

Universidad de las Ciencias Informáticas

Facultad 7



Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas.

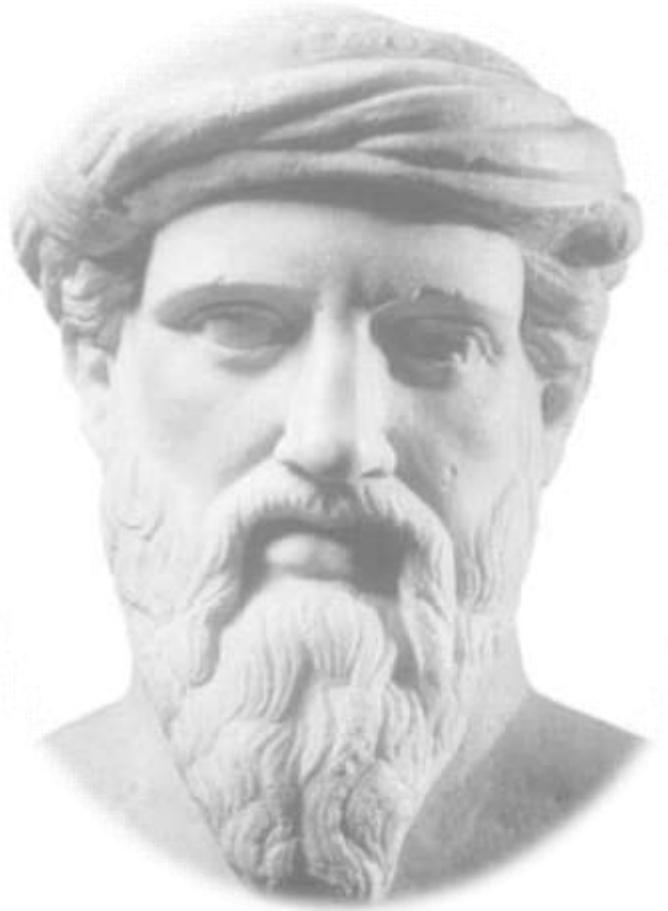
Desarrollo de una aplicación informática para el desarrollo de habilidades y hábitos en la representación gráfica de funciones lineales y cuadráticas.

Autores: Elizandra Sánchez Marrero.
Dunieska Blanco Agramonte.

Tutores: MSc. César N. Richard Martínez.
Ing. Alexander Rodríguez Rabelo.

La Habana, 13 de junio de 2013

“Año 55 de la Revolución.”



“Educad a los niños y no será necesario castigar a los hombres.”

Pythagoras de Samos.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

Desarrollo de una aplicación informática para el desarrollo de habilidades y hábitos en la representación gráfica de funciones lineales y cuadráticas y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los **6** días del mes de **junio** del año **2013**.

Dunieska Blanco Agramonte

Autor

Elizandra Sánchez Marrero

Autor

MSc. César R. Nicolás Martínez

Tutor

Ing. Alexander Rodríguez Rabelo

Tutor

DEDICATORIA

Les dedico este trabajo a las personas más importantes de mi vida:

A mi mamita linda: gracias por darme la vida, por estar siempre apoyándome en todo. Por inculcarme la ternura, el amor y el deseo inmenso de verme triunfadora, que sepa dar afecto, amor, regalar caricias, abrazos y sobre todo saber apreciar las cosas que nos da la vida. Porque cuando necesito una amiga, cuando necesito cariño o consejos tengo la suerte de contar contigo Tu fuerza y tu amor me guían.

A mi papito: por brindarme siempre su amor incondicional, cariño y comprensión, por estar siempre a mi lado. Por ser la persona más especial de mi vida. Por enseñarme a enfrentar mis errores con la frente siempre alta y aprender de ellos.

A tata: gracias por todos los momentos que hemos compartido y brindarme siempre tu amor. Por ser una ejemplo a seguir y por dedicarme tiempo. Por ser mi amiga y confidente.

A mi abuelita: por ser mi otra madre y brindarme su inmenso amor en todo momento. Gracias por tu dedicación y apoyo constante.

Elizandra Sánchez Marrero.

Dedico este trabajo a las personas que más quiero en mi vida:

A mis padres: y les pido vean en este trabajo el resultado de muchos años de sacrificio, cuidándome y guiándome. Siendo mi luz, mi apoyo.

A sobrinos, Jaddielito y Dianelis: que les sirva de motor impulsor el ejemplo que les estoy entregando.

A mis hermanos: Tintiri y Dunio, ahí está mis hermanos nuestro logro el mío que es también el suyo. Los quiero mucho.

A Pio: esto es para ti, por creer en mí. Por darme fuerzas cuando estaba cansada, por darme amor cuando parecía que todo era oscuro.

A todos los jóvenes que como yo se enfrentan al de cursar diario de la vida. Pasando retos cada día más complicados, a ustedes les digo que sigan para adelante que siempre se puede más. Solo se necesita deseos y convicción.

Dunieska Blanco Agramonte

AGRADECIMIENTOS

A todas aquellas personas que han hecho posible este día: mis padres Evelin y Enrique por apoyarme en todo momento a lo largo de mi vida, guiarme hacia el camino correcto, ofrecerme amor, educación, confianza y dedicación. Sin ellos, no hubiese podido alcanzar esta meta, mi hermana Elizabeth por su constante apoyo y dedicación, a mis abuelitas que siempre me han brindado mucho amor, a mi familia que siempre me ha impulsado a ser una mejor persona cada día en especial a mis primos Ernesto y Erick y mi tío Berto, a mi hermano Andrés por ayudarme en todo momento y permitirnos formar parte de su familia, a mi compañera de tesis Dunieska por su gran ayuda y brindarme su amistad, mis compañeros de aula por compartir estos inolvidables cinco años juntos y ayudarme cuando lo necesitaba y en especial a mis grandes amigas Laura e Ismary, a Diego, Pedro, Ariel, Ernesto, a Alain por su gran ayuda y dedicación. A las personas que han contribuido de alguna manera en mi formación profesional a través de sus enseñanzas y lecciones, a Leandro Tase y Maikel Blanco, a mis tutores César Nicolás y Alexander Rodríguez, a todas gracias por guiarme siempre por el buen camino. A la Revolución por darme la oportunidad de formarme como ingeniera.

Elizandra Sánchez Marrero.

AGRADECER... para mí esas son palabras muy grandes, pues nunca terminaré de agradecerles realmente a todas las personas que nombraré en este escrito o que simplemente llevo muy dentro a la hora que lo escribo. Para agradecerles deberían estar naciendo una vez más y yo justo a su lado para retribuirle la ayuda y el cariño que me han dado a mí. Y aunque no les pueda agradecer como se los merecen quiero que sepan que si estuviera en mis manos haría con ustedes lo mismo y mucho más, primero que todo quiero agradecer a las personas más importantes de mi vida, a los que me trajeron a este mundo, a los que me inculcaron el amor por el estudio, el amor por la vida, el amor por la familia, a los que me pusieron donde estoy parada justamente hoy. A mis padres Nancy Agramonte y Melquiades (Papi) Blanco. A mis hermanos por estar ahí siempre, cuidándome y guiándome a cada paso que doy: Duniesky y Tiliu. A mis amigos de toda la vida, los de aquí y los de allá: Eileen, Lilian yuya, Rosali, Raidel, Dory, Pedro, Ernesto, a mi peli Denis Bárbaro, al gordo Maikel Blanco, Lorenzo, al insoponible de Diego y a su novia, mi mejor amiga #1 Laura. Agradezco a personas que aunque hoy no están aquí hicieron mucho por mí y me ayudaron a llegar hoy hasta aquí, personas que cuando yo decía es muy complicado, me decían " métete que eso es muy fácil", donde quiera que estén Roberto, Soto yeber, David muchas gracias. Agradezco a mi fiel compañera de tesis, esa que aunque parezca muy noble es tremenda mandarína, Eli muchas gracias por ser mi compañera, la mejor que pude tener. Agradezco a todos los profesores que me inculcaron valores y conocimientos. A mi prima Yaima que incontables veces me ha tirado un salve

ayudándome cuando más lo he necesitado. A mi querida tía Prieta por ser una madre más en mi vida, una amiga fiel, una compañera devota y a su esposo Pedro. A Mily, Sandra, Alina, a Amarilis, Papiro, Lachi, Mariesther, a Miguel Ángel que me ha salvado el día en más de una ocasión y aún lo sigue haciendo. A Pio por ser la persona que más me ayudó a estar aquí, con su apoyo, con sus sacrificios y con su cariño. A todos los que estén aquí y a los que llevo en mi corazón. A mis tutores, por su guía. A mi suegra y su familia habanera por estar ahí de tienda en tienda y de calle en calle y por haber traído al mundo a una persona increíblemente maravillosa que me ha apoyado en todos los pasos de esta tesis. A ti mi querido novio "muchas, muchas gracias".

A todas muchas gracias desde lo más profundo de mí ser.

Muñe.

RESUMEN

El uso actual de la informática influye considerablemente en la vida diaria de un gran número de personas, pues está diseñada para automatizar y apoyar todas las áreas de la sociedad. En la educación su uso ha traído importantes cambios que favorecen el proceso de enseñanza aprendizaje. Ha proporcionado disímiles programas que apoyan todas sus disciplinas incluyendo la matemática.

Los denominados asistentes matemáticos son desarrollados y utilizados en todos los lugares donde surjan la necesidad o la idea de crear nuevas alternativas, que faciliten la formación desde el punto de vista matemático de la nueva generación. En el departamento de Ciencias Básicas de la Facultad 7 de la UCI, luego de realizar un estudio sobre los métodos y medios que se utilizan en el proceso de enseñanza de la asignatura Matemática I, específicamente en los temas concernientes a la consolidación de conocimientos y habilidades para la representación gráfica de funciones lineales y cuadráticas, se determinó la necesidad de desarrollar un sistema capaz de apoyar dicho proceso.

El sistema creado tuvo como base los requisitos obtenidos a partir de la observación de funcionalidades presentes en otros sistemas, con características similares a las perseguidas por la investigación. Para el diseño e implementación del mismo se utilizaron una serie de técnicas, tecnologías, lenguajes, plataformas, metodologías y herramientas que fueron seleccionadas luego de determinar que su uso facilitarían el desarrollo de la solución. La aplicación desarrollada es el resultado del uso de una arquitectura en tres capas y orientado a objetos.

Palabras clave: asistente matemático, funciones cuadráticas, funciones lineales, matemática, representación gráfica

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN.....	5
1.1. Términos y definiciones.....	5
1.1.1. <i>Aprendizaje de la matemática.</i>	5
1.1.2. <i>Funciones matemáticas.</i>	6
1.1.3. <i>TICs.</i>	6
1.1.4. <i>Asistentes matemáticos.</i>	7
1.1.5. <i>Interactividad.</i>	7
1.1.6. <i>Gráficas computarizadas.</i>	8
1.1.7. <i>Graficadores.</i>	8
1.2. Estado del arte.....	9
1.2.1. <i>Aplicaciones internacionales.</i>	9
1.2.2. <i>Aplicaciones nacionales.</i>	12
1.2.3. <i>Necesidad de la investigación.</i>	12
1.3. Lenguajes y notación de modelado.....	13
1.3.1. <i>Lenguaje unificado de modelado.</i>	13
1.3.2. <i>Notación para modelar procesos de negocio.</i>	14
1.3.3. <i>C# 4.0.</i>	14
1.4. Metodologías.....	15
1.4.1. <i>Proceso Unificado de Desarrollo.</i>	15
1.5. Herramientas.....	16
1.5.1. <i>Enterprise Architect.</i>	16
1.5.2. <i>Microsoft Visual Studio.</i>	17
1.6. Tecnologías.....	17
1.6.1. <i>Framework .Net 4.5.</i>	17
1.7. Conclusiones del capítulo.....	18
CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	19
2.1. Representación gráfica de funciones.....	19
2.1.1. <i>Gráfica.</i>	19
2.1.2. <i>Representación gráfica de funciones.</i>	19
2.2. Sistema propuesto.....	19
2.3. Modelo de dominio.....	20
2.4. Especificación de los requisitos de software.....	21
2.4.1. <i>Requisitos funcionales.</i>	22
2.4.2. <i>Requisitos no funcionales.</i>	24
2.5. Definición de los actores del sistema.....	28
2.6. Diagrama de casos de uso del sistema.....	28

2.7.	Descripción de los casos de uso del sistema.	30
2.8.	Propuesta de casos de uso por ciclos de desarrollo.	35
2.9.	Conclusiones del capítulo.	37
CAPÍTULO 3.	CONSTRUCCIÓN DE LA SOLUCIÓN.	38
3.1.	Modelo arquitectónico.	38
3.2.	Diseño.	41
3.3.	Implementación.	46
3.3.1.	<i>Diagrama de Componentes de GFLC.</i>	46
3.3.2.	<i>Estándares para la codificación.</i>	47
3.3.3.	<i>Diagrama de clases persistentes.</i>	48
3.4.	Conclusiones del capítulo.	49
CONCLUSIONES.	50
RECOMENDACIONES.	51
REFERENCIAS BIBLIOGRÁFICAS.	52
BIBLIOGRAFÍA.	58
ANEXOS.	65
Anexo 1	Prefijos de los requerimientos no funcionales por categoría.	65
Anexo 2	Descripción de los casos de uso.	66
Anexo 3	Diagramas de clases del diseño.	81
Anexo 4	Diagramas de secuencia de los casos de uso arquitectónicamente significativos.	91
Anexo 5	Descripción de las clases del sistema.	106
Anexo 6	Aval de aprobación del cliente.	114

ÍNDICE DE TABLAS

Tabla 1 Resumen del estado del arte.....	13
Tabla 2 Requisitos funcionales.	23
Tabla 3 Requisitos no funcionales.....	26
Tabla 4 Actor del sistema.....	28
Tabla 5 Resumen del CU #1.1 Graficar rectas y parábolas.....	31
Tabla 6 Resumen del CU #1.2Mostrar contenidos teóricos de las ecuaciones lineales y cuadráticas.	31
Tabla 7 Resumen del CU #1.3 Representar rectas y parábolas en el plano.	31
Tabla 8 Resumen del CU #1.4 Editar configuración.	32
Tabla 9 Resumen del CU #1.5 Eliminar.	32
Tabla 10 Resumen del CU #1.6 Guardar.	32
Tabla 11 Resumen del CU #1.7Revisar.	33
Tabla 12 Resumen del CU#1.8 Evaluar habilidades del estudiante para la representación gráfica de funciones.	33
Tabla 13 Resumen del CU #1.9 Abrir ficheros.	34
Tabla 14 Resumen del CU #1.10 Modificar escala del eje coordenado.....	34
Tabla 15 Resumen del CU #1.11 Mover sistema coordenado.....	34
Tabla 16 Resumen del CU #1.12 Mostrar datos de la gráfica.	35
Tabla 17 Resumen del CU#1.13Insertar ecuación.	35
Tabla 18 Casos de uso propuestos para el primer ciclo de desarrollo.....	36
Tabla 19 Casos de uso propuestos para el segundo ciclo de desarrollo.	36
Tabla 20 Casos de uso propuestos para el tercer ciclo de desarrollo.....	37

ÍNDICE DE FIGURAS

Figura # 1 Modelo de dominio.....	21
Figura # 2 Diagrama de requisitos funcionales.....	24
Figura # 3 Diagrama de requisitos no funcionales.....	27
Figura # 4 Modelo de casos de uso del sistema.....	29
Figura # 5 Diagrama de casos de uso del sistema.....	30
Figura # 6 Diagrama de la arquitectura en tres capas.....	39
Figura # 7 Evidencia del patrón Experto en GFLC.....	40
Figura # 8 Evidencia del patrón Creador en GFLC.....	40
Figura # 9 Evidencia del patrón Bajo acoplamiento en GFLC.....	41
Figura # 10 Diagrama de clases del diseño del CU #3 Representar rectas y parábolas en el plano.....	42
Figura # 11 Diagrama de clases del diseño del CU #7 Revisar.....	43
Figura # 12 Diagrama de secuencia del CU #3 Representar rectas y parábolas en el plano.....	44
Figura # 13 Diagrama de secuencia del CU #7 Revisar.....	45
Figura # 14 Diagrama de componentes.....	47
Figura # 15 Diagrama de clases persistentes.....	48
Figura # 16 Modelo de datos.....	49

INTRODUCCIÓN.

