mp24 As Double

    If ComboBox2.Text = 5 Then
        Label177.Text = "piramide pentagonal"
        Label176.Text = "base de 5 lados"
        Label177.Visible = True
        Label176.Visible = True
    End If
    If ComboBox2.Text = 6 Then
        Label177.Text = "piramide hexagonal"
        Label176.Text = "base de 6 lados"
        Label177.Visible = True
        Label176.Visible = True
    End If
    If ComboBox2.Text = 7 Then
        Label177.Text = "piramide heptagonal"
        Label176.Text = "base de 7 lados"
        Label177.Visible = True
        Label176.Visible = True
    End If
    If ComboBox2.Text = 8 Then

```

```

Label177.Text = "piramide octagonal"
Label176.Text = "base de 8 lados"
Label177.Visible = True
Label176.Visible = True
End If

If TextBox54.Text = "" Or TextBox57.Text = "" Then
    MessageBox.Show("debe ingresar valores en todos los campos")
Else
    If TextBox54.Text <= 0 Or TextBox57.Text <= 0 Then
        MessageBox.Show("debe ingresar un valor mayor a 0")
    Else
        If ComboBox3.Text = "ar1" And TextBox54.Text > TextBox57.Text Then
            MessageBox.Show("ar1 debe ser mayor que l, lado")
        Else
            If ComboBox3.Text = "ar1" Then
                mp7 = TextBox54.Text / 2
                mp8 = (TextBox54.Text * TextBox54.Text) - (mp7 * mp7) 'apotema
                base'

                mp9 = Math.Sqrt(mp8) 'apotema base'

                mp10 = (TextBox57.Text * TextBox57.Text) - (mp7 * mp7)
                mp11 = Math.Sqrt(mp10) 'apotema lateral'

                mp12 = ((ComboBox2.Text * TextBox54.Text) * (mp9 + mp11)) / 2
                'area'

                mp13 = (mp11 * mp11) - (mp9 * mp9)
                mp14 = Math.Sqrt(mp13) 'altura'
                base'
                mp15 = ((ComboBox2.Text * TextBox54.Text) * mp9) / 2 'apotema

                mp16 = ((1 * mp15) / 3) * mp14 'volumen'

                TextBox55.Text = mp12
                TextBox56.Text = mp16
            End If
            If ComboBox3.Text = "h" Then
                mp17 = TextBox54.Text / 2
                base'
                mp18 = (TextBox54.Text * TextBox54.Text) - (mp17 * mp17) 'apotema

                mp19 = Math.Sqrt(mp18) 'apotema base'
                lateral'
                mp20 = (TextBox57.Text * TextBox57.Text) + (mp19 * mp19) 'apotema

                mp21 = Math.Sqrt(mp20) 'apotema lateral'
                'area'
                mp22 = ((ComboBox2.Text * TextBox54.Text) * (mp19 + mp21)) / 2

                mp23 = ((ComboBox2.Text * TextBox54.Text) * mp19) / 2 'apotema
                base'
                mp24 = ((1 * mp23) / 3) * TextBox57.Text 'volumen'

                TextBox55.Text = mp22
                TextBox56.Text = mp24
            End If
            Button25.Enabled = True
            Button26.Enabled = False
        End If
    End If

```

```

        End If
    End If
End Sub

Private Sub Button25_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button25.Click
    'boton nuevo'
    TextBox54.Text = ""
    TextBox57.Text = ""
    TextBox57.Visible = False
    Label178.Visible = False
    Label177.Visible = False
    Label176.Visible = False
    TextBox55.Text = ""
    TextBox56.Text = ""
    ComboBox2.Text = "elija el numero de lados de la base"
    ComboBox3.Text = "elija la variable conocida"
    Button26.Enabled = True
    Button25.Enabled = False
End Sub

```

Código de la interfaz trigonometría (solo parte del código por motivo de extensión)

```

Public Class TRIGONOMETRIA
    'pestaña teorema de Pitágoras'
    'boton mostrar'
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If CheckBox1.Checked And CheckBox2.Checked And CheckBox3.Checked Then

            TextBox2.Visible = False
            TextBox1.Visible = False
            Label6.Visible = False
            Label7.Visible = False

            MessageBox.Show("debe seleccionar solo dos variables")

            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
        Else
            If CheckBox1.Checked And CheckBox2.Checked Then
                TextBox1.Visible = True
                TextBox2.Visible = True
                Label6.Text = "x ="
                Label6.Visible = True
                Label7.Text = "y ="
                Label7.Visible = True
            End If
            If CheckBox1.Checked And CheckBox3.Checked Then
                TextBox1.Visible = True
                TextBox2.Visible = True
                Label6.Text = "x ="
                Label6.Visible = True
                Label7.Text = "h ="
                Label7.Visible = True
            End If
        End If
    End Sub
End Class

```

```

End If
If CheckBox2.Checked And CheckBox3.Checked Then
    TextBox1.Visible = True
    TextBox2.Visible = True
    Label6.Text = "y ="
    Label6.Visible = True
    Label7.Text = "h ="
    Label7.Visible = True
End If
Button1.Enabled = True
Button2.Enabled = False

End If

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    'boton hallar'
    Dim q1, q2 As Double
    Dim w1 As String

    If TextBox1.Text = "" Or TextBox2.Text = "" Then
        MessageBox.Show("debe ingresar valores en los 2 campos")
    Else
        If TextBox1.Text = 0 Or TextBox2.Text = 0 Then
            MessageBox.Show("debe ingresar valores diferentes de 0")
        Else
            If TextBox1.Text < 0 Or TextBox2.Text < 0 Then
                MessageBox.Show("debe ingresar valores mayores que 0")
            Else
                If CheckBox1.Checked And CheckBox2.Checked Then
                    Label4.Text = "h^2 = x^2 + y^2"
                    Label5.Text = "h^2 =" + TextBox1.Text + "^2 + " + TextBox2.Text +
"
^2"
                    q1 = (TextBox1.Text * TextBox1.Text) + (TextBox2.Text *
TextBox2.Text)
                    q2 = Math.Sqrt(q1)
                    w1 = q2
                    Label8.Text = "h = " + w1
                    Label4.Visible = True
                    Label5.Visible = True
                    Label8.Visible = True
                    Button4.Enabled = True
                    Button1.Enabled = False

                End If
                If CheckBox1.Checked And CheckBox3.Checked Then
                    If TextBox2.Text < TextBox1.Text Then
                        Button1.Enabled = True
                        MessageBox.Show("el valor de h debe ser mayor que el valor de
x")
                        TextBox2.Text = ""

                    Else
                        Label4.Text = "y^2 = h^2 - x^2"
                        Label5.Text = "y^2 =" + TextBox2.Text + "^2 - " +
TextBox1.Text + "^2"

```

```

        TextBox1.Text)
        q1 = (TextBox2.Text * TextBox2.Text) - (TextBox1.Text *
        TextBox1.Text)
        q2 = Math.Sqrt(q1)
        w1 = q2
        Label8.Text = "y = " + w1
        Label4.Visible = True
        Label5.Visible = True
        Label8.Visible = True
        Button4.Enabled = True
        Button1.Enabled = False
    End If

End If
If CheckBox2.Checked And CheckBox3.Checked Then
    If TextBox1.Text > TextBox2.Text Then
        Button1.Enabled = True
        MessageBox.Show("el valor de h debe ser mayor que el valor de
y")
        TextBox2.Text = ""

    Else

        Label4.Text = "x^2 = h^2 - y^2"
        Label5.Text = "x^2 =" + TextBox2.Text + "^2 - " +
        TextBox1.Text + "^2"
        q1 = (TextBox2.Text * TextBox2.Text) - (TextBox1.Text *
        TextBox1.Text)
        q2 = Math.Sqrt(q1)
        w1 = q2
        Label8.Text = "x = " + w1
        Label4.Visible = True
        Label5.Visible = True
        Label8.Visible = True
        Button4.Enabled = True
        Button1.Enabled = False
    End If

End If
End If

End If
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    'BOTON PROCESO'
    Dim q4, q5, q6 As Double
    Dim w4, w5, w6, w7 As String