En la actualidad, el uso de las tecnologías influye considerablemente en la vida diaria de un gran número de personas a lo largo de todo el planeta, pues la mayoría de las innovaciones que, como las computadoras, facilitan el estudio y el trabajo, están diseñadas para automatizar y apoyar todos los campos de la sociedad, díganse: administración pública, ocio, comunicación, ciencia, arte y la gestión empresarial entre otros (1). Su puesta en práctica se ha convertido en una necesidad, pues hoy ya no es posible hablar de avances sociales sin incluir el desarrollo de las tecnologías.

Dentro de ellas juega un papel decisivo el uso de la informática, ciencia aplicada que abarca el estudio y aplicación del procesamiento automático de la información, utilizando sistemas computacionales, generalmente implementados como aparatos electrónicos. También puede ser definida como la gestión automática de la información o como la ciencia que estudia la utilización y el desarrollo de ordenadores y software. Para el procesamiento de la información define técnicas, herramientas y metodologías (2).

Su puesta en práctica ha revolucionado todas las esferas sociales incluyendo el sistema educativo, el mismo no ha quedado al margen de los nuevos cambios, pues se encarga de atender la formación de los nuevos ciudadanos. La incorporación de las nuevas tecnologías debe hacerse con la perspectiva de favorecer los aprendizajes y facilitar los medios que sustenten el desarrollo de los conocimientos y de las competencias necesarias para la inserción social y profesional de los educandos.

En la educación su uso ha traído importantes cambios que favorecen el proceso de enseñanza aprendizaje, dentro de los diferentes niveles por los que transita el estudiante. Las mismas suscitan la colaboración entre los alumnos, les ayuda a centrarse en el aprendizaje, mejoran la motivación y el interés, favorecen el espíritu de búsqueda, promueven la integración y estimulan el desarrollo de ciertas habilidades intelectuales tales como el razonamiento, la resolución de problemas, la creatividad y la capacidad de aprender a aprender.

Es considerado un medio a través del cual puede acercarse la acción educativa a los estudiantes y representa una herramienta para propiciar la obtención de buenos resultados docentes, por lo que su uso dentro de la educación es definido como “Informática educativa” (3). La informática educativa fomenta la producción creativa en el alumno, motivándolo a adquirir nuevas estructuras cognitivas como producto de la resolución de necesidades reales (4).

Es la rama que se ha encargado de proporcionar disímiles programas que apoyan todas las áreas de la educación y que en la actualidad no se detiene, sino que continúa en la búsqueda de nuevas alternativas para apoyar los procesos educativos, por lo que existen actualmente en todo el mundo una amplia colección de software que responden al desarrollo de la informática educativa. Los software educativos son considerados programas

computacionales cuyas características estructurales y funcionales, apoyan y contribuyen a elevar la calidad del proceso de enseñar y aprender (5). Constituyen materiales de aprendizaje especialmente diseñados para ser utilizados con una computadora.

Entre estos software se incluyen los dirigidos a la disciplina matemática y son denominados asistentes matemáticos. Están dirigidos a contribuir con el desarrollo de dicha disciplina sirviendo como herramientas para agilizar cálculos y representaciones gráficas de funciones. Estos programas son desarrollados y utilizados en todos los lugares donde surjan la necesidad o la idea de crear nuevas alternativas, que faciliten la formación integral y continua desde el punto de vista matemático, a la nueva generación.

Cuba no se encuentra exenta de esta realidad y como parte de la batalla de ideas creó en el año 2002 la Universidad de Ciencias Informáticas (UCI), centro educacional que tiene entre sus objetivos formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, capaces de producir aplicaciones y servicios informáticos.

Dentro de la UCI y específicamente en el departamento de Ciencias Básicas de la facultad 7, los profesores hacen uso de los asistentes matemáticos para la enseñanza de esta disciplina, dotando a los estudiantes nuevas y variadas opciones que les permitan consolidar o desarrollar habilidades y hábitos en la representación gráfica de funciones lineales y cuadráticas a través de herramientas cada vez más completas y potentes.

Teniendo en cuenta los asistentes matemáticos a los que se tienen acceso en dicho departamento, existe la problemática de lograr que los estudiantes, sin la necesidad de que el profesor se encuentre de manera presencial, logren desarrollar dichas habilidades. Para esto es necesario un medio que le permita al estudiante participar en la construcción de dichas gráficas, evaluando sus resultados y que no se limite a mostrar las gráficas de las funciones dadas.

Teniendo en cuenta la situación antes mencionada se plantea el siguiente **problema a resolver**: ¿Cómo contribuir a la consolidación o desarrollo en los estudiantes de los conocimientos referentes a la representación gráfica de funciones lineales y cuadráticas? Para llevar un estudio de la problemática y poder ampliar la investigación se toma como **objeto de estudio**: los procesos de representación gráfica de funciones con asistentes matemáticos enmarcándose como **campo de acción**: los procesos de representación gráfica de funciones lineales y cuadráticas con asistentes matemáticos.

Para limitar el alcance de la investigación y darle solución al problema planteado se concreta como **objetivo general**: desarrollar un sistema que constituya una alternativa para la consolidación o desarrollo en los estudiantes de conocimientos referentes a la representación gráfica de funciones lineales y cuadráticas de manera autodidacta.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la investigación**:

- Realizar una caracterización de los sistemas informáticos que permiten al estudiante consolidar habilidades y hábitos en la representación gráfica de funciones lineales y cuadráticas existentes a nivel de universidad, nacional e internacional, estableciendo similitudes con la investigación en curso.
- Seleccionar, a partir de una caracterización, metodologías, herramientas, lenguajes y tecnologías existentes en la actualidad que faciliten el desarrollo de la solución.
- Generar los artefactos correspondientes a las fases definidas dentro de la metodología de desarrollo a utilizar.
- Implementar la aplicación aplicando las pautas de diseño y siguiendo lo establecido en las especificaciones de requisitos de software.

Estructuración del contenido.

El trabajo consta de una estructura en capítulos donde se pueden encontrar los detalles de la investigación, así como la propuesta y diseño del sistema:

Capítulo 1: Fundamentación teórica de la investigación. Contiene un estudio del estado del arte de los principales software didácticos que existen en el mundo y en Cuba, que permiten la representación gráfica de funciones lineales y cuadráticas, permitiéndole al estudiante formar parte del proceso de representación y que le emiten una calificación a medida que evolucione en el desarrollo del proceso. El análisis de las diferentes metodologías, herramientas, lenguajes y tecnologías que se pueden emplear en la solución y la selección de las más apropiadas para llevar a cabo el proceso con calidad a medida que avance la investigación.

Capítulo 2: Descripción de la solución propuesta. Como parte de la propuesta de solución, se describen las características del sistema y el modelo de dominio, elaborado con los principales conceptos del campo de acción y las relaciones que se establecen entre ellos. Se realiza la especificación de los requisitos funcionales y no funcionales del software, determinándose a su vez los casos de uso del sistema y sus descripciones, así como los actores y las relaciones entre ellos.

Capítulo 3: Construcción de la solución. En este capítulo se hace referencia al modelo de diseño e incluye los diagramas de clases del diseño, de secuencia y la descripción de las clases del diseño. Se seleccionarán a partir de la definición de la arquitectura los patrones de diseño y estándares de codificación a utilizar, con una breve descripción de sus características y el propósito de su uso en el sistema. Se incluye además lo relacionado con el flujo de trabajo de implementación, especificando los elementos del sistema y sus interacciones a través del

diagrama de componentes, así como sus descripciones. También se definirá el modelo de datos a partir de la generación del diagrama de clases persistentes.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN.

Introducción.

Durante el presente capítulo se realizará un análisis detallado sobre los diferentes elementos y características que nutren la base teórica-conceptual para la implementación de un sistema informático, que permita al estudiante consolidar sus habilidades y hábitos en la representación gráfica de funciones lineales y cuadráticas. Se muestran los principales elementos que caracterizan el devenir de los principales asistentes matemáticos que poseen funcionalidades iguales o similares a las que poseerá el sistema a desarrollar. Esta información se encontrará acompañada por el estado del arte de las tecnologías, metodologías y herramientas así como los lenguajes que serán utilizados durante la investigación y sobre las cuales se desarrollará la propuesta de solución.

1.1. Términos y definiciones.

Lo más importante cuando se va a iniciar el desarrollo de un software es comprender realmente el tema que se desea informatizar, es por eso que el análisis de los principales términos que se corresponden con el tema brindará a clientes y desarrolladores una visión más clara sobre lo que se modela. En el presente epígrafe se nombrarán y caracterizarán una serie de términos asociados a la matemática, a la representación gráfica de funciones y a las nuevas tecnologías entre otros que permitirán entender más claramente lo que se desea realizar.

1.1.1. Aprendizaje de la matemática.

La matemática es una ciencia que es enseñada y aprendida en el mundo por millones de personas, ejerce una influencia marcada en la sociedad, de manera que ha contribuido con su evolución cultural. Resulta un lenguaje que surgió por las necesidades de múltiples índoles como: la explicación de fenómenos naturales (aquellos procesos permanentes de movimientos y de transformaciones que sufre la naturaleza, ejemplos: terremotos, tsunamis, erupciones volcánicas, etc.) (6), la contabilidad de elementos (alimentos, familia, sociedad entre otros).

Su aprendizaje resulta ser un buen aliado para: el perfeccionamiento de capacidades cognitivas (de razonamiento, abstracción, inducción, deducción, reflexión, análisis), el tratamiento de actitudes, tales como la confianza de los alumnos en sus propios procedimientos y conclusiones, favoreciendo la autonomía de pensamientos, para aceptar que se pueden equivocar y que es necesario detectar y corregir los errores. Influye en el inicio al análisis de sus propias estrategias de reflexión, de diversidad de procedimientos y de nuevas ideas (7).

Permite formar adecuadamente el pensamiento analítico, el rigor demostrativo, el sentido de la exactitud y el de la aproximación aceptable, la objetividad numérica y la propensión a la medición (8). En aspirantes a profesionales

de la informática favorece el desarrollo de habilidades específicas de observación, análisis, síntesis, modelaje, planificación y toma de decisiones (9).

El aprendizaje de la matemática contribuye al desarrollo de habilidades comunicativas, que hacen más precisa y rigurosa la expresión de ideas y razonamientos, incorporando en el lenguaje y argumentaciones habituales, las diversas formas de expresión matemática (numérica, gráfica, simbólica, lógica, probabilística y estadística).

1.1.2. Funciones matemáticas.

La matemática es una ciencia asociada a distintas áreas del conocimiento, hoy en día se usan en todo el mundo como una herramienta esencial en muchos campos, entre los que se encuentran las ciencias naturales, la ingeniería, la medicina y las ciencias sociales, e incluso disciplinas que, aparentemente, no están vinculadas con ella, como la música (10). Por su naturaleza abstracta contempla una serie de contenidos que pueden ser vistos como complejos por la mayoría de las personas, a pesar que de alguna manera se relacionan cada día con conceptos matemáticos que son estudiados en los diferentes niveles educativos (11).

La representación gráfica de funciones lineales y cuadráticas constituye uno de los contenidos de la matemática que entre otros tanto son para muchas personas un trabajo complicado y hasta imposible. Dichas funciones matemáticas (lineales y cuadráticas) entran en este grupo ya que son conceptos centrales en el aprendizaje de las matemáticas.

La construcción del concepto de función es la base para que posteriormente el alumno pueda comprender otros conceptos matemáticos. Este proceso lógico se aplica a todo lo que tiene relación a un resultado o efecto, sea este medible o no en forma cuantitativa (12). De ahí que facilitar su aprendizaje ha sido una necesidad durante la evolución de las Tecnologías de la Información y las Comunicaciones (TICs). A partir de dicha necesidad surge la idea de crear sistemas informáticos capaces de realizar la representación gráfica de funciones, llamados asistentes matemáticos.

1.1.3. TICs.

Las TICs se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático (13).

Son parte de las tecnologías emergentes que habitualmente suelen identificarse con las siglas TIC y que hacen referencia a la utilización de medios informáticos para procesar y difundir todo tipo de información o procesos de formación educativa (14).

Son herramientas teóricos-conceptuales, soportes y canales que procesan, sintetizan, recuperan y presentan información de la forma más variada. El uso de las TIC representa una variación notable en la sociedad y a la

larga, un cambio en la educación, en las relaciones interpersonales y en la forma de difundir y generar conocimientos. Les facilitan a los estudiantes tener experiencias auténticas, acceder de inmediato a una gran fuente de materiales, contar con múltiples medios de visualización de problemas reales y de conexión de éstos con experiencias previas, proveen herramientas idóneas para comunicarse auténticamente con personas distantes en cualquier momento (15).

En el área de las matemáticas las TICs fomentan los siguientes beneficios:

- Facilitan el aprendizaje de materias y conceptos.
- Favorece la autonomía del estudiante al permitirles interactuar con objetos matemáticos de manera simple y natural.
- Facilitan y estimulan la comprensión de los principales cambios existentes en las tecnologías, la naturaleza y la sociedad a partir de la modelación y la observación computarizada de gráficos y patrones.

1.1.4. Asistentes matemáticos.

Los asistentes matemáticos constituyen herramientas computacionales que se pueden utilizar para la representación gráfica de funciones y dar solución a problemas vinculados a la matemática. Desde el punto de vista de la formación, el uso de un asistente matemático abre la atractiva posibilidad de experimentar con las matemáticas ya que son programas diseñados con intencionalidad pedagógica.

Son considerados recursos tecnológicos que permiten al estudiante concentrar esfuerzos en el razonar, solucionar y formular problemas, así como en verificar teoremas y propiedades matemáticas (16). Les posibilita además combinar datos de forma numérica y graficar con un sustancial ahorro de tiempo ya que minimizan los engorrosos cálculos (17).

La utilización de estas tecnologías informáticas favorece la simulación de fenómenos de la realidad, ayudan y motivan a un trabajo más creativo en el aula al ser utilizadas para formular conjeturas, buscar soluciones, explorar patrones. Permiten junto con los medios educativos tradicionales mejorar el aprendizaje (18).

1.1.5. Interactividad.

Como resultado de los increíbles avances de las TICs, hoy las personas forman parte de una época en que parece que todos los esfuerzos están dirigidos a hacer de la realidad una realidad interactiva y todo cuanto se muestra acompañado de este adjetivo se hace máspreciado, innovador y actual. Lo interactivo, refiriéndose a un programa informático, describe una interacción a modo de diálogo entre ordenador y usuario; o puede ser interpretado más allá de la simple navegación por una aplicación, como que el usuario tiene verdadero control sobre determinadas acciones (19).

El nivel de interactividad mide las posibilidades y el grado de libertad del usuario dentro del sistema, así como la capacidad de respuesta de este sistema en relación al usuario, en cuanto a cualidad y en cantidad; esta relación se podría poner en paralelo con el esquema de comunicación: emisor, receptor, respuesta. Si se incrementan las posibilidades de interactuar con los productos, siendo cada vez más grande el grado de libertad del usuario a la hora de tomar decisiones en relación a ¿Qué hacer? o ¿cómo hacerlo? y se consigue aumentar el grado de eficacia de la aplicación al obedecer las instrucciones que dé el usuario, se podría decir que se incrementa el nivel de interactividad.

Las herramientas interactivas constituyen un elemento importante que permite reflejar, bajo la abstracción de conceptos matemáticos, la intuición de los estudiantes (20). De esta forma un alumno puede trabajar sobre un problema de forma gráfica y observar cómo el cambio en un determinado elemento se ve reflejado de forma inmediata en el resto, como si estuviera ante el proceso real.