    Label19.Text = Label4.Text
    Label110.Text = Label5.Text
    If CheckBox1.Checked And CheckBox2.Checked Then
        Label111.Text = "h = " + "√ ( ( " + TextBox1.Text + "^2" + " )" + " + " + " + "( "
+ TextBox2.Text + "^2 ) )"
        q4 = TextBox1.Text * TextBox1.Text
        q5 = TextBox2.Text * TextBox2.Text
        w4 = q4

```

```

w5 = q5
w6 = q4 + q5
q6 = Math.Sqrt(q4 + q5)
w7 = q6

Label12.Text = "h = " + "√ ( " + w4 + " + " + w5 + " )"
Label13.Text = "h = " + "√ ( " + w6 + " )"
Label14.Text = "h = " + w7

End If
If CheckBox1.Checked And CheckBox3.Checked Then
    Label11.Text = "y = " + "√ ( ( " + TextBox2.Text + "^2" + " )" + " - " + "( "
+ TextBox1.Text + "^2 ) )"
    q4 = TextBox1.Text * TextBox1.Text
    q5 = TextBox2.Text * TextBox2.Text
    w4 = q4
    w5 = q5
    w6 = q5 - q4
    q6 = Math.Sqrt(q5 - q4)
    w7 = q6

    Label12.Text = "y = " + "√ ( " + w5 + " - " + w4 + " )"
    Label13.Text = "y = " + "√ ( " + w6 + " )"
    Label14.Text = "y = " + w7

End If
If CheckBox2.Checked And CheckBox3.Checked Then
    Label11.Text = "x = " + "√ ( ( " + TextBox2.Text + "^2" + " )" + " - " + "( "
+ TextBox1.Text + "^2 ) )"
    q4 = TextBox1.Text * TextBox1.Text
    q5 = TextBox2.Text * TextBox2.Text
    w4 = q4
    w5 = q5
    w6 = q5 - q4
    q6 = Math.Sqrt(q5 - q4)
    w7 = q6

    Label12.Text = "x = " + "√ ( " + w5 + " - " + w4 + " )"
    Label13.Text = "x = " + "√ ( " + w6 + " )"
    Label14.Text = "x = " + w7

End If
Label19.Visible = True
Label10.Visible = True
Label11.Visible = True
Label12.Visible = True
Label13.Visible = True
Label14.Visible = True

Button3.Enabled = True
Button4.Enabled = False

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    'boton nuevo'
    TextBox1.Visible = False

```

```

TextBox2.Visible = False
TextBox1.Text = ""
TextBox2.Text = ""

Label16.Visible = False
Label17.Visible = False

Label14.Visible = False
Label15.Visible = False
Label18.Visible = False

Label19.Visible = False
Label10.Visible = False
Label11.Visible = False
Label12.Visible = False
Label13.Visible = False
Label14.Visible = False

Button2.Enabled = True
Button3.Enabled = False
CheckBox1.Checked = False
CheckBox2.Checked = False
CheckBox3.Checked = False
End Sub
'pestaña funciones trigonometricas'
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
'BOTON HALLAR'
Dim e1, e2, e3, e4, e5, e6 As Double

If TextBox3.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "" Then
    MessageBox.Show(" debe ingresar valores en todos los campos")
Else
    If TextBox3.Text = 0 Or TextBox4.Text = 0 Or TextBox5.Text = 0 Then
        MessageBox.Show(" debe ingresar valores diferentes de 0")
    Else
        TextBox6.Text = "y"
        TextBox7.Text = "h"
        TextBox9.Text = TextBox4.Text
        TextBox8.Text = TextBox5.Text
        e1 = TextBox4.Text / TextBox5.Text
        TextBox10.Text = e1

        TextBox15.Text = "x"
        TextBox14.Text = "h"
        TextBox13.Text = TextBox3.Text
        TextBox12.Text = TextBox5.Text
        e2 = TextBox3.Text / TextBox5.Text
        TextBox11.Text = e2

        TextBox20.Text = "y"
        TextBox19.Text = "x"
        TextBox18.Text = TextBox4.Text
        TextBox17.Text = TextBox3.Text
        e3 = TextBox4.Text / TextBox3.Text
        TextBox16.Text = e3

        TextBox25.Text = "x"
    End If
End If

```

```
TextBox24.Text = "y"  
TextBox23.Text = TextBox3.Text  
TextBox22.Text = TextBox4.Text  
e4 = TextBox3.Text / TextBox4.Text  
TextBox21.Text = e4
```

```
TextBox30.Text = "h"  
TextBox29.Text = "x"  
TextBox28.Text = TextBox5.Text  
TextBox27.Text = TextBox3.Text  
e5 = TextBox5.Text / TextBox3.Text  
TextBox26.Text = e5
```

```
TextBox35.Text = "h"  
TextBox34.Text = "y"  
TextBox33.Text = TextBox5.Text  
TextBox32.Text = TextBox4.Text  
e6 = TextBox5.Text / TextBox4.Text  
TextBox31.Text = e6
```

```
Label153.Text = TextBox5.Text  
Label154.Text = TextBox3.Text  
Label155.Text = TextBox4.Text  
Label153.Visible = True  
Label154.Visible = True  
Label155.Visible = True  
PictureBox6.Visible = True
```

```
TextBox6.Visible = True  
TextBox7.Visible = True  
TextBox15.Visible = True  
TextBox14.Visible = True  
TextBox9.Visible = True  
TextBox8.Visible = True  
TextBox13.Visible = True  
TextBox12.Visible = True  
TextBox10.Visible = True  
TextBox11.Visible = True  
TextBox20.Visible = True  
TextBox19.Visible = True  
TextBox25.Visible = True  
TextBox24.Visible = True  
TextBox30.Visible = True  
TextBox29.Visible = True  
TextBox35.Visible = True  
TextBox34.Visible = True  
TextBox18.Visible = True  
TextBox17.Visible = True  
TextBox23.Visible = True  
TextBox22.Visible = True  
TextBox28.Visible = True  
TextBox27.Visible = True  
TextBox33.Visible = True  
TextBox32.Visible = True  
TextBox16.Visible = True  
TextBox21.Visible = True  
TextBox26.Visible = True
```

```
        TextBox31.Visible = True

        Label123.Visible = True
        Label133.Visible = True
        Label131.Visible = True
        Label125.Visible = True
        Label130.Visible = True
        Label126.Visible = True
        Label129.Visible = True
        Label127.Visible = True
        Label128.Visible = True
        Label132.Visible = True
        Label138.Visible = True
        Label143.Visible = True
        Label148.Visible = True
        Label137.Visible = True
        Label142.Visible = True
        Label147.Visible = True
        Label152.Visible = True
        Label136.Visible = True
        Label141.Visible = True
        Label146.Visible = True
        Label151.Visible = True
        Label135.Visible = True
        Label140.Visible = True
        Label145.Visible = True
        Label150.Visible = True
        Label134.Visible = True
        Label139.Visible = True
        Label144.Visible = True
        Label149.Visible = True
        Label124.Visible = True

        Button5.Enabled = False
        Button6.Enabled = True
    End If
End If
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
    'boton nuevo'
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""

    TextBox6.Visible = False
    TextBox7.Visible = False
    TextBox15.Visible = False
    TextBox14.Visible = False
    TextBox9.Visible = False
    TextBox8.Visible = False
    TextBox13.Visible = False
    TextBox12.Visible = False
    TextBox10.Visible = False
    TextBox11.Visible = False
    TextBox20.Visible = False
    TextBox19.Visible = False

```

```
TextBox25.Visible = False  
TextBox24.Visible = False  
TextBox30.Visible = False  
TextBox29.Visible = False  
TextBox35.Visible = False  
TextBox34.Visible = False  
TextBox18.Visible = False  
TextBox17.Visible = False  
TextBox23.Visible = False  
TextBox22.Visible = False  
TextBox28.Visible = False  
TextBox27.Visible = False  
TextBox33.Visible = False  
TextBox32.Visible = False  
TextBox16.Visible = False  
TextBox21.Visible = False  
TextBox26.Visible = False  
TextBox31.Visible = False
```

```
Label123.Visible = False  
Label133.Visible = False  
Label131.Visible = False  
Label125.Visible = False  
Label130.Visible = False  
Label126.Visible = False  
Label129.Visible = False  
Label127.Visible = False  
Label128.Visible = False  
Label132.Visible = False  
Label138.Visible = False  
Label143.Visible = False  
Label148.Visible = False  
Label137.Visible = False  
Label142.Visible = False  
Label147.Visible = False  
Label152.Visible = False  
Label136.Visible = False  
Label141.Visible = False  
Label146.Visible = False  
Label151.Visible = False  
Label135.Visible = False  
Label140.Visible = False  
Label145.Visible = False  
Label150.Visible = False  
Label134.Visible = False  
Label139.Visible = False  
Label144.Visible = False  
Label149.Visible = False  
Label124.Visible = False
```

```
Label153.Visible = False  
Label154.Visible = False  
Label155.Visible = False  
PictureBox6.Visible = False
```

```
Button5.Enabled = True  
Button6.Enabled = False
```

```
End Sub
```

Código interfaz funciones (solo una parte por motivo de extensión)

```
Public Class Form2
    'pestaña lineal'
    'subpestaña y = ax + b'
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        'boton continuar'
        Dim q1 As Double
        Dim w1 As String
        If TextBox1.Text = "" Or TextBox2.Text = "" Then
            MessageBox.Show("debe ingresar valores en los campos")
        Else
            If TextBox2.Text > 0 Then
                Label16.Text = "y = " + TextBox1.Text + "x + " + TextBox2.Text
                Label16.Visible = True
            End If
            If TextBox2.Text < 0 Then
                q1 = TextBox2.Text * (-1)
                w1 = q1
                Label16.Text = "y = " + TextBox1.Text + "x - " + w1
                Label16.Visible = True
            End If
            TextBox1.Enabled = False
            TextBox2.Enabled = False

            Button1.Enabled = False
            Button2.Enabled = True
            Button3.Enabled = False
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        'boton hallar'
        Dim q2, q3, q4 As Double
        Dim w2, w3, w4 As String
        If TextBox3.Text = "" Then
            MessageBox.Show("debe ingresar valores en los campos")
        Else
            TextBox4.Text = (TextBox1.Text * TextBox3.Text) + TextBox2.Text

            'proceso'
            Label10.Text = Label16.Text
            Label11.Text = "damos valores a x"
            Label12.Text = "cuando x = " + TextBox3.Text
            If TextBox2.Text > 0 Then
                q3 = TextBox1.Text * TextBox3.Text 'multiplicacion'
                q4 = q3 + TextBox2.Text 'suma o resta'
                w3 = q3
                w4 = q4
                Label13.Text = "y = ( " + TextBox1.Text + " * ( " + TextBox3.Text + " ) )
+ " + TextBox2.Text
                Label14.Text = "y = " + w3 + " + " + TextBox2.Text
            End If
        End If
    End Sub
End Class
```

```

        Label15.Text = "y = " + w4
    End If
    If TextBox2.Text < 0 Then
        q3 = TextBox1.Text * TextBox3.Text 'multiplicacion'
        q4 = q3 + TextBox2.Text 'suma o resta'
        w3 = q3
        w4 = q4
        q2 = TextBox2.Text * (-1)
        w2 = q2
        Label13.Text = "y = ( " + TextBox1.Text + " * ( " + TextBox3.Text + " ) )
- " + w2
        Label14.Text = "y = " + w3 + " - " + w2
        Label15.Text = "y = " + w4
    End If
    Label10.Visible = True
    Label11.Visible = True
    Label12.Visible = True
    Label13.Visible = True
    Label14.Visible = True
    Label15.Visible = True

    Button1.Enabled = False
    Button2.Enabled = False
    Button3.Enabled = True
End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    'boton almacenar'
    DataGridView1.Rows.Add(TextBox3.Text, TextBox4.Text)
    TextBox3.Text = ""
    TextBox4.Text = ""
    Button1.Enabled = False
    Button2.Enabled = True
    Button3.Enabled = False
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    'boton nuevo'
    TextBox1.Enabled = True
    TextBox2.Enabled = True
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox1.Text = ""
    TextBox2.Text = ""

    Label10.Visible = False
    Label11.Visible = False
    Label12.Visible = False
    Label13.Visible = False
    Label14.Visible = False
    Label15.Visible = False
    Label16.Visible = False

    DataGridView1.Rows.Clear()
    Button1.Enabled = True

```

```

        Button2.Enabled = False
        Button3.Enabled = False
    End Sub
    'subpestaña y = x + b'
    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button7.Click
        'boton continuar'
        Dim e1 As Double
        Dim r1 As String
        If TextBox7.Text = "" Then
            MessageBox.Show("debe ingresar valores en los campos")
        Else
            If TextBox7.Text > 0 Then
                Label21.Text = "y = x + " + TextBox7.Text
                Label21.Visible = True
            End If
            If TextBox7.Text < 0 Then
                e1 = TextBox7.Text * (-1)
                r1 = e1
                Label21.Text = "y = x - " + r1
                Label21.Visible = True
            End If
            TextBox7.Enabled = False

            Button7.Enabled = False
            Button6.Enabled = True
            Button5.Enabled = False
        End If
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        'boton hallar'
        Dim e2, e3, e4 As Double
        Dim r2, r3, r4 As String
        If TextBox6.Text = "" Then
            MessageBox.Show("debe ingresar valores en los campos")
        Else
            TextBox5.Text = Val(TextBox6.Text) + Val(TextBox7.Text)

            'proceso'
            Label27.Text = Label21.Text
            Label28.Text = "damos valores a x"
            Label29.Text = "cuando x = " + TextBox6.Text
            If TextBox7.Text > 0 Then
                e3 = TextBox6.Text 'multiplicacion'
                e4 = e3 + TextBox7.Text 'suma o resta'
                r3 = e3
                r4 = e4
                Label130.Text = "y = " + TextBox6.Text + " + " + TextBox7.Text

                Label131.Text = "y = " + r4
            End If
            If TextBox7.Text < 0 Then
                e3 = TextBox6.Text 'multiplicacion'
                e4 = e3 + TextBox7.Text 'suma o resta'
                r3 = e3
                r4 = e4
            End If
        End Sub
    End Sub

```

```

        e2 = TextBox7.Text * (-1)
        r2 = e2
        Label130.Text = "y = " + TextBox6.Text + " - " + r2

        Label131.Text = "y = " + r4
    End If
    Label127.Visible = True
    Label128.Visible = True
    Label129.Visible = True
    Label130.Visible = True
    Label131.Visible = True

    Button7.Enabled = False
    Button6.Enabled = False
    Button5.Enabled = True
End If
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    'almacenar'
    DataGridView2.Rows.Add(TextBox6.Text, TextBox5.Text)
    TextBox6.Text = ""
    TextBox5.Text = ""
    Button7.Enabled = False
    Button6.Enabled = True
    Button5.Enabled = False
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button8.Click
    'nuevo'
    TextBox7.Enabled = True

    TextBox7.Text = ""
    TextBox6.Text = ""
    TextBox5.Text = ""

    Label127.Visible = False
    Label128.Visible = False
    Label129.Visible = False
    Label130.Visible = False
    Label131.Visible = False

    Label121.Visible = False

    DataGridView2.Rows.Clear()
    Button7.Enabled = True
    Button6.Enabled = False
    Button5.Enabled = False
End Sub
'subpestaña y = ax'
Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
    'continuar'

    If TextBox11.Text = "" Then
        MessageBox.Show("debe ingresar valores en los campos")
    End If
End Sub

```

```

Else
    Label41.Text = "y = " + TextBox11.Text + "x"
    Label41.Visible = True

    TextBox11.Enabled = False

    Button12.Enabled = False
    Button11.Enabled = True
    Button10.Enabled = False
End If
End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    'hallar'
    Dim t1 As Double
    Dim y1 As String
    If TextBox9.Text = "" Then
        MessageBox.Show("debe ingresar valores en los campos")
    Else
        TextBox8.Text = (TextBox11.Text * TextBox9.Text)
        t1 = TextBox11.Text * TextBox9.Text 'multiplicacion'
        y1 = t1
        'proceso'
        Label32.Text = Label41.Text
        Label33.Text = "damos valores a x"
        Label34.Text = "cuando x = " + TextBox9.Text
        Label35.Text = "y = ( " + TextBox11.Text + " * ( " + TextBox9.Text + " ) )"
        Label36.Text = "y = " + y1

        Label32.Visible = True
        Label33.Visible = True
        Label34.Visible = True
        Label35.Visible = True
        Label36.Visible = True