Los objetos interactivos son aquellos que responden a determinados eventos ante las acciones del usuario, por ejemplo los botones (16). Se dice que un software es interactivo cuando responde inmediatamente a las acciones de los clientes mediante un diálogo o intercambio de información.

1.1.6. Gráficas computarizadas.

La visualización de funciones para lograr un mejor entendimiento del proceso que se modela, constituye el objetivo principal de un asistente para la representación gráfica. La mayoría de los asistentes matemáticos brindan la posibilidad a los usuarios de realizar representaciones gráficas de los elementos que se procesan y en la mayoría de los casos dichas representaciones pueden ser editables.

El resultado de la representación gráfica computarizada se emplea con fines científicos, artísticos, etc. Han hecho que las interfaces gráficas de usuario (GUI, según sus siglas en inglés) y los sistemas para computadoras sean más fáciles de usar, les permiten a los usuarios de ordenadores seleccionar imágenes para ejecutar órdenes, lo que elimina la necesidad de memorizar instrucciones complicadas (21). Este proceso puede ser realizado únicamente por los clientes (estudiantes, profesores u otra persona) pero nunca por sistemas externos.

1.1.7. Graficadores.

En la enseñanza de la matemática, el uso de programas graficadores es una de las aplicaciones más prometedoras. Los software didácticos para computadora poseen la capacidad de interactuar con el usuario, facilitan el proceso de aprendizaje. Ofrecen la oportunidad al estudiante de realizar funciones con rapidez y de forma precisa. El uso de graficadores representa una alternativa para la enseñanza de la matemática por: su

versatilidad, fácil uso, abarca la mayoría de los objetivos y requiere un conocimiento mínimo sobre el uso del computador.

1.2. Estado del arte.

En este epígrafe se realiza un análisis sobre los principales asistentes matemáticos existentes actualmente a nivel de universidad, nacional e internacional, que entre sus funcionalidades incluyan:

- Resolver y representar gráficamente funciones lineales y cuadráticas.
- Brindar al usuario la posibilidad de representar gráficamente funciones lineales y cuadráticas de forma manual.
- Muestren al usuario una evaluación de la representación gráfica realizada.

1.2.1. Aplicaciones internacionales.

A lo largo y ancho del planeta existe una amplia colección de aplicaciones que permiten representar gráficamente funciones, entre los que se destacan los asistentes matemáticos internacionalmente conocidos y utilizados: MATLAB, Microsoft Mathematics y Derive. Asistentes matemáticos reconocidos por su amplia colección de funcionalidades.

MATLAB

MATLAB significa Matrix Laboratory, es un entorno interactivo para el cálculo numérico, visualización y programación que utiliza para su implementación un lenguaje de alto nivel con igual nombre. Fue creado por Cleve Moler en 1984, surgiendo la primera versión con la idea de emplear paquetes de subrutinas escritas en Fortran en los cursos de álgebra lineal y análisis numérico, sin necesidad de escribir programas en ese lenguaje. Actualmente es distribuido por MathWorks.

MATLAB cuenta con una gran diversidad de usos entre los que destacan: simular, modelar, crear prototipos, analizar datos y encontrar soluciones a sistemas complejos. Actualmente cuenta con cerca de 20 paquetes de herramientas para diferentes usos tales como: acústica, aeronáutica, astronomía, biotecnología y estadísticas, entre otros.

Usando MATLAB se puede analizar datos, desarrollar algoritmos, crear modelos y aplicaciones. El lenguaje, las herramientas y funciones incorporadas de matemáticas le permiten explorar múltiples enfoques y llegar a una solución más rápida que con las hojas de cálculo o lenguajes de programación tradicionales, tales como C / C ++ o Java (22).

Se encuentra disponible para las plataformas Unix, Windows y Mac OS X entre otras (23).

Le permite al usuario realizar cálculos numéricos con vectores y matrices. Puede también trabajar en casos particulares con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas que posee es la de realizar una amplia variedad de gráficos en dos y tres dimensiones (24).

Microsoft Mathematics

Herramienta educativa que agrupa en su entorno muchas de las funciones que cualquier calculadora científica tendría. Entre estas se encuentran: capacidad para representar funciones de distintos grados, calcular derivadas e integrales, resolver ejercicios de álgebra lineal, números complejos y ecuaciones paso a paso. Permite la conversión entre unidades y la representación gráfica de funciones en 2D y 3D.

Derive

Es un programa de álgebra computacional (CAS), surge como un sucesor de muMATH por Soft Warehouse en Honolulu, Hawaii, Estados Unidos (EE. UU) en 1988, diseñado para el cálculo matemático avanzado. Es desarrollado por Texas Instruments (25).

Procesa variables, expresiones algebraicas, ecuaciones, funciones, vectores, matrices, trigonometría, derivadas, integrales y expresiones booleanas, etc. (26). Posee capacidades de calculadora científica y puede representar funciones gráficas en dos y tres dimensiones en varios sistemas coordenados (27).

Se encuentra disponible para las plataformas Windows y DOS. Es usado ampliamente con propósitos educativos y es considerado un lenguaje de programación muy cercano al usuario o lenguaje de alto nivel, por las características peculiares que posee, entre las que destacan:

- La utilización por defecto de aritmética exacta, lo que le permite manipular expresiones racionales como $1/6$, sin necesidad de tener que operar con su expresión en coma flotante 0,1666666667.
- La manipulación de variables sin asignación o expresiones no numéricas y en consecuencia expresiones algebraicas, donde los datos no han de ser valores numéricos.
- El soporte de estructuras de datos de tipo vectorial y matricial (28).

Los anteriormente mencionados asistentes matemáticos son conocidos y utilizados, además gozan de la aceptación de un gran número de usuarios, son utilizados como material de apoyo en algunas universidades del mundo, pero ellos no cuentan con el privilegio de ser los únicos existentes a nivel internacional. Existe una amplia colección de no tan conocidos programas para ordenador que le ofrecen al cliente la posibilidad de graficar y resolver funciones matemáticas y cuyo principio de funcionamiento es el de agilizar el proceso de enseñanza aprendizaje. Entre ellos se incluyen los que se describen a continuación.

Winplot

Es un generador de funciones gráficas especialmente diseñado para el estudio visual de una serie de ecuaciones matemáticas. Se incluye en un conjunto de diferentes programas conocido con el nombre de "Peanut Software" desarrollado por Rick Parris. La versión inicial fue en 1985 y la más actualizada salió al mercado en el año 2009 (29).

Con Winplot se pueden generar gráficas de ecuaciones explícitas, paramétricas, implícitas y cilíndricas, curvas simples, tubos e incluso representar ecuaciones diferenciales tanto en dos como en tres ejes (2D y 3D). Permite observar el comportamiento de cada una de las funciones y la forma que van tomando según la variación de determinados valores y personalizar los parámetros de todas las ecuaciones (29).

Winplot le brinda al usuario la facilidad de modificar el valor de X, Y, Z; el número de divisiones, los puntos de corte y definir la calidad de la representación. Este graficador además de generar y representar funciones gráficamente incluye, entre sus funcionalidades adicionales, una serie de test que ayudarán a evaluar los conocimientos de los usuarios (30).

GRAFIFUN

Es una herramienta creada por Proyecto A y D SEC CHA (de Análisis y Desarrollo de un Sistema de Enlaces Conceptuales y Aplicativos de Contenidos como Herramienta de Apoyo al Alumno), realizado por un grupo de docentes del departamento de informática de la facultad de ingeniería de la Universidad Nacional de la Patagonia "San Juan Bosco" (31).

Surge para ayudar a los alumnos ingresantes a comprender y ejercitarse en el estudio de funciones lineales y cuadráticas, conocer las características más importantes de dichas funciones y verificar las gráficas realizadas por cada uno de ellos, de manera sencilla y a través de una interfaz gráfica intuitiva (32).

Octave

Es un asistente matemático previsto de similares funcionalidades que Matlab y con completa compatibilidad; sin embargo, dada la poca interactividad que posee y la carencia de una interfaz gráfica, muchas veces es renegado a la última de las opciones por estudiantes y profesionales.

Octave o GNU Octave es un programa libre para realizar cálculos numéricos. Como indica su nombre es parte del proyecto GNU. Apoyado en una amplia comunidad de desarrolladores y usuarios, Octave cuenta con herramientas para la resolución de problemas de cálculos numéricos lineales y no lineales: álgebra lineal, aproximación de raíces de ecuaciones, integración numérica, integración de ecuaciones diferenciales, etc., así como para la representación de gráficos en dos y tres dimensiones.

Es fácilmente extensible y adaptable mediante funciones definidas por el usuario, bien utilizando el propio lenguaje de Octave o bien mediante módulos escritos en C++, C, Fortran u otros lenguajes.

El proyecto fue creado alrededor del año 1988 con la finalidad de ser utilizado en un curso de diseño de reactores químicos. Posteriormente en el año 1992 se decide extenderlo y comienza su desarrollo hasta nuestros días (33).

1.2.2. Aplicaciones nacionales.

En Cuba existe una gran cantidad de software educativos que apoyan la docencia y en específico están dirigidos al campo de las matemáticas; aplicaciones en muchos casos de procedencia internacional y que, por poseer un código abierto, han sido modificadas o adaptadas a las necesidades propias de la educación cubana. Como parte de ese proceso surge en la UCI una versión del Sistema Octave anteriormente analizada y que sus autores llamaron: QTOCTAVIZ: módulo para la visualización científica integrado al asistente matemático OCTAVE.

QTOCTAVIZ

Es un representador gráfico interactivo para la visualización científica desde el ambiente de Octave que dispone de prestaciones para la representación de funciones en 2D y 3D, así como para la gestión completa de la información relativa a estas. Abstrae al usuario del tratamiento de las representaciones y solo se concentra en la interacción con un sistema de ventanas y botones.

Eureka

Es otro bien conocido asistente matemático, dicho programa es parte de la Colección Futuro, surgió en enero del año 2004, producido por los centros de desarrollo de software de los 16 institutos superiores pedagógicos del país y rectorados por el Departamento de Software Educativos (DSE) del Ministerio de Educación. Está dirigido a la enseñanza preuniversitaria y cuenta entre sus módulos con uno destinado a la simulación.

Simulador: es el nombre del módulo que permite representar gráficamente funciones matemáticas predefinidas dentro del mismo. El programa es capaz de interpretar los parámetros de las funciones para poder ser variados durante su simulación y así observar los cambios que se producen en su gráfico.

1.2.3. Necesidad de la investigación.

Los sistemas antes mencionados cuentan de elevadas prestaciones, pero entre sus funcionalidades no se encuentra la de permitir al usuario realizar de forma manual la consolidación de las habilidades propias. Los mismos fueron desarrollados con el objetivo de simplificar el proceso, obteniéndose como resultado, que el estudiante conozca cada vez menos de lo que ocurre durante la representación gráfica, centrando sus esfuerzos solamente en entender cómo funcionan estos programas informáticos.

Para dar constancias de los resultados anteriormente expuestos se muestra a continuación una tabla resumen, en la cual se reflejan las capacidades que poseen los sistemas analizados para cumplir con las pautas definidas en la presente investigación.

Sistemas analizados	Resolver y representar gráficamente funciones lineales y cuadráticas	Representar gráficamente funciones lineales y cuadráticas de forma manual	Evaluación de la representación gráfica realizada
Matlab	X		
Derive	X		
Winplot	X		
GRAFIFUN	X		
Octave	X		
QTOCTAVIZ	X		
Eureka	X		
Microsoft Mathematics	X		

Tabla 1 Resumen del estado del arte.

Partiendo de esta premisa el departamento de Ciencias Básicas de la Facultad 7 de la UCI decidió desarrollar un sistema, que constituya una alternativa para la consolidación o desarrollo en los estudiantes de conocimientos referentes a la representación gráfica de funciones lineales y cuadráticas de manera autodidacta.

Para la implementación del sistema que se propone, se realiza a continuación un estudio de las principales tecnologías, herramientas, lenguajes y metodologías que se utilizarán durante su confección.

1.3. Lenguajes y notación de modelado.

1.3.1. Lenguaje unificado de modelado.

El lenguaje unificado de modelado (UML, por sus siglas en inglés), es el lenguaje seleccionado por ser de propósito general, usado por un gran número de diseñadores, es uno de los más conocidos y utilizados en la actualidad (34). Es la notación que usa los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño (principalmente gráfico).

UML destaca como sus principales funciones las de especificar, construir, visualizar y documentar cada una de las partes que comprende el desarrollo de software. La primera permite especificar cuáles son las características de un sistema antes de su construcción. La segunda brinda la posibilidad de construir a partir de los modelos

especificados los sistemas diseñados. La tercera permite expresar de una forma gráfica un sistema de formas que otro pueda entender. Por último los propios elementos gráficos sirven como documentación del sistema desarrollado, sirviendo para su futura revisión (35).

UML es una especificación de la notación orientada a objetos. Su utilización es independiente del lenguaje de programación y de las características de los proyectos ya que ha sido diseñado para modelar cualquier tipo de proyecto, tanto informáticos, como de arquitectura o de cualquier otra rama (36).

UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales y concretos como: expresiones de lenguajes de programación y esquemas de componentes de software reutilizables.

1.3.2. Notación para modelar procesos de negocio.

La notación para modelar procesos de negocio (BPMN, por sus siglas en inglés) es una notación gráfica en un formato de flujo de trabajo (workflow) en su versión 1.1. Fue escogida por ser la notación más utilizada en el mundo del desarrollo de software, debido a la facilidad de entendimiento que brinda a los usuarios de negocio. BPMN está diseñado para cubrir varios tipos de modelado y permite la creación tanto de segmentos de proceso como de procesos de negocio de comienzo a fin y todo ello en diferentes niveles de representatividad.

Fue creada debido a una necesidad en el mercado de inventar un lenguaje común para describir los procesos y asignarlos a la lógica de ejecución (37). Es independiente de cualquier metodología de modelado de procesos. Brinda una técnica de modelamiento de flujos naturales y consistentes con la manera de pensar y actuar de los analistas de negocios. Es mantenida por Object Management Group (OMG), la misma compañía encargada de UML (38).

El principal objetivo de BPMN es proveer una notación estándar que sea de fácil lectura y entendible por parte de todos los involucrados e interesados del negocio. En síntesis, BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. Así se ve que para BPMN, su dominio específico son los procesos de negocio y, como muchos otros, su objetivo es acercar el lenguaje al dominio, haciendo que todos puedan entenderlo.

1.3.3. C# 4.0

Como lenguaje de programación ha sido escogido C# 4.0 ya que el mismo utiliza el paradigma de programación orientada a objetos basada en tipado fuerte. Este paradigma usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos (39).

Fue desarrollado por Microsoft en las primeras versiones de la plataforma .Net y aprobado como un lenguaje estándar por la Organización Internacional basada en membresías de estándares para la comunicación y la información (Ecma International) e ISO (40).

Desde el paradigma orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Además de estos principios básicos de la Programación Orientados a Objetos (POO), C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen los siguientes:

- Atributos: que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Propiedades: que actúan como descriptores de acceso para variables privadas, con el objetivo de lograr un nivel de expresividad superior cuando se busque encapsulamiento y lograr el principio de elección única que la POO promueve (41).

1.4. Metodologías.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el proceso de creación de productos software, en la que se van indicando paso a paso todas las actividades a realizar para lograr informatizar el proceso deseado, indicando qué personas deben participar en el desarrollo de las actividades y qué papel deben desempeñar. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (42).

En el mundo existen diversas y variadas metodologías, pero no todas se aplican a las necesidades requeridas, por lo que la metodología a utilizar fue escogida de acuerdo a las peculiaridades del proyecto a realizar, con el objetivo de minimizar al máximo los riesgos a los que se estará expuesto a la hora de desarrollar el software y así, obtener los resultados esperados con el menor costo y esfuerzo posible (43).