        Button12.Enabled = False
        Button11.Enabled = False
        Button10.Enabled = True
    End If
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
    'almacenar'
    DataGridView3.Rows.Add(TextBox9.Text, TextBox8.Text)
    TextBox8.Text = ""
    TextBox9.Text = ""
    Button12.Enabled = False
    Button11.Enabled = True
    Button10.Enabled = False
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button9.Click
    'nuevo'
    TextBox11.Enabled = True

```

```

    TextBox11.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""

    Label32.Visible = False
    Label33.Visible = False
    Label34.Visible = False
    Label35.Visible = False
    Label36.Visible = False

    Label41.Visible = False

    DataGridView3.Rows.Clear()
    Button12.Enabled = True
    Button11.Enabled = False
    Button10.Enabled = False
End Sub
'subpestaña y=x'
Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
    'hallar'
    If TextBox12.Text = "" Then
        MessageBox.Show("debe ingresar valores en los campos")
    Else
        TextBox10.Text = TextBox12.Text

        'proceso'
        Label47.Text = "y = x"
        Label48.Text = "damos valores a x"
        Label49.Text = "cuando x = " + TextBox12.Text
        Label50.Text = "y = " + TextBox12.Text

        Label47.Visible = True
        Label48.Visible = True
        Label49.Visible = True
        Label50.Visible = True

        Button15.Enabled = False

        Button14.Enabled = True
    End If
End Sub

Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
    'almacenar'
    DataGridView4.Rows.Add(TextBox12.Text, TextBox10.Text)
    TextBox12.Text = ""
    TextBox10.Text = ""

    Button15.Enabled = True

    Button14.Enabled = False
End Sub

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
    'nuevo'

```

```

TextBox12.Text = ""
TextBox10.Text = ""

DataGridView4.Rows.Clear()

Label147.Visible = False
Label148.Visible = False
Label149.Visible = False
Label150.Visible = False

End Sub
'pestaña cuadrática'
'subpestaña y = ax2 + bx + c'
Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button19.Click
'continuar'
Dim u1, u2 As Double
Dim i1, i2 As String

If TextBox16.Text = "" Or TextBox15.Text = "" Or TextBox17.Text = "" Then
    MessageBox.Show("debe ingresar valores en los campos")
Else
    If TextBox15.Text > 0 And TextBox17.Text > 0 Then
        Label166.Text = "y = " + TextBox16.Text + "x^2 + " + TextBox15.Text + "x +
" + TextBox17.Text
        Label166.Visible = True
    End If
    If TextBox15.Text > 0 And TextBox17.Text < 0 Then
        u1 = TextBox17.Text * (-1)
        i1 = u1

        Label166.Text = "y = " + TextBox16.Text + "x^2 + " + TextBox15.Text + " -
" + i1
        Label166.Visible = True
    End If
    If TextBox15.Text < 0 And TextBox17.Text > 0 Then
        u1 = TextBox15.Text * (-1)
        i1 = u1
        Label166.Text = "y = " + TextBox16.Text + "x^2 - " + i1 + "x + " +
TextBox17.Text
        Label166.Visible = True
    End If
    If TextBox15.Text < 0 And TextBox17.Text < 0 Then
        u1 = TextBox15.Text * (-1)
        i1 = u1
        u2 = TextBox17.Text * (-1)
        i2 = u2
        Label166.Text = "y = " + TextBox16.Text + "x^2 - " + i1 + "x - " + i2
        Label166.Visible = True
    End If
    TextBox16.Enabled = False
    TextBox15.Enabled = False
    TextBox17.Enabled = False

    Button19.Enabled = False
    Button18.Enabled = True
    Button17.Enabled = False

```

```

    End If
End Sub

Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
    'hallar'
    Dim u4, u5, u6, u7, u8, u9, u10, u11 As Double
    Dim i4, i5, i6, i7, i8, i9, i10, i11 As String

    If TextBox14.Text = "" Then
        MessageBox.Show("debe ingresar valores en los campos")
    Else
        'hallar y'
        TextBox13.Text = (TextBox16.Text * (TextBox14.Text * TextBox14.Text)) +
        (TextBox15.Text * TextBox14.Text) + TextBox17.Text
        'proceso'
        Label73.Text = Label66.Text
        Label74.Text = "damos valores a x"
        Label75.Text = "cuando x = " + TextBox14.Text
        If TextBox15.Text > 0 And TextBox17.Text > 0 Then
            Label76.Text = "y = ( " + TextBox16.Text + " * ( " + TextBox14.Text +
"x^2 ) ) + ( " + TextBox15.Text + " * ( " + TextBox14.Text + " ) ) + " + TextBox17.Text

        End If
        If TextBox15.Text > 0 And TextBox17.Text < 0 Then
            u10 = TextBox17.Text * (-1)
            i10 = u10
            Label76.Text = "y = ( " + TextBox16.Text + " * ( " + TextBox14.Text +
"x^2 ) ) + ( " + TextBox15.Text + " * ( " + TextBox14.Text + " ) ) - " + i10

        End If
        If TextBox15.Text < 0 And TextBox17.Text > 0 Then
            u10 = TextBox15.Text * (-1)
            i10 = u10
            Label76.Text = "y = ( " + TextBox16.Text + " * ( " + TextBox14.Text +
"x^2 ) ) - ( " + i10 + " * ( " + TextBox14.Text + " ) ) + " + TextBox17.Text

        End If
        If TextBox15.Text < 0 And TextBox17.Text < 0 Then
            u10 = TextBox15.Text * (-1)
            i10 = u10
            u11 = TextBox17.Text * (-1)
            i11 = u11
            Label76.Text = "y = ( " + TextBox16.Text + " * ( " + TextBox14.Text +
"x^2 ) ) - ( " + i10 + " * ( " + TextBox14.Text + " ) ) - " + i11

        End If

        u4 = TextBox14.Text * TextBox14.Text 'x2'
        u5 = TextBox16.Text * u4 'ax2'
        u6 = TextBox15.Text * TextBox14.Text 'bx'
        u7 = u5 + u6 + TextBox17.Text 'ax2+bx+c'

        i4 = u4 'x2'
        i5 = u5 'ax2'
        i6 = u6 'bx'
        i7 = u7 'ax2+bx+c'
    End If
End Sub

```

```

        If u6 > 0 And TextBox17.Text > 0 Then
            Label177.Text = "y = ( " + TextBox16.Text + " * " + i4 + " ) + " + i6 + "
+ " + TextBox17.Text
            Label178.Text = "y = " + i5 + " + " + i6 + " + " + TextBox17.Text
            Label179.Text = "y = " + i7
        End If
        If u6 > 0 And TextBox17.Text < 0 Then
            u8 = TextBox17.Text * (-1)
            i8 = u8
            Label177.Text = "y = ( " + TextBox16.Text + " * " + i4 + " ) + " + i6 + "
- " + i8
            Label178.Text = "y = " + i5 + " + " + i6 + " - " + i8
            Label179.Text = "y = " + i7
        End If
        If u6 < 0 And TextBox17.Text > 0 Then
            u8 = u6 * (-1)
            i8 = u8
            Label177.Text = "y = ( " + TextBox16.Text + " * " + i4 + " ) - " + i8 + "
+ " + TextBox17.Text
            Label178.Text = "y = " + i5 + " - " + i8 + " + " + TextBox17.Text
            Label179.Text = "y = " + i7
        End If
        If u6 < 0 And TextBox17.Text < 0 Then
            u8 = u6 * (-1)
            i8 = u8
            u9 = TextBox17.Text * (-1)
            i9 = u9
            Label177.Text = "y = ( " + TextBox16.Text + " * " + i4 + " ) - " + i8 + "
- " + i9
            Label178.Text = "y = " + i5 + " - " + i8 + " - " + i9
            Label179.Text = "y = " + i7
        End If
        Label173.Visible = True
        Label174.Visible = True
        Label175.Visible = True
        Label176.Visible = True
        Label177.Visible = True
        Label178.Visible = True
        Label179.Visible = True

        Button19.Enabled = False
        Button18.Enabled = False
        Button17.Enabled = True

    End If
End Sub

Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
    'almacenar'
    DataGridView5.Rows.Add(TextBox14.Text, TextBox13.Text)
    TextBox14.Text = ""
    TextBox13.Text = ""
    Button19.Enabled = False
    Button18.Enabled = True
    Button17.Enabled = False
End Sub

```

```

Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
    'nuevo'
    TextBox16.Enabled = True
    TextBox15.Enabled = True
    TextBox17.Enabled = True
    TextBox16.Text = ""
    TextBox15.Text = ""
    TextBox17.Text = ""
    TextBox14.Text = ""
    TextBox13.Text = ""

    Label73.Visible = False
    Label74.Visible = False
    Label75.Visible = False
    Label76.Visible = False
    Label77.Visible = False
    Label78.Visible = False
    Label79.Visible = False
    Label66.Visible = False

    DataGridView5.Rows.Clear()
    Button19.Enabled = True
    Button18.Enabled = False
    Button17.Enabled = False

End Sub

```

Código de la interfaz consultas (solo parte del código por motivo de extensión)

```

Public Class CONSULTAS
    Private Property puntaje As Integer
    Private Property acertado As Integer
    'pestaña actividades'
    'subpestaña ecuación lineal'
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        'boton ver resultado'
        puntaje = 0
        acertado = 0
        'puntaje'
        If RadioButton1.Checked = True Then
            puntaje = puntaje + 5
        End If
        If RadioButton6.Checked = True Then
            puntaje = puntaje + 5
        End If
        If RadioButton7.Checked = True Then
            puntaje = puntaje + 5
        End If
        If RadioButton11.Checked = True Then
            puntaje = puntaje + 5
        End If
        If RadioButton13.Checked = True Then
            puntaje = puntaje + 5
        End If
    End Sub
End Class

```

```
End If
If RadioButton17.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton20.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton23.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton25.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton30.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton31.Checked = True Then
    puntaje = puntaje + 5
End If
'repuestas acertadas'
If RadioButton1.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton6.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton7.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton11.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton13.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton17.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton20.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton23.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton25.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton30.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton31.Checked = True Then
    acertado = acertado + 1
End If
Label2.Text = "su puntaje es: " + Str(puntaje) + " de 55"
Label4.Text = "su acertacion es: " + Str(acertado) + " de 11"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
```

```

Label4.Visible = True

Button1.Enabled = False
Button2.Enabled = True
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    'nuevo intento'
    Label1.Visible = False
    Label2.Visible = False
    Label3.Visible = False
    Label4.Visible = False
    RadioButton1.Checked = False
    RadioButton2.Checked = False
    RadioButton3.Checked = False
    RadioButton4.Checked = False
    RadioButton5.Checked = False
    RadioButton6.Checked = False
    RadioButton7.Checked = False
    RadioButton8.Checked = False
    RadioButton9.Checked = False
    RadioButton10.Checked = False
    RadioButton11.Checked = False
    RadioButton12.Checked = False
    RadioButton13.Checked = False
    RadioButton14.Checked = False
    RadioButton15.Checked = False
    RadioButton16.Checked = False
    RadioButton17.Checked = False
    RadioButton18.Checked = False
    RadioButton19.Checked = False
    RadioButton20.Checked = False
    RadioButton21.Checked = False
    RadioButton22.Checked = False
    RadioButton23.Checked = False
    RadioButton24.Checked = False
    RadioButton25.Checked = False
    RadioButton26.Checked = False
    RadioButton27.Checked = False
    RadioButton28.Checked = False
    RadioButton29.Checked = False
    RadioButton30.Checked = False
    RadioButton31.Checked = False
    RadioButton32.Checked = False
    RadioButton33.Checked = False

    Button1.Enabled = True
    Button2.Enabled = False
End Sub
'subpestaña ecuacion cudratica'
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    'ver resultado'
    puntaje = 0
    acertado = 0
    'puntaje'

```

```
If RadioButton35.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton37.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton42.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton43.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton47.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton51.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton54.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton55.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton59.Checked = True Then
    puntaje = puntaje + 5
End If
If RadioButton61.Checked = True Then
    puntaje = puntaje + 5
End If

'repuestas acertadas'
If RadioButton35.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton37.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton42.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton43.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton47.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton51.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton54.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton55.Checked = True Then
    acertado = acertado + 1
End If
If RadioButton59.Checked = True Then
    acertado = acertado + 1
```

```

End If
If RadioButton61.Checked = True Then
    acertado = acertado + 1
End If

Label19.Text = "su puntaje es: " + Str(puntaje) + " de 50"
Label17.Text = "su acertacion es: " + Str(acertado) + " de 10"
Label19.Visible = True
Label17.Visible = True
Label110.Visible = True
Label18.Visible = True

Button4.Enabled = False
Button3.Enabled = True
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    'nuevo intento'
    Label19.Visible = False
    Label17.Visible = False
    Label110.Visible = False
    Label18.Visible = False

    RadioButton35.Checked = False
    RadioButton36.Checked = False
    RadioButton34.Checked = False
    RadioButton38.Checked = False
    RadioButton39.Checked = False
    RadioButton37.Checked = False
    RadioButton41.Checked = False
    RadioButton42.Checked = False
    RadioButton40.Checked = False
    RadioButton47.Checked = False
    RadioButton48.Checked = False
    RadioButton46.Checked = False
    RadioButton50.Checked = False
    RadioButton51.Checked = False
    RadioButton49.Checked = False
    RadioButton53.Checked = False
    RadioButton54.Checked = False
    RadioButton52.Checked = False
    RadioButton59.Checked = False
    RadioButton60.Checked = False
    RadioButton58.Checked = False
    RadioButton62.Checked = False
    RadioButton63.Checked = False
    RadioButton61.Checked = False
    RadioButton44.Checked = False
    RadioButton45.Checked = False
    RadioButton43.Checked = False
    RadioButton56.Checked = False
    RadioButton57.Checked = False
    RadioButton55.Checked = False

    Button4.Enabled = True
    Button3.Enabled = False
End Sub

```

```

'subpestaña determinantes'
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
    'ver resultado'
    puntaje = 0
    acertado = 0
    'puntaje'
    If RadioButton64.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton69.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton71.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton75.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton78.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton80.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton82.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton87.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton88.Checked = True Then
        puntaje = puntaje + 5
    End If

    'repuestas acertadas'
    If RadioButton64.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton69.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton71.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton75.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton78.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton80.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton82.Checked = True Then
        acertado = acertado + 1
    End If
    If RadioButton87.Checked = True Then

```

```

        acertado = acertado + 1
    End If
    If RadioButton88.Checked = True Then
        acertado = acertado + 1
    End If

    Label14.Text = "su puntaje es: " + Str(puntaje) + " de 45"
    Label12.Text = "su acertacion es: " + Str(acertado) + " de 9"
    Label14.Visible = True
    Label12.Visible = True
    Label15.Visible = True
    Label13.Visible = True

    Button6.Enabled = False
    Button5.Enabled = True
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    'nuevo intento'
    Label14.Visible = False
    Label12.Visible = False
    Label15.Visible = False
    Label13.Visible = False

    RadioButton65.Checked = False
    RadioButton66.Checked = False
    RadioButton64.Checked = False
    RadioButton67.Checked = False
    RadioButton68.Checked = False
    RadioButton69.Checked = False
    RadioButton70.Checked = False
    RadioButton71.Checked = False
    RadioButton72.Checked = False
    RadioButton73.Checked = False
    RadioButton74.Checked = False
    RadioButton75.Checked = False
    RadioButton76.Checked = False
    RadioButton77.Checked = False
    RadioButton78.Checked = False
    RadioButton79.Checked = False
    RadioButton80.Checked = False
    RadioButton81.Checked = False
    RadioButton82.Checked = False
    RadioButton83.Checked = False
    RadioButton84.Checked = False
    RadioButton85.Checked = False
    RadioButton86.Checked = False
    RadioButton87.Checked = False
    RadioButton88.Checked = False
    RadioButton89.Checked = False
    RadioButton90.Checked = False

    Button6.Enabled = True
    Button5.Enabled = False
End Sub

```

```

'subpestaña puntos de corte'
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button8.Click
    'ver resultado'
    puntaje = 0
    acertado = 0
    'puntaje'
    If RadioButton92.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton94.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton99.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton101.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton105.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton106.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton109.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton113.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton117.Checked = True Then
        puntaje = puntaje + 5
    End If
    If RadioButton118.Checked = True Then
        puntaje = puntaje + 5
    End If