1.4.1. Proceso Unificado de Desarrollo.

La metodología a seguir durante el desarrollo del sistema es el proceso unificado de desarrollo (RUP, según sus siglas en inglés). Constituye el resultado de varios años de perfeccionamiento y uso práctico en el que se han unificado técnicas de desarrollo. Sus características fundamentales son (44):

- Guiado y manejado por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

Esta metodología está compuesta por una serie de flujos de trabajo que se mencionan y caracterizan a continuación en correspondencia con las aspiraciones que se tiene con la aplicación a desarrollar:

- Modelamiento del negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- Análisis y diseño: describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas, indica con precisión lo que se debe programar.
- Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Su desarrollo queda recogido en cuatro fases:

- Inicio: descripción del negocio y delimitación del proyecto, especificando sus alcances con la identificación de los casos de uso del sistema. Durante esta etapa se obtiene la visión del proyecto.
- Elaboración: define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la implican. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- Construcción: se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias liberaciones del producto que han pasado las pruebas y se ponen a consideración de un subconjunto de usuarios.
- Transición: El producto liberado ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

1.5. Herramientas.

Las herramientas que van a ser utilizadas para dar solución a la propuesta que se presentará al concluir las fases de la metodología que fue escogida son las que se enuncian a continuación.

1.5.1. Enterprise Architect.

El Enterprise Architect (EA) en su versión 7.5 será utilizado como herramienta de modelado ya que constituye la solución que permite modelar y gestionar información compleja, diseñar y visualizar el software o construir y desplegar diversos sistemas. Combina la potencia de la última especificación UML 2.1 y BPMN 1.1, con un alto rendimiento y una intuitiva interfaz de usuario. Abarca por completo el ciclo de vida del desarrollo del software, con herramientas que proporcionan una infraestructura enormemente competitiva en torno al modelaje de negocio, diseño de software, ingeniería de sistemas y gestión de requerimientos entre otros (46).

EA muestra una interfaz de usuario intuitiva y alto nivel de flexibilidad respecto a la documentación que será generada. Es basada en Windows pero también compatible con sistemas Linux (46).

Una característica a destacar sobre EA es la generación de código fuente e ingeniería inversa para muchos lenguajes, incluyendo C#, siendo precisamente este lenguaje seleccionado para la implementación de la aplicación. EA posee complementos que permiten su vinculación con Visual Studio .Net, elemento de gran importancia ya que este último es el entorno para el desarrollo elegido (47).

EA soporta la trazabilidad completa, elemento fundamental en la administración de requisitos. Otra cuestión primordial es el bajo costo de sus licencias en comparación al resto de las herramientas utilizadas en la UCI, como Visual Paradigm y Rational (47).

1.5.2. Microsoft Visual Studio.

Se seleccionó como herramienta para la implementación porque es un ambiente de desarrollo integrado (IDE por sus siglas en inglés) que asegura la calidad del código y la rapidez durante todo el ciclo de construcción de aplicaciones, desde el modelado de requisitos hasta el despliegue.

Permite realizar diferentes comprobaciones, comenzando en rendimiento y finalizando en las pruebas unitarias, procesos posibles a partir de la existencia de todo un entorno de pruebas que posee. Realiza cálculos que analizan el código y muestran, por mencionar algunos datos: el nivel de mantención que una solución posee, complejidad ciclomática, la profundidad de herencia, por mencionar solo algunos. Posee una interfaz muy amigable, cómoda y fácil de aprender, con facilidades para la escritura de código en gran cantidad de lenguajes y notaciones (48).

Visual Studio 2012 Express soporta actualmente, los siguientes:

- Visual Basic
- Visual C#
- Visual C++
- Visual F#

Permite realizar diagramas de todo tipo entre los que se encuentran: los de clases y la administración de requisitos.

1.6. Tecnologías.

1.6.1. Framework .Net 4.5.

Se ha escogido la plataforma .Net 4.5 debido a que goza de gran aceptación a nivel mundial por las bondades que brinda dentro de las que se pueden nombrar: facilidad de uso, calidad de software, buen rendimiento para una plataforma que genera código gestionado, entre otras. Se define físicamente como un componente instalable al sistema operativo Windows (49).

Esta plataforma define un flujo de trabajo sencillo, el cual posibilita la rápida, eficiente y eficaz salida de productos de gran calidad y seguridad. Marca nuevos paradigmas para desarrollar aplicaciones de escritorio, componentes reutilizables tanto para sistemas operativos como para otras aplicaciones; ofreciendo soluciones destinadas a gran cantidad de problemas cotidianos que se encuentran en el desarrollo de software en la actualidad (49). Promueve el ampliamente usado paradigma de la programación orientada a objetos, lo cual, brinda extensibilidad y reusabilidad.

La plataforma se compone de dos componentes principales, el lenguaje común de tiempo de ejecución (CLR, por sus siglas en inglés) y la gran librería que esta proporciona (50).

Se puede utilizar Framework .Net 4.5 para desarrollar los siguientes tipos de aplicaciones y servicios entre otras:

- ✓ Aplicaciones de consola.
- ✓ Aplicaciones de usuario gráficas de Windows (WinForms).
- ✓ Servicios de Windows (51).

1.7. Conclusiones del capítulo.

Durante el presente capítulo se realizó un estudio a una selección de aplicaciones informáticas que tienen como característica común el asistir el proceso de representación gráfica de funciones matemáticas. El estudio realizado tuvo como objetivo caracterizarlos, teniendo en cuenta que tuvieran entre sus funcionalidades las definidas en esta investigación y que fueron enunciadas en el epígrafe 1.3. Concluyendo que los mismos entre sus funcionalidades no incluyen la de permitir al estudiante consolidar las habilidades propias para la representación gráfica de forma manual de funciones lineales y cuadráticas; por lo que se hace necesario el desarrollo de un nuevo asistente matemático que cumpla con las especificaciones previamente planteadas.

Para el desarrollo del asistente se realizó un estudio de las principales tendencias existentes en la actualidad en cuanto a metodologías, tecnologías, herramientas y lenguajes, seleccionándose finalmente aquellos que aligeraran el proceso de desarrollo.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Un sistema informático por pequeño que sea, por lo general es complicado. Por lo que se hace necesario dividirlo en pequeñas partes que al ser unidas muestren como resultado una aplicación terminada en su totalidad, además la fragmentación del proceso brinda a usuarios y desarrolladores mejores oportunidades para comprender su complejidad y gestionar el proceso de desarrollo de una manera más óptima.

Esas pequeñas partes pueden ser representadas a través de modelos que permitan abstraer sus rasgos esenciales. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y estar bien relacionados. Para dar solución al problema planteado en esta investigación, durante el presente capítulo se expone la propuesta del sistema a desarrollar. Se listan los requerimientos funcionales y no funcionales y finalmente, se detallan los casos de uso del sistema identificados, mostrándose un breve resumen de cada uno.

2.1. Representación gráfica de funciones.

2.1.1. Gráfica.

Representación de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que esos datos guardan entre sí. Pueden representar además coordenadas cartesianas (52).

2.1.2. Representación gráfica de funciones.

Permite analizar el comportamiento de un proceso, un conjunto de elementos o signos que favorecen la interpretación de un fenómeno. Facilita el estudio de una función que con una variable dependiente y otra independiente, se puede representar gráficamente en un eje de ordenadas y abscisas correspondiendo el valor de cada variable a la posición en los ejes.

2.2. Sistema propuesto.

Partiendo de la existencia de diversos sistemas informáticos que en Cuba se utilizan con la finalidad de apoyar a los estudiantes durante el proceso de consolidación de las representaciones gráficas de funciones lineales y cuadráticas, sistemas que aunque son de mucha ayuda no le brindan la posibilidad de evaluarse durante el proceso, se decidió crear una nueva aplicación informática que reúna las principales funcionalidades de los sistemas similares existentes y que fueron analizados durante el desarrollo de esta tesis y que además incorpore otras nuevas funcionalidades que sean útiles al desarrollo del proceso en cuestión.

Con el desarrollo de esta nueva aplicación informática se le brindará a los estudiantes cubanos una nueva vía para consolidar sus conocimientos, una nueva herramienta que responda a sus necesidades de interactuar y evacuar dudas referentes a la representación gráfica de funciones de primer o segundo grado, explotando así los bienes informáticos a los que tienen acceso a lo largo y ancho del país.

El sistema que se implementará será una aplicación de escritorio que les brindará a los estudiantes la posibilidad de graficar ecuaciones lineales y cuadráticas. Poseerá funcionalidades que permitirán evaluar los resultados de la representación gráfica realizada. Esto será posible porque al usuario realizar una gráfica de forma manual de

una función y elegir la opción “Revisar”, el sistema mostrará la gráfica que se corresponde con dicha ecuación, mostrando las diferencias existentes entre ambas. Las diferencias entre las gráficas se representan de diferentes colores teniendo en cuenta la escala de valores que fue definida. Además brindará la opción de ver los datos asociados a los valores obtenidos (diferencias o similitudes), así como un resultado final que se obtiene de promediar los resultados.

El sistema permitirá además consultar contenidos teóricos relacionados con las funciones lineales y cuadráticas a través de la existencia de ficheros que recogen contenido actualizado sobre estos temas y que podrá ser modificado por los profesores en caso de ser necesario.

Otra de las funcionalidades que el sistema le permitirá al usuario es la de evaluar sus habilidades relacionadas con la representación gráfica de funciones, esto es posible porque a medida que se les va dando solución al compendio de ejercicios que poseerá la aplicación, el estudiante irá consolidando o haciendo uso de contenidos más complejos que ya posee o se verá en la necesidad de investigarlos, para poder avanzar en la realización de los ejercicios de la colección. Por cada ejercicio que realice obtendrá una evaluación que le permitirá evaluar si posee habilidades o no.

El usuario podrá configurar los campos de interés dentro de la aplicación díganse lenguajes, vista de las grillas y color.

Las funcionalidades antes mencionadas permitirán que el estudiante al hacer uso de la aplicación:

1. Consolide habilidades y hábitos en la representación gráfica de funciones.
2. Consolide los aspectos teóricos de mayor relevancia relacionados con las ecuaciones de primer y segundo grado.

2.3. Modelo de dominio.

El modelo de dominio es la representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir. Define el punto de partida para el diseño del software.

En la aplicación para la consolidación de habilidades y hábitos en la representación gráfica de funciones lineales y cuadráticas denominada Graficador de Funciones Lineales y Cuadráticas (GFLC), los principales conceptos a los que se hace referencia para comprender lo que se realiza y que quedan evidenciados en el modelo de dominio que se presenta son los siguientes: asistentes matemáticos, gráfica, función, función lineal, función cuadrática, cálculos y propiedades. Estos conceptos son relevantes pues su correcto entendimiento permite comprender como ocurre la representación gráfica con asistentes computarizados.

Las relaciones entre los conceptos y que se pueden observar en la figura pueden leerse como: un asistente matemático genera la gráfica de una o muchas funciones, las cuales pueden ser funciones lineales o

cuadráticas, pero para su representación el asistente matemático debe realizar de uno a muchos cálculos, los que determinarán las propiedades que tienen las funciones.

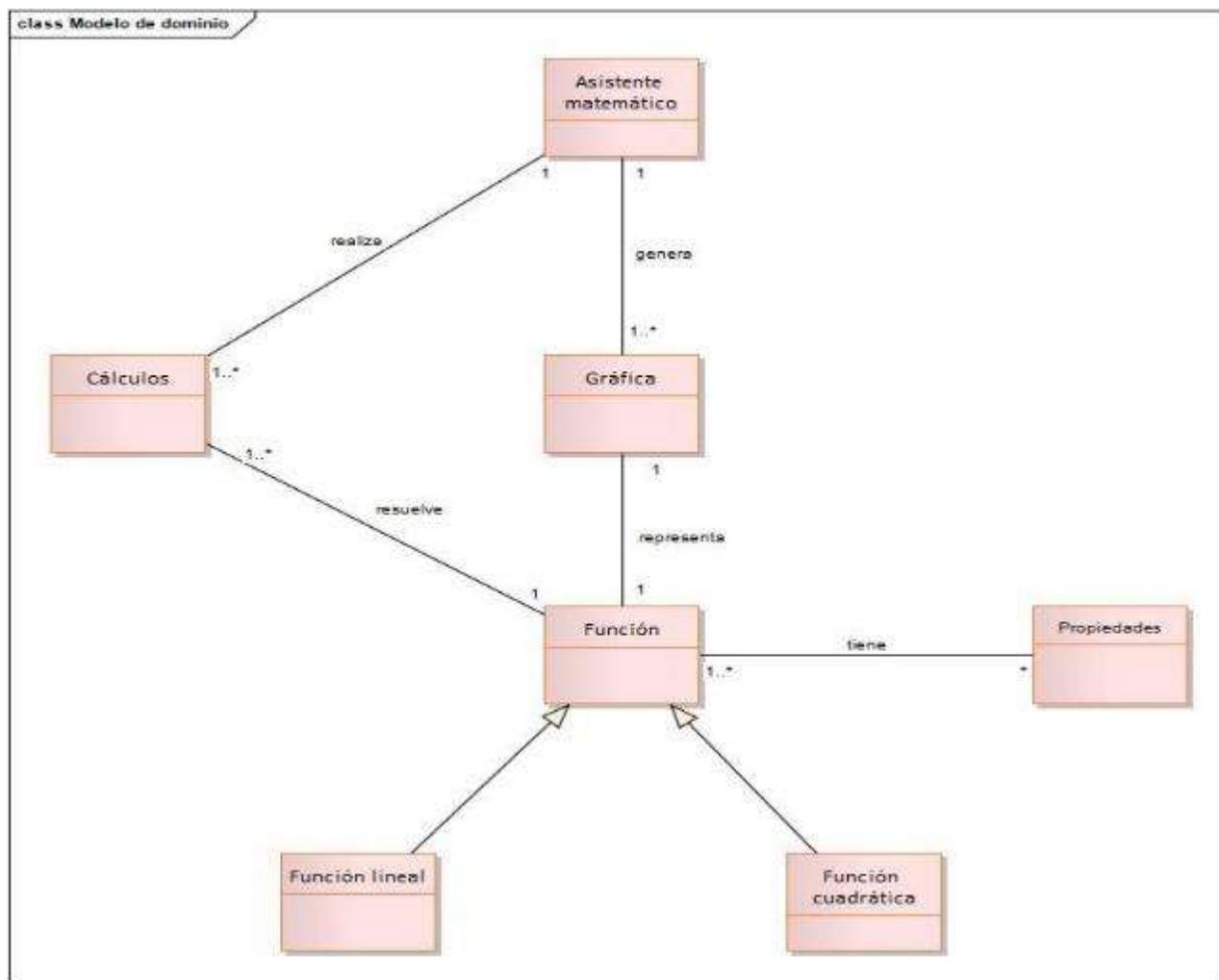


Figura # 1 Modelo de dominio.

2.4. Especificación de los requisitos de software.

En este epígrafe se realizará un análisis de los requisitos que han sido definidos y que dan solución al objetivo planteado, los mismos se agrupan en dos grandes clasificaciones: requisitos funcionales y no funcionales. Para su especificación se tuvo en consideración algunas de las funcionalidades de los sistemas estudiados en el capítulo I del presente documento y la necesidad de dar cumplimiento al objetivo general de esta investigación. ¿Qué son los requisitos?

El propósito principal del flujo de requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema, de manera que se pueda llegar a un acuerdo entre el cliente (o usuarios) y los desarrolladores, sobre qué debe y no debe hacer el sistema.

2.4.1. Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto (53).