    'respuestas acertadas'
    If RadioButton92.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton94.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton99.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton101.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton105.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton106.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton109.Checked = True, Then

```

```

        acertado = acertado + 1
    End If
    If RadioButton113.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton117.Checked = True, Then
        acertado = acertado + 1
    End If
    If RadioButton118.Checked = True, Then
        acertado = acertado + 1
    End If

    Label19.Text = "su puntaje es: " + Str(puntaje) + " de 50"
    Label17.Text = "su acertacion es: " + Str(acertado) + " de 10"
    Label19.Visible = True
    Label17.Visible = True
    Label20.Visible = True
    Label18.Visible = True

    Button8.Enabled = False
    Button7.Enabled = True
End Sub

```

```

Private Sub Button7_Click (ByVal sender as System.Object, ByVal e as
System.EventArgs) Handles Button7.Click

```

```

    'nuevo intento'
    Label19.Visible = False
    Label17.Visible = False
    Label20.Visible = False
    Label18.Visible = False

    RadioButton91.Checked = False
    RadioButton92.Checked = False
    RadioButton93.Checked = False
    RadioButton94.Checked = False
    RadioButton95.Checked = False
    RadioButton96.Checked = False
    RadioButton97.Checked = False
    RadioButton98.Checked = False
    RadioButton99.Checked = False
    RadioButton100.Checked = False
    RadioButton101.Checked = False
    RadioButton102.Checked = False
    RadioButton103.Checked = False
    RadioButton104.Checked = False
    RadioButton105.Checked = False
    RadioButton106.Checked = False
    RadioButton107.Checked = False
    RadioButton108.Checked = False
    RadioButton109.Checked = False
    RadioButton110.Checked = False
    RadioButton111.Checked = False
    RadioButton112.Checked = False
    RadioButton113.Checked = False
    RadioButton114.Checked = False
    RadioButton115.Checked = False
    RadioButton116.Checked = False
    RadioButton117.Checked = False

```

```

RadioButton118.Checked = False
RadioButton119.Checked = False
RadioButton120.Checked = False