La siguiente tabla muestra los requisitos funcionales definidos para la aplicación acompañados con la descripción de cada uno de ellos:

Requisitos	Descripción
RF 1 Insertar ecuación.	Este requisito permite al usuario introducir la ecuación de la recta o parábola que desea graficar.
RF 2 Graficar ecuación.	Este requisito permite representar la gráfica que se corresponde con la ecuación entrada.
RF 3 Mostrar contenidos teóricos sobre ecuaciones lineales y cuadráticas.	Este requisito permite mostrar contenidos con definiciones, teoremas y conceptos sobre ecuaciones lineales y cuadráticas.
RF 4 Trazar recta o parábola.	Este requisito permite al usuario trazar la recta o parábola de forma manual que considere que corresponda con la ecuación.
RF 5 Revisar.	Este requisito permite al usuario revisar la gráfica representada en el sistema coordinado, donde realiza una comparación entre la gráfica realizada por el usuario y la que realice el sistema, emitiendo según el color de los errores una evaluación.
RF 6.1 Guardar.	Este requisito permite guardar los cambios efectuados en el proyecto abierto, sobrescribiendo su última versión.
RF 6.2 Guardar como.	Este requisito permite guardar como: proyecto o como imagen.
RF 7.1 Aumentar escala del eje coordinado.	Este requisito permite que el eje coordinado aumente su tamaño original.
RF 7.2 Disminuir escala del eje coordinado.	Este requisito permite reducir el tamaño del eje coordinado.
RF 8 Abrir ficheros.	Este requisito permite abrir ficheros existentes a través de una dirección.
RF 9.1 Seleccionar la configuración.	Este requisito permite modificar los campos que se corresponden con el sistema de coordenadas.
RF 9.2 Seleccionar la vista.	Este requisito permite escoger como aparece la interfaz de la pantalla en la que estas graficando (enumerada y con grillas o lo contrario).

RF 9.3 Seleccionar idioma.	Este requisito permite seleccionar el idioma en que aparecen los campos de la aplicación.
RF 10 Limpiar	Muestra una nueva vista.
RF 11.1 Eliminar gráfica del sistema.	Elimina la gráfica realizada por el sistema.
RF 11.2 Eliminar gráfica del usuario.	Elimina la gráfica realizada por el usuario.
RF 12 Mostrar ejercicios.	Muestra un ejercicio en un formulario.
RF 13 Seleccionar ecuación.	Selecciona en el texto del ejercicio la ecuación a graficar.
RF 14 Mover eje coordenado.	Este requisito permite mover el eje coordenado para lograr una mejor visualización de la gráfica.
RF 15 Mostrar propiedades de la ecuación.	Este requisito permite mostrar las propiedades de la ecuación representada en el plano coordenado.
RF 16 Validar función.	Determina si la función entrada está bien escrita teniendo en cuenta las especificaciones del sistema.
RF 17 Salir.	Este requisito permite al usuario salir de la aplicación.

Tabla 2 Requisitos funcionales.

Una vez analizados cada uno de los requisitos del sistema podemos observar el diagrama realizado utilizando la herramienta EA a través de la siguiente figura:

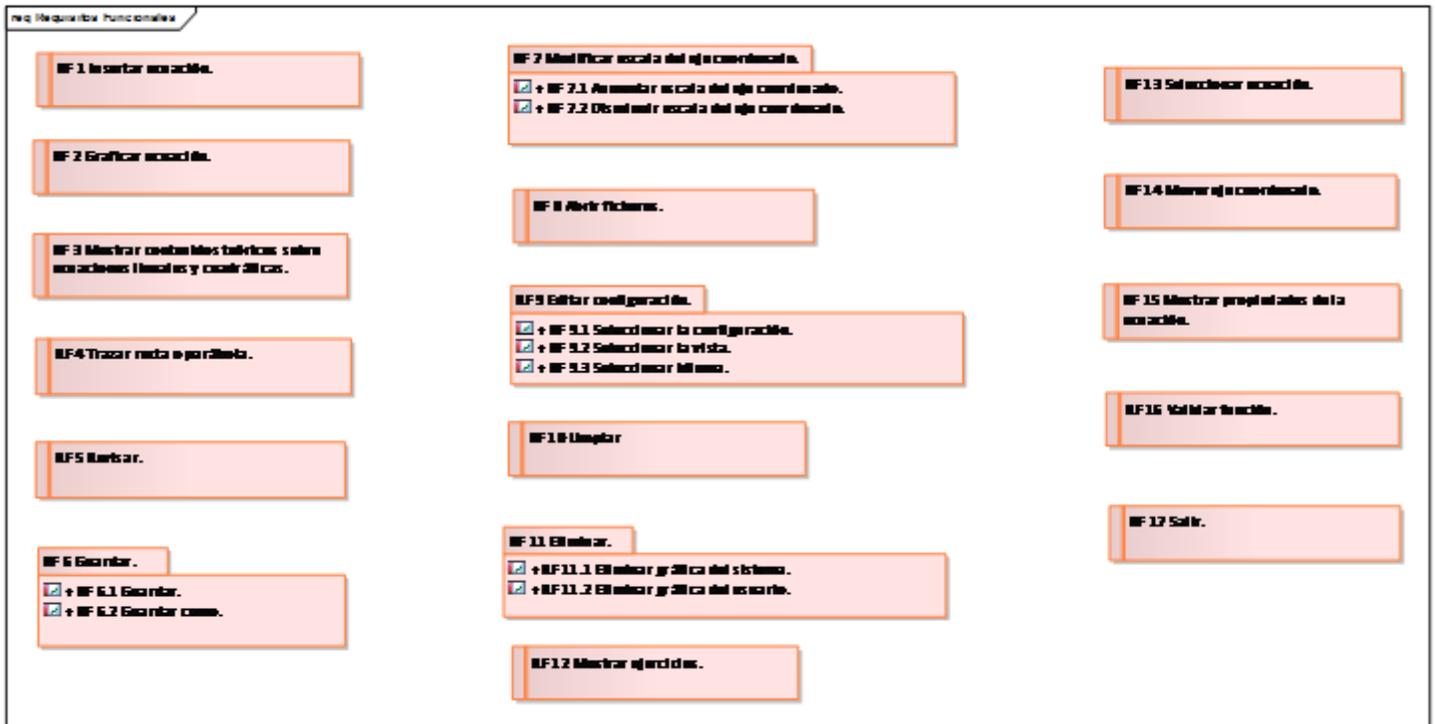


Figura # 2 Diagrama de requisitos funcionales.

2.4.2. Requisitos no funcionales.

Un requisito no funcional define una cualidad, propiedad o restricción con la que un sistema de software debe cumplir. Estableciendo una comparación con los requisitos funcionales, los no funcionales definen cómo el sistema debe ser, mientras que los funcionales definen lo que el sistema debe hacer (54).

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

La siguiente tabla muestra los requisitos no funcionales y sus descripciones pertenecientes a la aplicación informática, a los cuales se le asignan un prefijo, estos prefijos están definidos en las Pautas para la Fase de Requerimientos del Departamento de Software Médico Imagenológico (SWMI). En el [Anexo # 1](#) se muestra la tabla de los prefijos según la categoría del requisito.

Requerimiento	Descripción
RNU 1 Facilidad de empleo para usuarios sin experiencia.	El sistema debe tener una interfaz de fácil entendimiento para que usuarios inexpertos puedan familiarizarse rápidamente.
RNU 2 Configurar funcionalidades.	Debe permitir configurar según las necesidades del usuario la mayor cantidad de funcionalidades.
RNF 1 Exactitud en las salidas del sistema.	El sistema debe brindar salidas precisas.
RNE 1 Tiempo de respuesta del sistema al pedido del usuario no mayor de 1 segundo.	Ante una petición realizada y teniendo en cuenta su complejidad el sistema debe responder en no más de 1 segundo, en caso de la petición poseer una complejidad alta el tiempo de respuesta puede ser superior.
RNSO 1 Ayuda y documentación.	Se debe brindar un manual de usuario y documentación del sistema para lograr un correcto entendimiento de las funcionalidades del mismo y una mayor facilidad y experiencia de uso.
RNDI 1 Uso de Framework .Net 4.5.	Se especifica el uso de Microsoft Framework .Net 4.5 que ofrece mejoras en cuanto a administración y rendimiento.
RNDI 2 C# como lenguaje de programación.	Se deberá utilizar C# como lenguaje de programación pues está diseñado y optimizado para la plataforma .Net.
RNDI 3 Entorno Integrado de Desarrollo Microsoft Visual Studio 2012 Express.	Se utilizará Microsoft Visual Studio 2012 Express (versión libre) como entorno integrado de desarrollo.
RNDI 4 Uso de EA como herramienta CASE.	Se utilizará como herramienta CASE para el modelado el EA en su versión 7.5.
RNDI 5 Uso de UML como lenguaje de modelado.	Se utilizará como lenguaje de modelado UML en su versión 2.1.
RNDI 6 Uso de HelpNDoc como herramienta para crear ayudas.	Se utilizará HelpNDoc en su versión 3 como herramienta para crear la ayuda del sistema.
RNDI 7 Estándar de codificación.	Se utilizará el estándar con notación CamelCase (métodos), PascalCase (regiones) y Notación Húngara (atributos y variables).
RNIU 1 Deben prevalecer colores azules.	Deben prevalecer en las diferentes pantallas el color azul, para la identificación del sistema.

RNIU 2 Usar distintos tipos de líneas para el dibujo de las grillas.	La vista de las grillas puede ser a través de distintos tipos de líneas díganse puntos y líneas discontinuas.
RNFO 1 Memoria RAM de 512 Mb.	Para el correcto funcionamiento del sistema, se necesita 512 Mb o superior de memoria RAM.
RNFO 2 CPU Dual Core 3.0 GHz.	Para el correcto funcionamiento del sistema, se necesita como premisa un CPU Dual Core a 3.0 GHz.
RNFO 3 Sistema operativo Windows XP Service Pack 3 o superior.	El software debe instalarse sobre el sistema operativo Windows XP Service Pack 3 o superior.
RNFO 4 Capacidad de disco duro 50 Mb como mínimo.	La capacidad del disco duro debe ser como mínimo de 50 Mb.

Tabla 3 Requisitos no funcionales.

La siguiente figura muestra el diagrama de requisitos no funcionales del sistema realizado con la herramienta para el modelado EA.

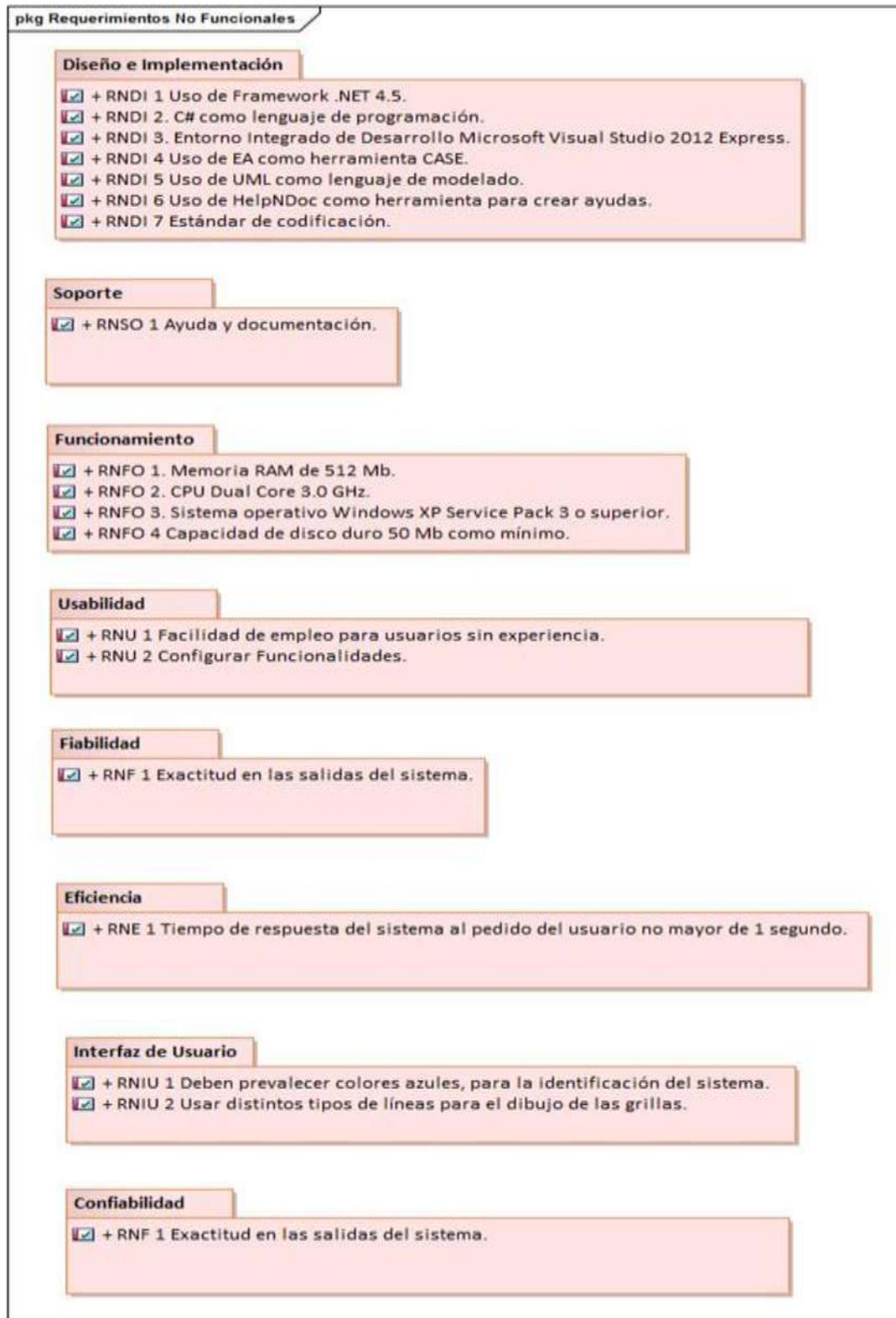


Figura # 3 Diagrama de requisitos no funcionales.

2.5. Definición de los actores del sistema.

Un actor del sistema es una persona, organización, o sistema de software que interactúa con el mismo y se beneficia de este. Un actor usa un caso de uso para desempeñar alguna porción de trabajo que es de valor para el negocio. El conjunto de casos de uso al que un actor tiene acceso define su rol global en el sistema y el alcance de su acción. En la aplicación que da solución al problema planteado el usuario final es el estudiante.

La siguiente tabla muestra la interacción del actor con los casos de uso del sistema.

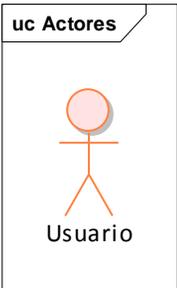
Actor	Justificación
	Inicia los casos de uso del paquete de las funcionalidades.

Tabla 4 Actor del sistema.

2.6. Diagrama de casos de uso del sistema.

El modelo de casos de uso describe las funcionalidades propuestas para el nuevo sistema, en el caso del GFLC quedaron definidos un total de 14 casos de usos que se representan a través del siguiente paquete:

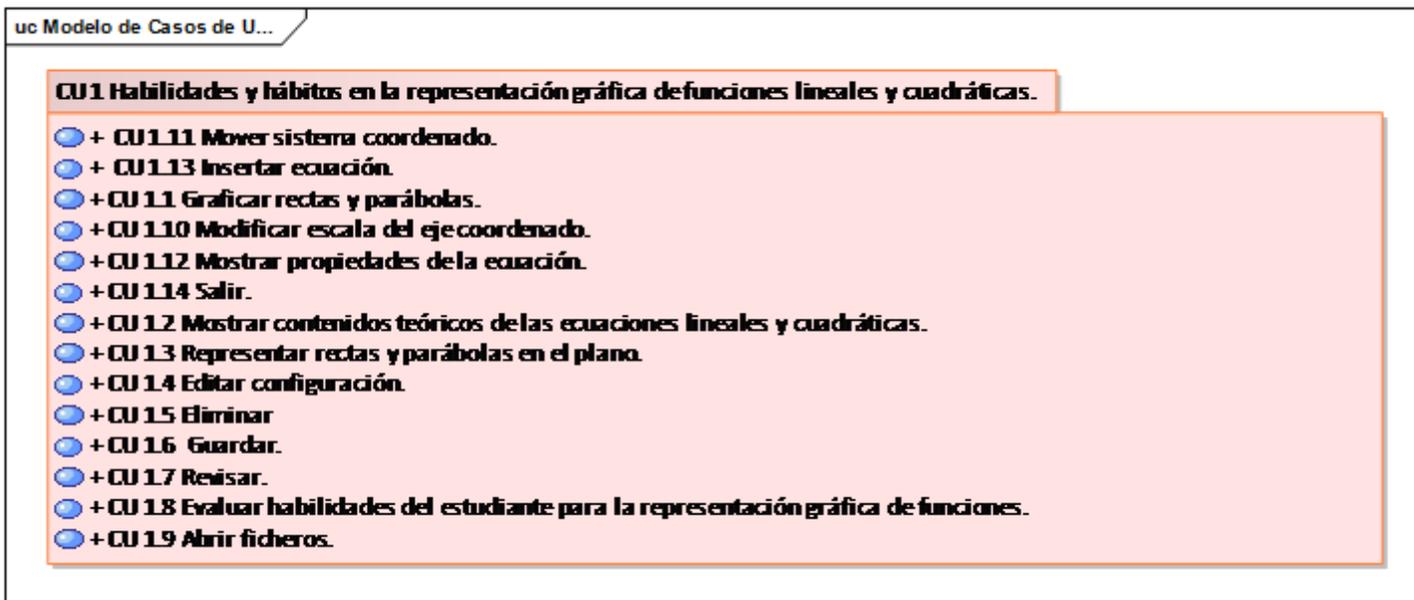


Figura # 4 Modelo de casos de uso del sistema.

Un diagrama de casos de uso del sistema representa gráficamente un proceso y su interacción con los actores del sistema. En el mismo quedan evidenciados cuales son las asociaciones que existen entre los casos de uso así como su relación con los actores del sistema.

En la figura que se muestra a continuación se reflejan la interacción existente entre los casos de uso del sistema GFLC con el usuario y entre ellos.

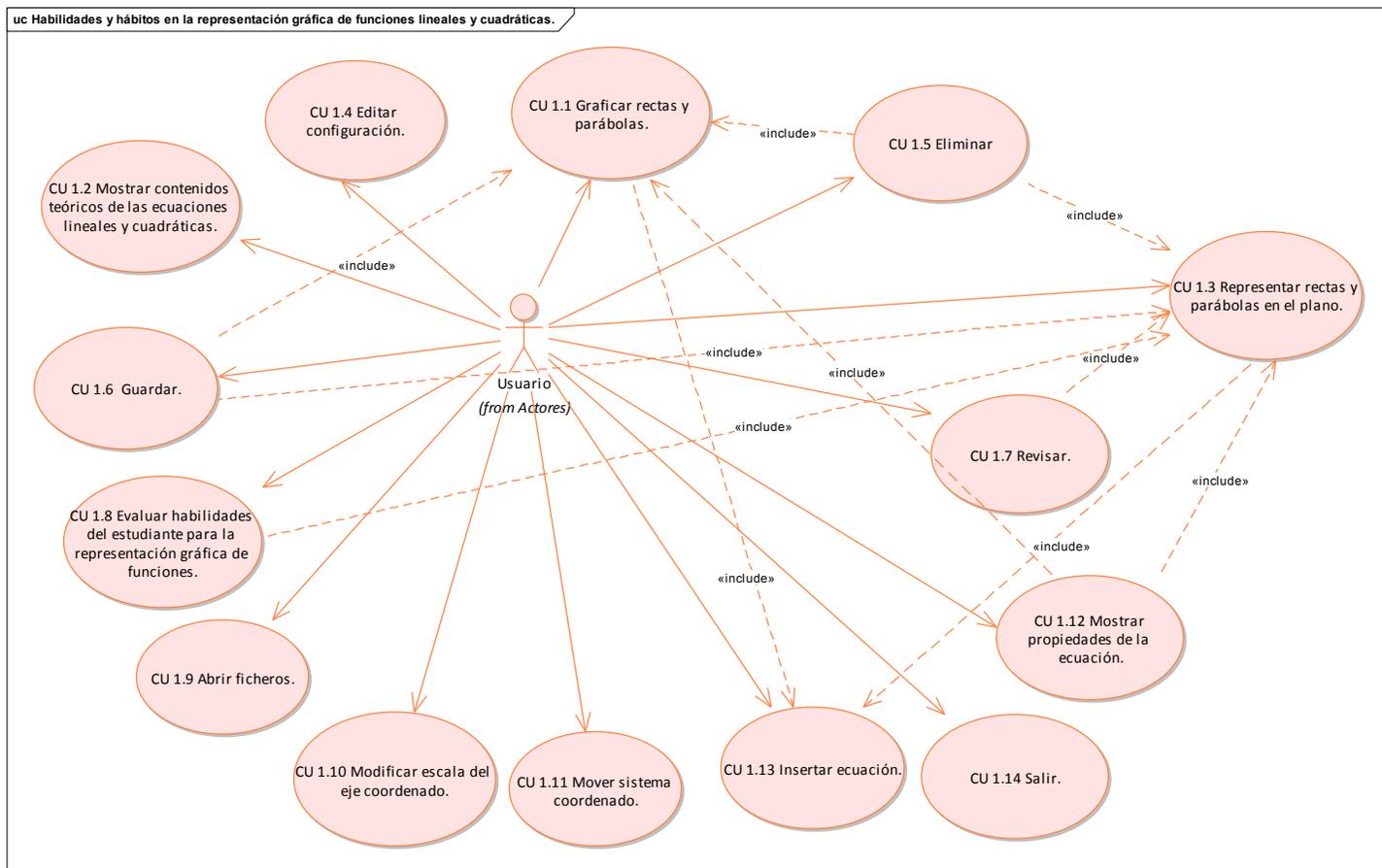


Figura # 5 Diagrama de casos de uso del sistema.

2.7. Descripción de los casos de uso del sistema.

Las tablas que se muestran a continuación contienen un resumen de los casos de uso definidos para la aplicación GFLC. Para consultar una descripción más detallada, ver [Anexo 2](#).

- Caso de uso – Graficar rectas y parábolas.

Objetivo	El objetivo que persigue dicho CU es mostrar la gráfica que le corresponde a la ecuación entrada por parámetro.
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario hace clic en el botón “Graficar” o presiona la tecla “Enter” y el sistema representa la gráfica que se corresponde con la ecuación entrada por parámetros.
Complejidad	Alta.
Prioridad	Crítica.

Referencias	RF 2.
--------------------	-------

Tabla 5 Resumen del CU #1.1 Graficar rectas y parábolas.

- Caso de uso – Mostrar contenidos teóricos de las ecuaciones lineales y cuadráticas.

Objetivo	El objetivo que persigue el actor al realizar este CU es mostrar vistas con definiciones, teoremas y conceptos sobre ecuaciones lineales y cuadráticas, el cual podrá consultar para evacuar dudas.
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Contenidos teóricos”.
Complejidad	Alta.
Prioridad	Crítica.
Referencias	RF 3.

Tabla 6 Resumen del CU #1.2Mostrar contenidos teóricos de las ecuaciones lineales y cuadráticas.

- Caso de uso – Representar rectas y parábolas en el plano.

Objetivo	El objetivo que persigue dicho CU es representar de forma manual rectas y parábolas en el plano coordenado.
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario realiza la representación gráfica de forma manual.
Complejidad	Alta.
Prioridad	Crítica.
Referencias	RF 1, RF 4, RF 16

Tabla 7 Resumen del CU #1.3 Representar rectas y parábolas en el plano.

- Caso de uso – Editar configuración.

Objetivo	El objetivo que persigue dicho CU es permitir modificar los campos que se corresponden con el sistema coordenado.
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Configuración” y modifica los campos.

Complejidad	Medio.
Prioridad	Medio.
Referencias	RF 9.1, RF 9.2, RF 9.3

Tabla 8 Resumen del CU #1.4 Editar configuración.

- Caso de uso – Eliminar.

Objetivo	El objetivo que persigue este CU es eliminar las gráficas realizadas (gráfica del usuario o gráfica del sistema).
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Eliminargráfica del sistema” o “Eliminargráfica del usuario.
Complejidad	Medio.
Prioridad	Medio.
Referencias	RF 11.1, RF 11.2, RF 10.

Tabla 9 Resumen del CU #1.5 Eliminar.

- Caso de uso – Guardar.

Objetivo	El objetivo que persigue dicho CU es permitir guardar la interfaz en la cual se está trabajando como una imagen o fichero. También brinda la opción de guardar los cambios efectuados en el archivo abierto sobrescribiendo su última versión. Este comando permite actualizar el archivo activo en cualquier momento.
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Guardar”.
Complejidad	Medio.
Prioridad	Medio.
Referencias	RF 6.1, RF 6.2

Tabla 10 Resumen del CU #1.6 Guardar.

- Caso de uso – Revisar.

Objetivo	El objetivo que persigue el actor al realizar este CU es mostrar los resultados de la revisión.
Actores	Usuario (Inicia).
Resumen	Consiste en mostrar los resultados de la revisión donde realiza una comparación de la gráfica realizada por el usuario con la que realice el sistema emitiendo según el color de la gráfica una evaluación.
Complejidad	Alta.
Prioridad	Crítica.
Referencias	RF 5.

Tabla 11 Resumen del CU #1.7 Revisar.

- Caso de uso – Evaluar habilidades del estudiante para la representación gráfica de funciones.

Objetivo	El objetivo que persigue el actor al realizar este CU es evaluar a los estudiantes mediante un conjunto de ejercicios en el cual se podrá avanzar de manera gestionada.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Ejercitación” y muestra un conjunto de ejercicios en el cual se podrá avanzar de manera gestionada.
Complejidad	Alta.
Prioridad	Crítica.
Referencias	RF 12

Tabla 12 Resumen del CU#1.8 Evaluar habilidades del estudiante para la representación gráfica de funciones.

- Caso de uso – Abrir ficheros.

Objetivo	El objetivo que persigue el actor al realizar este CU es abrir un sistema de coordenadas ya guardado como archivo.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Abrir”.
Complejidad	Medio.

Prioridad	Medio.
Referencias	RF 8

Tabla 13 Resumen del CU #1.9 Abrir ficheros.

- Caso de uso – Modificar escala del eje coordenado.

Objetivo	El objetivo que persigue dicho CU es permitir aumentar y disminuir la escala del eje coordenado.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Zoom”.
Complejidad	Baja.
Prioridad	Baja.
Referencias	RF 7.1, RF 7.2

Tabla 14 Resumen del CU #1.10 Modificar escala del eje coordenado.

- Caso de uso – Mover sistema coordenado.

Objetivo	El objetivo que persigue dicho CU es mover el sistema coordenado hacia cualquier lugar de la pantalla de la aplicación para lograr una mejor visualización de la gráfica.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar el botón “Mover” e ir al sistema coordenado y desplazarlo hacia el lugar que desea.
Complejidad	Baja.
Prioridad	Baja.
Referencias	RF 14.

Tabla 15 Resumen del CU #1.11 Mover sistema coordenado.

- Caso de uso – Mostrar propiedades de la ecuación.

Objetivo	El objetivo que persigue dicho CU es mostrar las propiedades de la ecuación representada en el plano coordenado.
Actores	Usuario (Inicia).

Resumen	Consiste en mostrar en el lado derecho de la aplicación las propiedades de la gráfica representada en el plano coordenado.
Complejidad	Medio.
Prioridad	Medio.
Referencias	RF 15.

Tabla 16 Resumen del CU #1.12 Mostrar datos de la gráfica.

- Caso de uso – Insertar ecuación.

Objetivo	El objetivo que persigue dicho CU es permitir al usuario introducir la ecuación de la recta o parábola que desea graficar.
Actores	Usuario (Inicia).
Resumen	Consiste en introducir la ecuación de la recta o parábola que desea graficar.
Complejidad	Alta.
Prioridad	Crítica.
Referencias	RF 1, RF 16.

Tabla 17 Resumen del CU#1.13Insertar ecuación.

- Caso de uso – Salir.

Objetivo	El objetivo que persigue dicho CU es permitir al usuario salir de la aplicación.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Salir”.
Complejidad	Baja.
Prioridad	Baja.
Referencias	RF 17.

Tabla 18 Resumen del CU#1.14 Salir de la aplicación.

2.8. Propuesta de casos de uso por ciclos de desarrollo.

Para el desarrollo del sistema se establecieron los siguientes ciclos de desarrollo, mediante los que se obtiene según el nivel de importancia que posean el orden en que deben ser implementados los casos de uso. En la

aplicación GFLC los casos de uso más importantes se incluyen en el primer ciclo de desarrollo ya que estos poseen las funcionalidades que hacen del sistema una aplicación funcional de no realizarse el mismo no funcionaría. Para continuar con la ubicación de los restantes se realizó un procedimiento similar al expuesto previamente.

Primer ciclo de desarrollo.

Nombre del caso de uso	Justificación
Revisar.	Estos casos de usos brindan servicios significativos para la aplicación informática.
Graficar rectas y parábolas.	
Insertar ecuación.	
Evaluar habilidades del estudiante para la representación gráfica de funciones.	
Mostrar contenidos teóricos de las ecuaciones lineales y cuadráticas.	
Evaluar habilidades del estudiante para la representación gráfica de funciones.	

Tabla 18 Casos de uso propuestos para el primer ciclo de desarrollo.

Segundo ciclo de desarrollo.

Nombre del caso de uso	Justificación
Abrir ficheros.	Estos casos de uso son de mediana prioridad.
Mostrar propiedades de la ecuación.	
Editar configuración.	
Guardar.	
Eliminar.	

Tabla 19 Casos de uso propuestos para el segundo ciclo de desarrollo.

Tercer ciclo de desarrollo.

Nombre del caso de uso	Justificación
Mover sistema coordinado.	Estos casos de uso son de baja prioridad ya que no son necesarios para lograr los objetivos de la aplicación.
Modificar escala del eje coordinado.	
Salir.	

Tabla 20 Casos de uso propuestos para el tercer ciclo de desarrollo.

2.9. Conclusiones del capítulo.

Durante el progreso del presente capítulo se consolidaron los temas referentes a los procesos de representación gráfica de funciones lineales y cuadráticas, obteniéndose de esta forma las principales características a poseer por este tipo de software. Teniendo en cuenta lo anterior se definió la propuesta de solución, la cual permite tener una visión inicial y global de lo que será el sistema a desarrollar, constituyendo una guía enfocada a obtener un sistema que cumpla con el objetivo planteado.

En esta sección del documento se definieron y establecieron los requisitos funcionales y no funcionales sobre los cuales se sustentará el desarrollo de la solución garantizando cubrir todas las capacidades o condiciones que el sistema debe tener, así como las cualidades, propiedades o restricciones con las que debe cumplir. Se identificaron y describieron los casos de uso del sistema así como su distribución por ciclos de desarrollo, obteniéndose una visión estructurada sobre cómo se realiza la interacción entre el actor y el sistema. La distribución por ciclos de desarrollo permitió desglosar en tres etapas el proceso de desarrollo, lográndose de esta forma un mejor control sobre el estado del avance del mismo.

CAPÍTULO 3. CONSTRUCCIÓN DE LA SOLUCIÓN.

En el presente capítulo se presenta la arquitectura que contendrá la aplicación, así como los diagramas de clases de diseño y de secuencia que se derivan de la realización de los casos de uso del sistema. Se brinda una breve descripción de las clases involucradas en la realización de los casos de uso arquitectónicamente significativos para el desarrollo del software.

Partiendo de los resultados del diseño, se analizará el flujo de trabajo implementación, el cual describe cómo los elementos del modelo del diseño se implementan en términos de componentes, lo que quedará representado en el diagrama de componentes. Además se realizará el modelo de datos y se observarán las relaciones entre ellos.

3.1. Modelo arquitectónico.

La arquitectura de software se refiere al diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad y disponibilidad (55).

La arquitectura es un conjunto de patrones y abstracciones coherentes que brindan el marco de referencia para guiar la construcción del software, dando cumplimiento a los objetivos del sistema. El objetivo fundamental de la arquitectura es la identificación de los requisitos que producen un impacto en la estructura del sistema y a la vez reducir los riesgos asociados con la construcción del mismo. La misma debe orientarse a las necesidades del cliente y ser flexible en cuanto a futuras modificaciones tanto de software como de hardware evitando que efectos inesperados puedan tener repercusión en el sistema.