Button8.Enabled = True
Button7.Enabled = False
End Sub

```

14.3 Compilación, instalación y ejecución del software (ALGMATH)

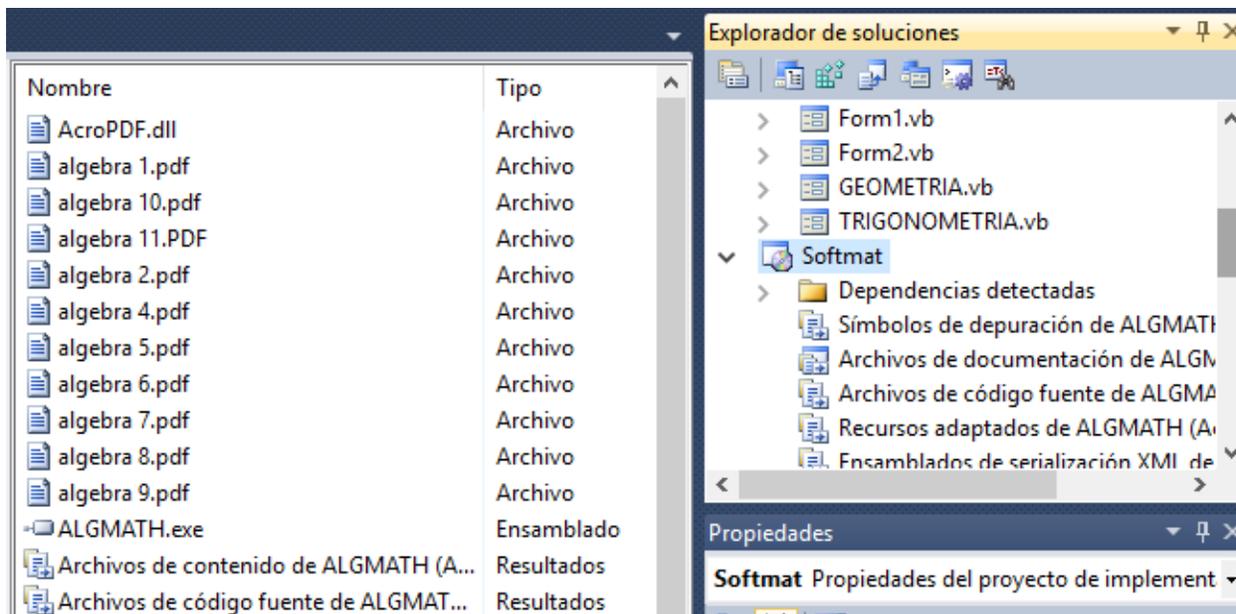


Figura 134: compilación del programa desarrollado. Mueses, J. (2017). Fuente: autoría propia

Instalación del ejecutable.

Nota: el programa como tal es ALGMATH, y el instalador se le puso el nombre de Softmat, esto porque no se puede colocar dos proyectos con el mismo nombre.

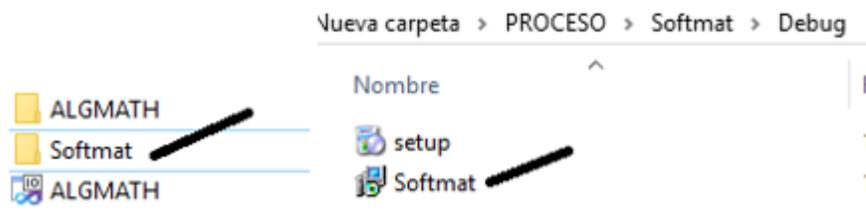


Figura 135: instalación del programa. Mueses, J. (2017). Fuente: autoría propia



Figura 136: proceso de instalación de ALGMATH. Mueses, J. (2017). Fuente: autoría propia

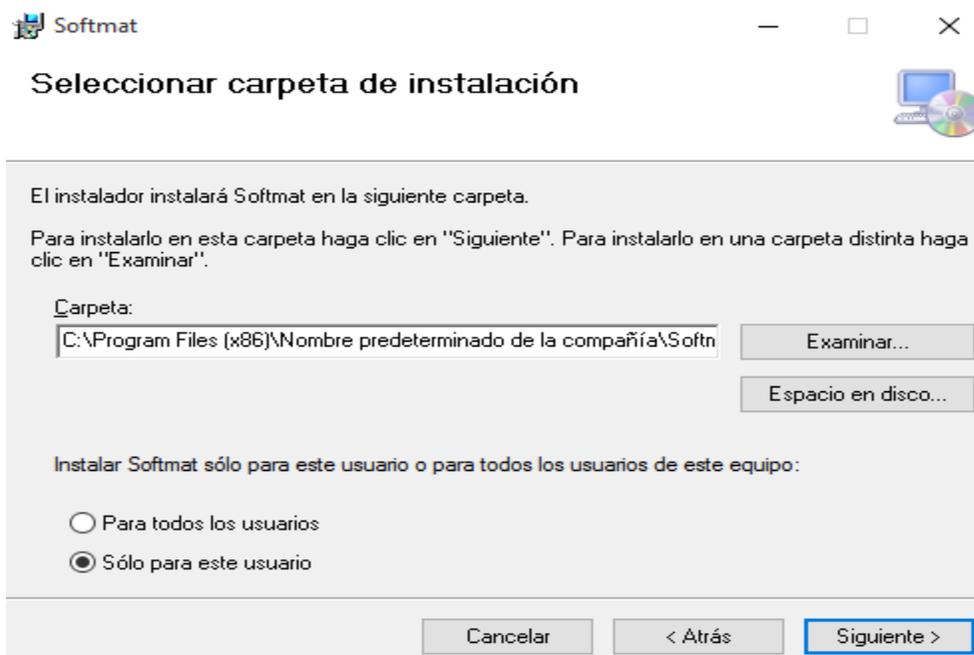


Figura 137: lugar donde se instalará el programa. Mueses, J. (2017). Fuente: autoría propia

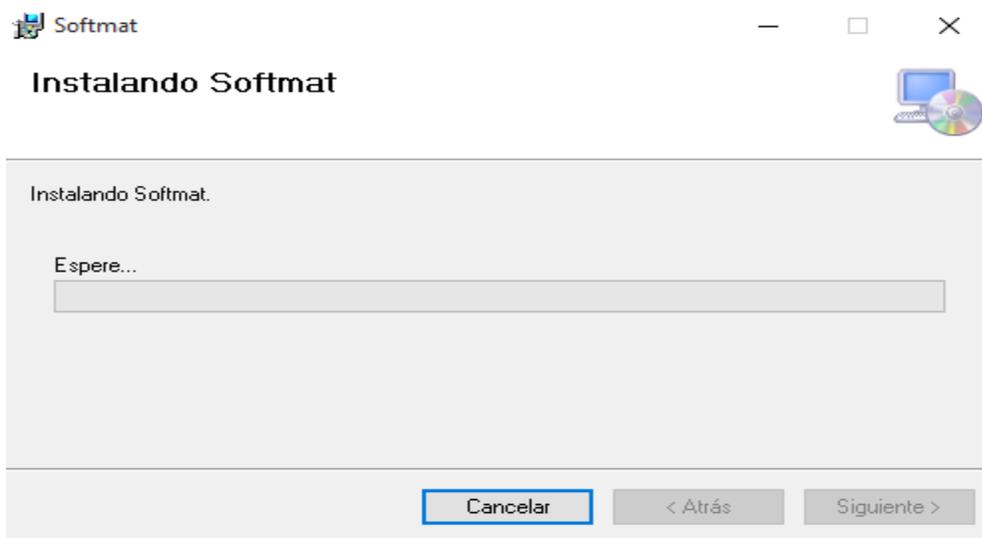


Figura 138: iniciándose la instalación del programa. Mueses, J. (2017). Fuente: autoría propia

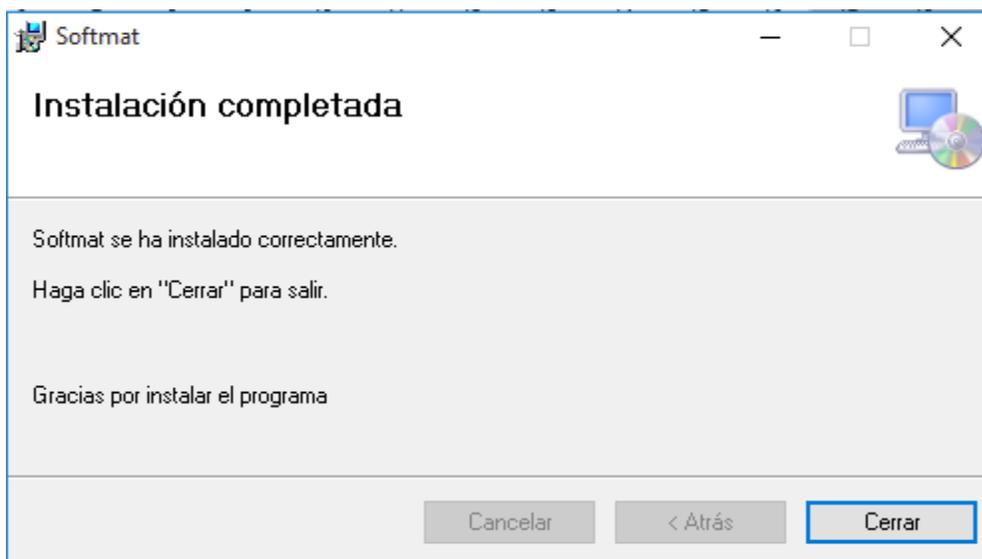


Figura 139: instalación completa del software de manera exitosa. Mueses, J. (2017). Fuente: autoría propia

Abrimos ALGMATH y realizamos cálculos.

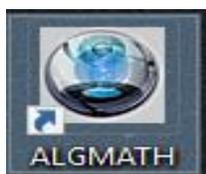


Figura 140: icono del programa ALGMATH como acceso directo en el escritorio. Mueses, J. (2017). Fuente: autoría propia



Figura 141: ejecutando el programa. Mueses, J. (2017). Fuente: autoría propia



Figura 142: interfaz principal. Mueses, J. (2017). Fuente: autoría propia

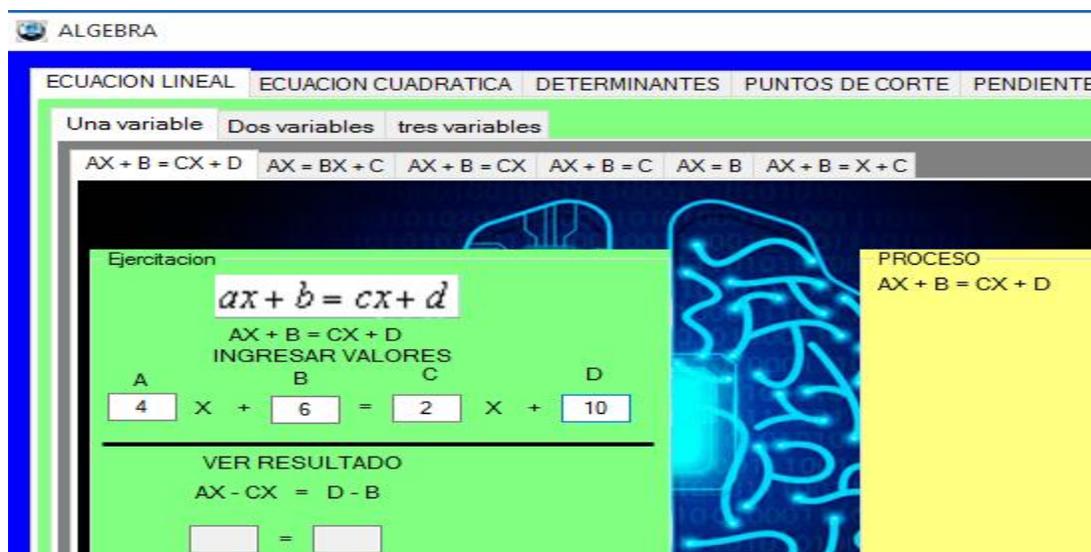


Figura 143: ingresando valores (en la interfaz ecuacion lineal). Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECUACION DE UNA RECTA

Una variable Dos variables tres variables

AX + B = CX + D AX = BX + C AX + B = CX AX + B = C AX = B AX + B = X + C

Ejercitacion

$ax + b = cx + d$

AX + B = CX + D
INGRESAR VALORES

A	B	C	D
4	6	2	10

X + 6 = 2X + 10

VER RESULTADO

AX - CX = D - B

2 = 4

4 D - B

X = $\frac{4}{2}$ A - C

X = 2

HALLAR

NUEVO

LA ECUACION TIENE SOLUCION

PROCESO

AX + B = CX + D

$4x + 6 = 2x + 10$

$4x - 2x = 10 - 6$

$2x = 4$

$x = 4/2$

$x = 2$

Prueba

$4x + 6 = 2x + 10$

$(4 * (4/2)) + 6 = (2 * (4/2)) + 10$

$(16/2) + 6 = (8/2) + 10$

$(16 + (2 * 6))/2 = (8 + (2 * 10))/2$

$28/2 = 28/2$

14 = 14

PROCESO

Desarrollado por Jonny Mueses

Figura 144: muestra del resultado y proceso generado por el programa. Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA ECU.