El sistema Generador de Gráficas Lineales y Cuadráticas es una aplicación de escritorio sencilla y para su desarrollo se definió una arquitectura en tres capas. La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. Permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles.

Capas:

1. Capa de presentación: contiene las interfaces que ve el usuario, captura la entrada de datos y comunica resultados. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio. En la aplicación GFLC la capa de presentación contiene dos niveles: uno con el diseño de todas las vistas y el otro

con la implementación de los eventos de los componentes presentes en dichas vistas (ejemplos de vistas en GFLC: Configuracion.cs, Graficar.cs, GraficarUsuario.cs, etc.).

2. Capa de negocio: incluye las clases encargadas de procesar la información necesaria para brindar servicios a la capa de presentación (recibir solicitudes y presentar resultados) y con la capa de datos, para solicitar datos existentes en ella. Ejemplos de clases dentro de esta capa en la aplicación GFLC: Calculos.cs, Funcion.cs, Signos.cs, etc.
3. Capa de datos: esta capa según las buenas prácticas de los diseñadores debe ser dividida en dos capas independientes que se muestran a continuación:
 - 3.1. Acceso a datos: se encarga de interactuar con la capa de datos brindándole la información persistida en esta a la capa inmediata superior (capa de negocio). Permite abstraerse de los medios en que permanecen almacenados los datos (ficheros o bases de datos). Una de las clases en la aplicación que realizan esta funcionalidad es Fichero.cs.
 - 3.2. Datos: es donde residen los datos, en el caso de la aplicación la forma de persistencia que se utiliza es a través de ficheros.

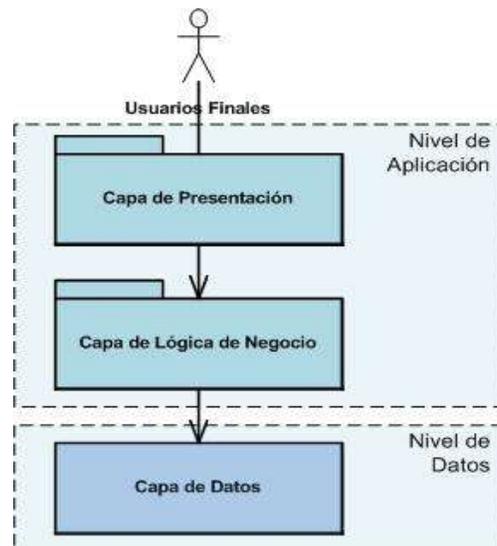


Figura # 6 Diagrama de la arquitectura en tres capas.

Los patrones de diseño frecuentes dentro de la arquitectura en tres capas son los patrones de asignación de responsabilidades (GRASP, por sus siglas en inglés): Experto, Creador, Bajo acoplamiento y Alta cohesión. Ellos garantizan robustez, adaptabilidad, estabilidad, facilidad de mantenimiento, flexibilidad, reusabilidad y

extensibilidad, debido al desacoplamiento entre sus componentes, lo que permitirá la sustitución, en tiempo de ejecución, de componentes por otros, sin afectar el correcto funcionamiento del sistema y reducirá la necesidad de realizar modificaciones en el código fuente (56). En la aplicación GFLC se pusieron en práctica en las siguientes clases:

Experto: se evidencia en las clases o vistas que poseen la responsabilidad y la información necesaria para asignarle a cada elemento el cumplir una responsabilidad.

- Clase: Graficar y se puede observar en el siguiente fragmento de código.

```
private void graficarFuncion(object sender, EventArgs e)
{
    try
    {
        this.Cursor = Cursors.WaitCursor;
        a = default(float);
        b = default(float);
        EliminarGraficaMenu.Enabled = true;
        todoToolStripMenuItem.Enabled = true;
        arbol = new Funcion(FuncionText.Text, IntervaloText.Text);
        objConfiguracion.CorrMientox = objConfiguracion.CorrMientoy = 0;
        grafico.Clear(Color.White);
        pintarGrilla();
        pintarEjesCoordenadas();
        pintarFuncion();
        pnlEscala.Visible = false;
        cargarPropiedades(arbol.Propiedades.Obtener());
        GuardarButton.Enabled = true;
        this.Cursor = Cursors.Default;
        tstIrA.Text = "(X;Y)";
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);
        EliminarGraficaMenu.Enabled = false;
        eliminarGrafica(sender, e);
        this.Cursor = Cursors.Default;
    }
}
```

Figura # 7 Evidencia del patrón Experto en GFLC.

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos.

- Clase: Funcion y se puede observar en el siguiente fragmento de código.

```
public Funcion(string raiz, Funcion izquierdo, Funcion derecho)
{
    this.raiz = raiz;
    this.derecho = derecho;
    this.izquierdo = izquierdo;
    this.c = new Controler();
    this.listaPuntosX = new List<double>();
    this.listaPuntosY = new List<double>();
    this.inicioIntervalo = new IntervaloInicio(-1000000, true);
    this.finIntervalo = new IntervaloFin(1000000, true);
}
```

Figura # 8 Evidencia del patrón Creador en GFLC.

Bajo acoplamiento: es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Significa que una clase no depende de muchas clases.

- En el caso de GFLC la clase que más relaciones tiene es la Funcion.cs y solo se relaciona con 3 clases.

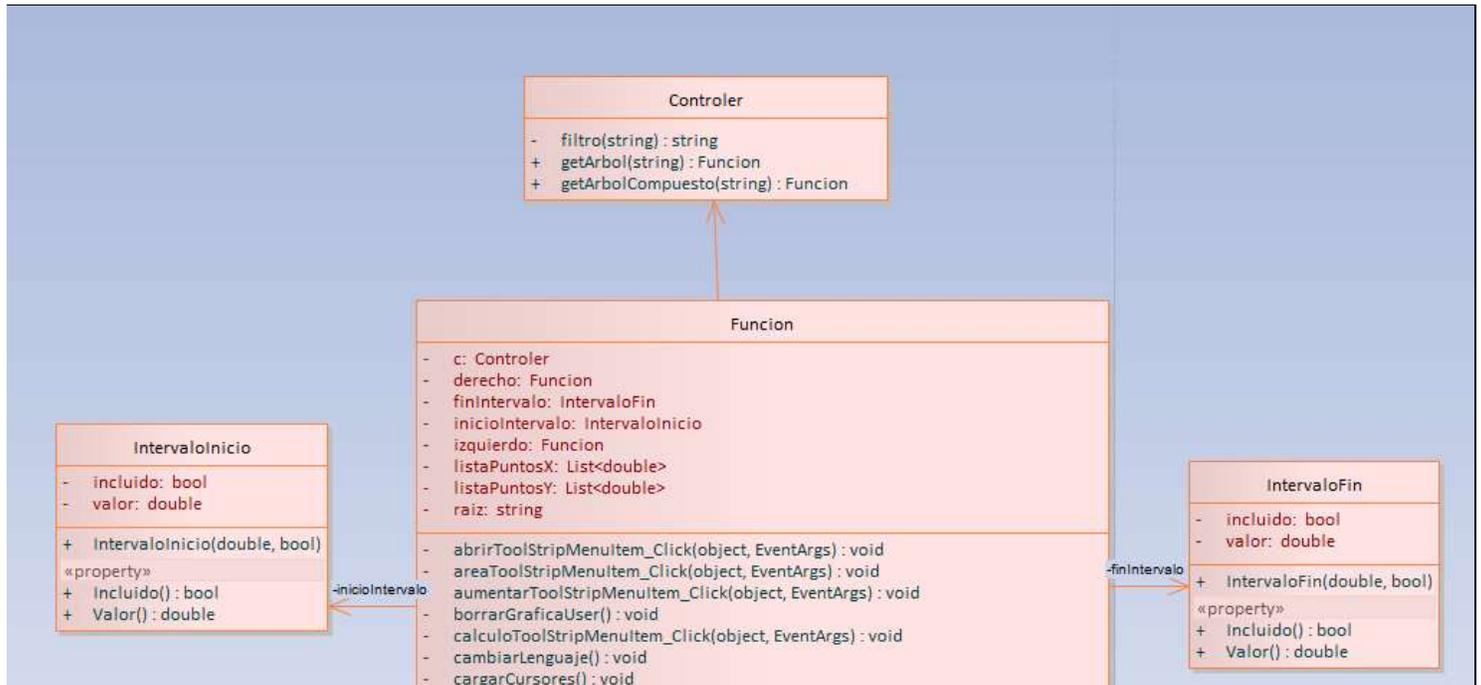


Figura # 9 Evidencia del patrón Bajo acoplamiento en GFLC.

Alta cohesión: la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas para que no realicen un trabajo enorme.

- Clase: Calculo pone de manifiesto este patrón ya que ella es la única responsable de realizar las operaciones de cálculos de las funciones.

3.2. Diseño.

En la fase de diseño se modela el sistema, se especifica y construye una arquitectura sólida para el mismo. El diseño provee una imagen completa del software, desde una perspectiva de implementación de tal forma que el sistema cumpla con parámetros como la reusabilidad, portabilidad, compatibilidad y potencia, lo que permite a todo el software alcanzar un alto nivel de calidad. Define la relación entre los elementos estructurales más

importantes del software, los estilos arquitectónicos y patrones de diseño que pueden usarse para satisfacer los requisitos especificados para el sistema, incluyendo los no funcionales.

Esta fase tiene como objetivo fundamental convertir los requisitos del software en especificidades que describan cómo va a ser implementado el sistema. El diseño debe estar bien especificado para que el sistema pueda ser implementado sin imprecisiones, para ello se elaboran los diagramas de clases del diseño que muestran las clases participantes en la realización de un caso de uso con todos sus atributos y las relaciones entre ellas.

A continuación se muestran los diagramas de clases del diseño para los casos de uso arquitectónicamente significativos en la aplicación a realizar y que fueron definidos en el primer ciclo de desarrollo



Figura # 10 Diagrama de clases del diseño del CU #3 Representar rectas y parábolas en el plano.

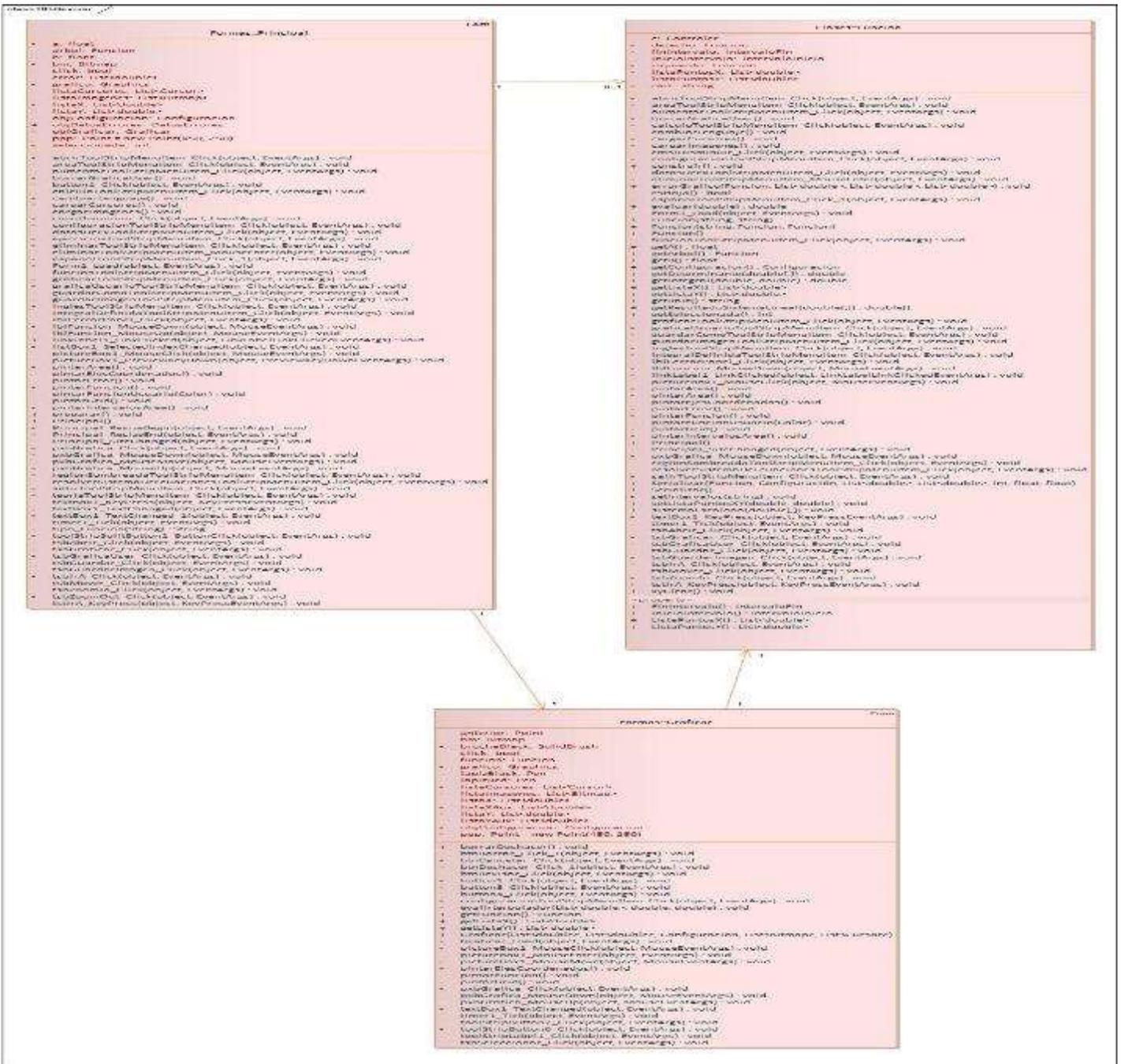


Figura # 11 Diagrama de clases del diseño del CU #7 Revisar.

Los restantes diagramas de clases del diseño se muestran en el [Anexo 3](#).

Los diagramas de secuencia muestran gráficamente las interacciones del actor y de las operaciones a las que dan origen. Con el objetivo de alcanzar un entendimiento de las actividades que se llevan a cabo en los casos de uso arquitectónicamente significativos, se realizan los siguientes diagramas utilizando el EA como herramienta para el modelado.