2 x 2 con numeros 2 x 2 con variables 3 x 3 con numeros 3 x 3 con variables

EJERCITACION

DETERMINANTE 2 X 2

HALLAR LA DETERMINANTE
USANDO LA REGLA DE CRAMER

INGRESE VALORES

4	6
3	9

$\begin{bmatrix} m & n \\ p & q \end{bmatrix}$

$\det = (m * q) - (p * n)$

VER RESULTADO

HALLAR

NUEVO

Figura 145: ingresando valores en la interfaz (determinantes). Mueses, J. (2017). Fuente: autoría propia

ALGEBRA

ECUACION LINEAL ECUACION CUADRATICA DETERMINANTES PUNTOS DE CORTE PENDIENTE DE LA RECTA

2 x 2 con numeros 2 x 2 con variables 3 x 3 con numeros 3 x 3 con variables

EJERCITACION
DETERMINANTE 2 X 2
HALLAR LA DETERMINANTE
USANDO LA REGLA DE CRAMER

INGRESE VALORES

4	6
3	9

$$\begin{bmatrix} m & n \\ p & q \end{bmatrix}$$

$$\det = (m * q) - (p * n)$$

VER RESULTADO HALLAR NUEVO

usamos la regla de cramer

DET = (4 * 9) - (3 * 6)

DET = 36 - 18

DET = 18

Figura 146: resultado generado por el programa. Mueses, J. (2017). Fuente: autoría propia

Acerca de ALGMATH

ALGMATH

Versión 1.0.0.0

Copyright © 2017

Autor: JonnyMueses

El software Algmath es un programa de matematicas de codigo abierto y de uso libre, usado para realizar calculos y para consultas basicas

Aceptar

Figura 147: interfaz acerca de. Mueses, J. (2017). Fuente: autoría propia

Al realizar la instalación y ejecutar el software como tal, se evaluó cada interfaz, se ingresaron datos y se vieron los resultados, los cuales eran correctos.

14.4 Instalación del software en los computadores del colegio.

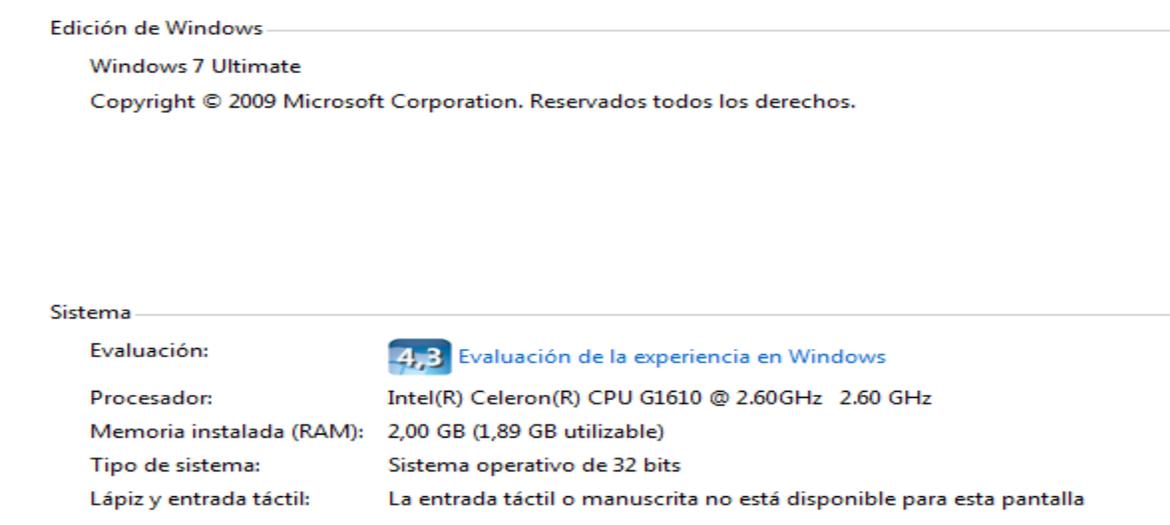


Figura 148: propiedades del sistema operativo Windows 7 Ultimate. Mueses, J. (2017). Fuente: autoría propia



Figura 149: inicio de instalación del programa. Mueses, J. (2017). Fuente: autoría propia



Figura 150: ubicación de instalación del programa. Mueses, J. (2017). Fuente: autoría propia



Figura 151: proceso de instalación del software (ALGMATH). Mueses, J. (2017). Fuente: autoría propia



Figura 152: programa instalado en el sistema, icono del software (ALGMATH). Mueses, J. (2017). Fuente: autoría propia



Figura 153: iniciando el programa instalado en el computador del colegio. Mueses, J. (2017). Fuente: autoría propia

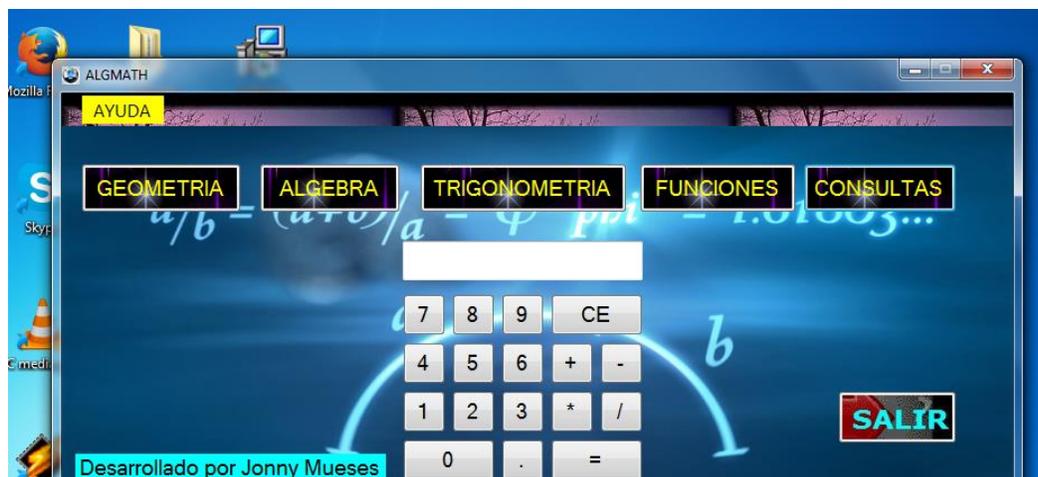


Figura 154: interfaz principal del software (ALGMATH). Mueses, J. (2017). Fuente: autoría propia

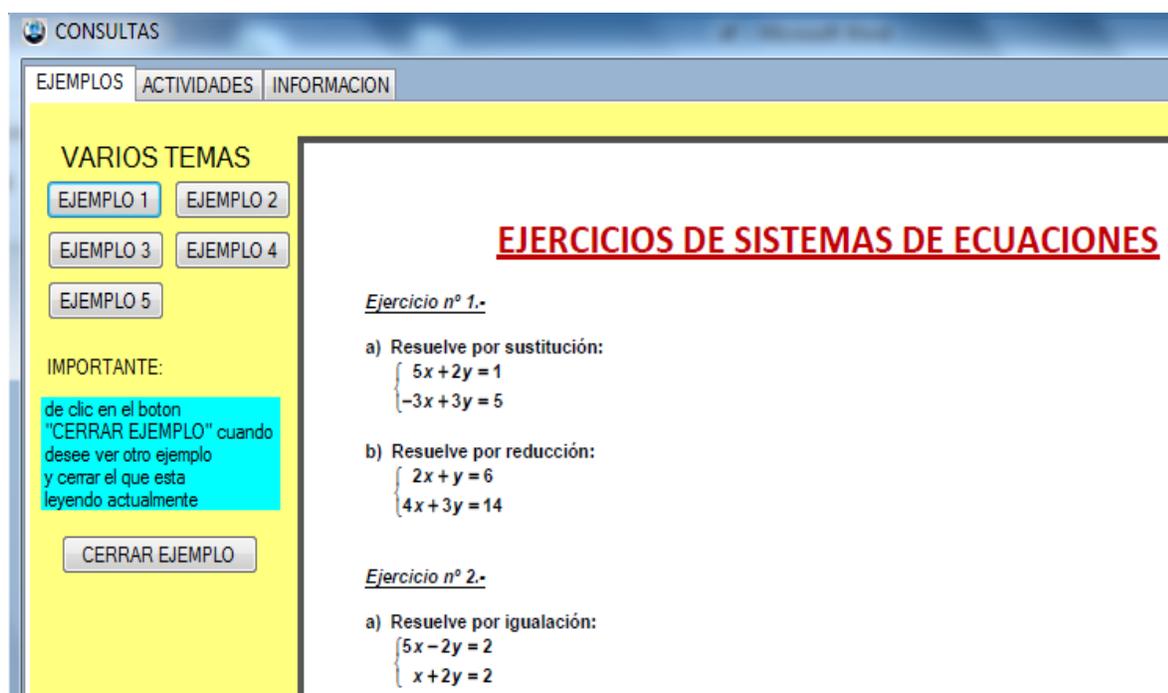


Figura 155: interfaz (ejemplos), muestra de un texto en formato pdf. Mueses, J. (2017). Fuente: autoría propia