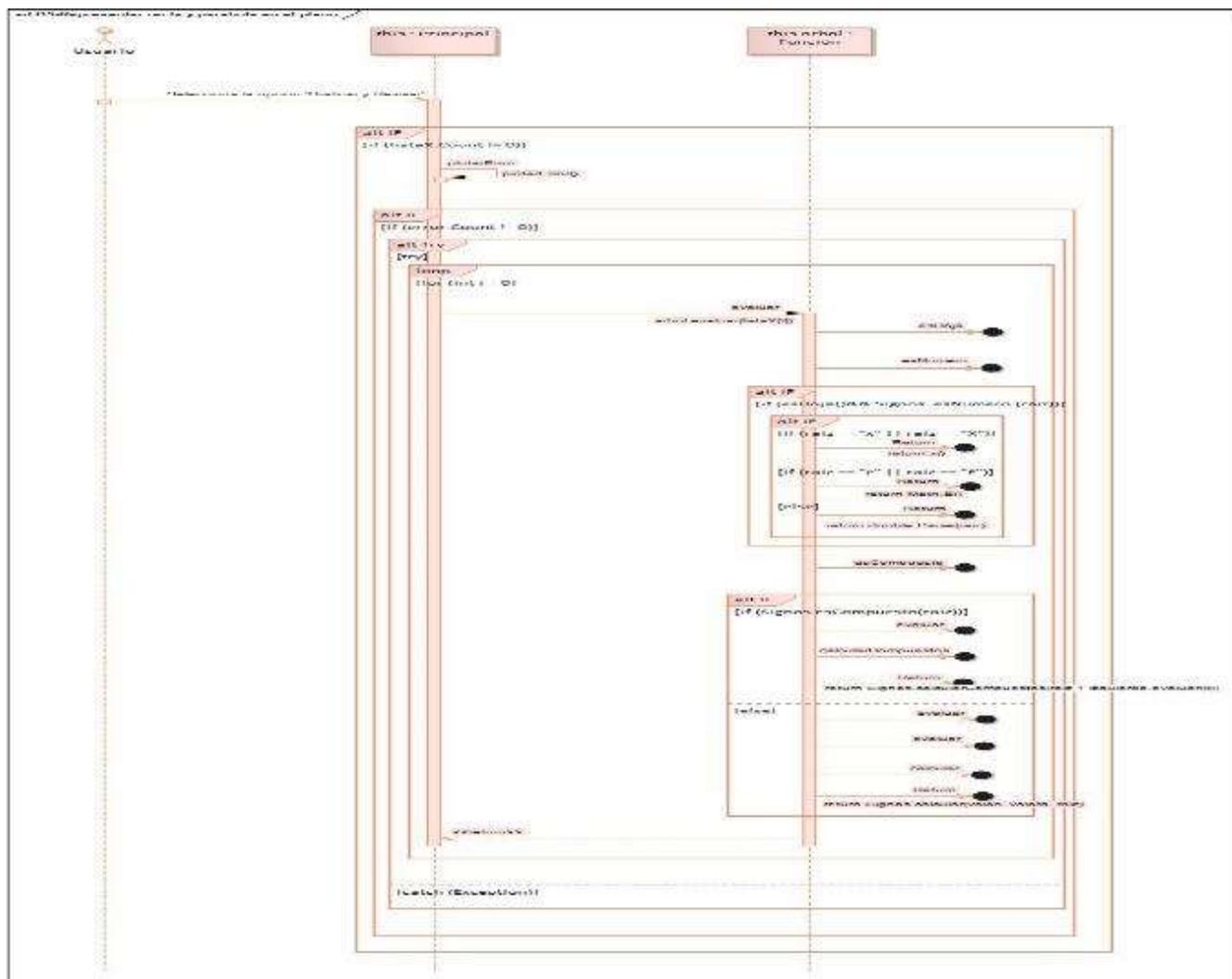


Figura # 12 Diagrama de secuencia del CU #3 Representar rectas y parábolas en el plano.

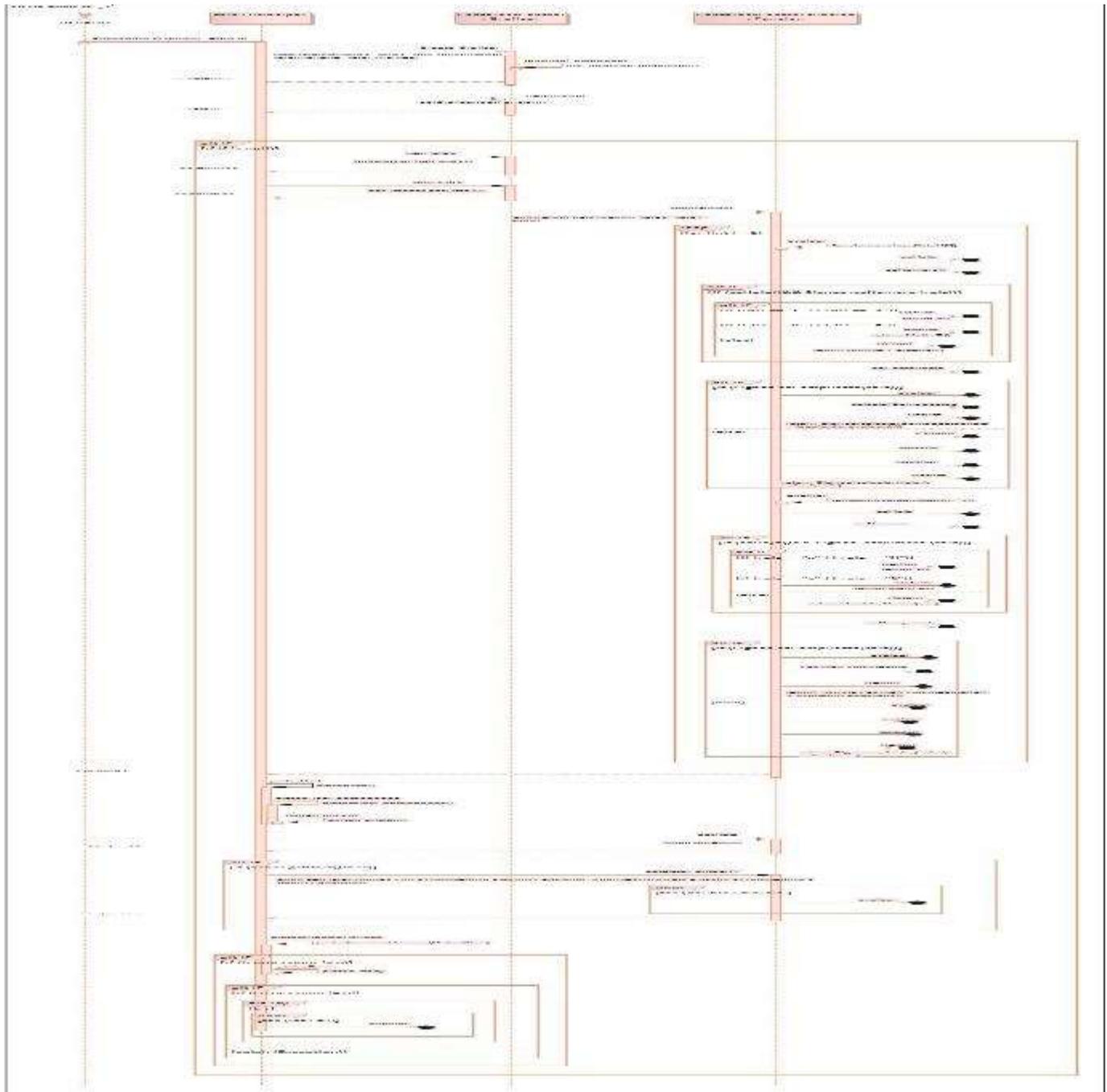


Figura # 13 Diagrama de secuencia del CU #7 Revisar.

Para mayor profundización en la interacciones usuario (estudiante) / sistema ver [Anexo 4](#)

Descripción de las clases

La descripción de las clases del sistema brinda información acerca de la interacción de las clases involucradas en la realización de los casos de uso, en las mismas se describen cada uno de los elementos presentes díganse métodos y atributos. Esta información podrá ser usada por los desarrolladores quienes encontrarán en ellas descripciones específicas sobre lo que deben construir. Las descripciones de estas clases se encuentran en el [Anexo 5](#).

3.3. Implementación.

El objetivo principal de la etapa de implementación es definir la organización del código, planificar las integraciones de sistema necesarias en cada iteración e implementar las clases y subsistemas encontrados durante la fase de diseño. Una vez terminada la misma se pueden mostrar al usuario resultados visuales del sistema, de las partes físicas que lo componen, así como la manera en que deben ser puestos en funcionamiento. Unos de los resultados visuales obtenidos durante la implementación del sistema GFLC fue el diagrama de los componentes que lo integra (57).

3.3.1. Diagrama de Componentes de GFLC.

Un componente es una parte física y reemplazable de un sistema, conformado por un conjunto de interfaces y que realizan esas interfaces. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos como ficheros ejecutables, binarios; entre otros. En un componente se puede implementar uno o varios elementos, ejemplo varias clases. Sin embargo la forma exacta en que se crea esta traza depende de cómo van a ser estructurados y modularizados los ficheros de código fuente, dado el lenguaje de programación que se esté usando (C# en este caso) (58).

El diagrama de componentes de GFLC muestra las organizaciones y dependencias lógicas entre componentes de dicho software. Para su realización se tuvo en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por el lenguaje de programación y las herramientas utilizadas en su desarrollo. En el mismo se muestran un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones (59).

Los estereotipos estándar de componentes utilizados en el software son los siguientes:

- <<executable>> programa que puede ser ejecutado en un nodo.
- <<file>> fichero que contiene código fuente o datos.
- <<document>> documento.

En la figura se muestra el diagrama de componentes de la aplicación GFLC en el que se observan las relaciones existentes entre el ejecutable y los ficheros binarios que contienen el código de las operaciones que se realizan dentro de la aplicación. Su realización sirve para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. Además muestra la interacción que establece el ejecutable con los documentos adjuntos al sistema (Documento de ayuda para el uso de GFLC). Las relaciones de dependencia entre ellos son de uso (<<use>>).

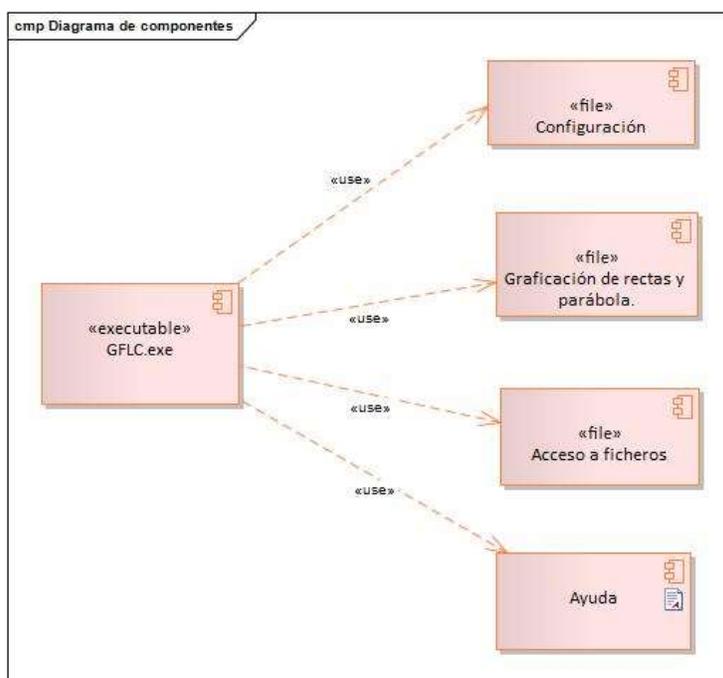


Figura # 14 Diagrama de componentes.

3.3.2. Estándares para la codificación

Durante la implementación del software GFLC se utilizaron los siguientes estándares para la codificación según lo que planteaba el RNDI 6 y los mismos fueron utilizados en las diferentes partes que conforman la solución como se especifica a continuación:

Los métodos: definidos con la notación CamelCase, estilo de escritura que se aplica a frases o palabras compuestas, se puede traducir como Mayúsculas/Minúsculas Camello. De los dos tipos existentes y especificados a continuación, la aplicación exhibe el estilo lowerCamelCase.

- UpperCamelCase, primera letra de cada una de las palabras es mayúscula. Ejemplo: EjemploDeUpperCamelCase

- lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: ejemploDeLowerCamelCase.

Las regiones: usan la notación PascalCase que al igual que en UpperCamelCase la primera letra también es mayúscula.

Los atributos y variables: exhiben un estilo definido por la Notación Húngara notación que se basa en Camel, añadiendo al principio del identificador una secuencia de letras en minúscula, que indica alguna característica del identificador, como su tipo en el caso de variables. Por ejemplo:

- iValue podría denotar una variable entera.
- fValue una variable float.
- frmPartner la instancia de un formulario (60).

3.3.3. Diagrama de clases persistentes.

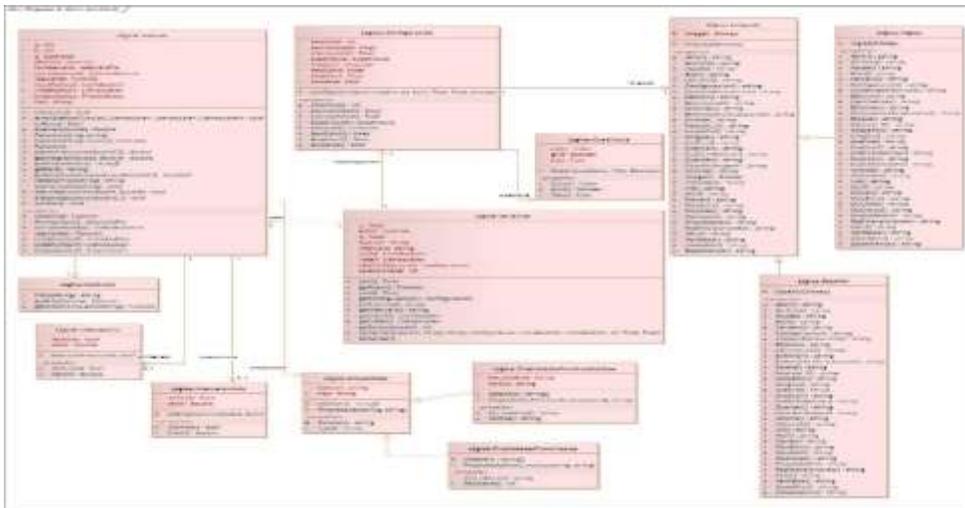


Figura # 15 Diagrama de clases persistentes.

Una vez realizado el modelo de las clases persistentes dentro de la aplicación, se determinó la relación entre los datos que ellas gestionan a través de un modelo de datos. Dicho modelo no es más que la representación de conjunto de ideas lógicas utilizadas para representar la estructura y las relaciones entre ellos.

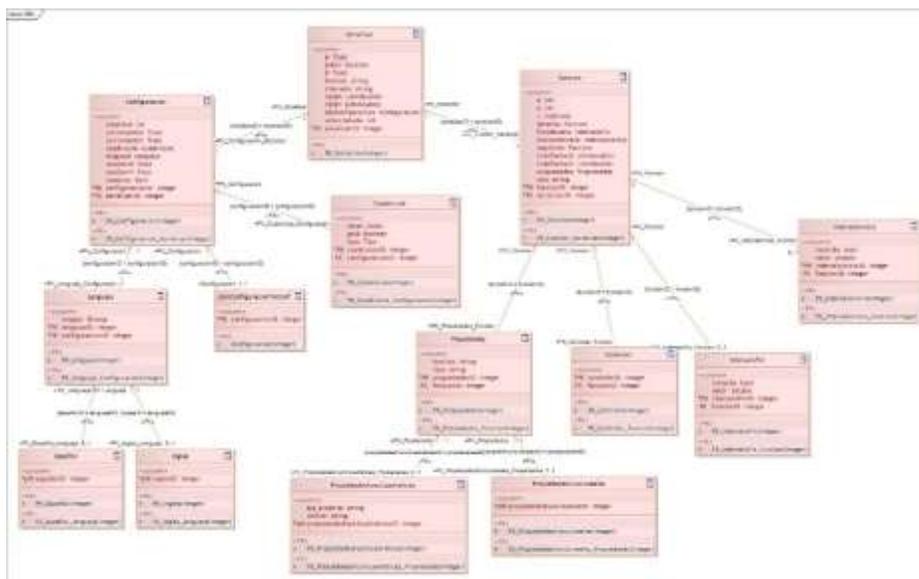


Figura # 16 Modelo de datos.

3.4. Conclusiones del capítulo.

Durante el desarrollo del presente capítulo se definió la arquitectura del sistema. La misma permitió la identificación de los atributos de calidad que producen un impacto en su estructura y a la vez redujo los riesgos asociados a la construcción del mismo. Fueron escogidos los patrones de diseño que permiten satisfacer las especificaciones de la actual investigación y que cuyas soluciones resolvieron escenarios similares.

A partir del modelamiento del sistema se realizaron las especificaciones de las clases para el diseño y sus diagramas (de clases del diseño, de secuencia y de trazabilidad) correspondientes a cada caso de uso del sistema. Dichos diagramas y especificaciones detallaron las particularidades de cada una de las clases involucradas en la realización de los casos de uso, brindándole al implementador una guía más precisa, en términos de programación, sobre lo que debe hacer; facilitando así el proceso de implementación.

Se realizaron las actividades correspondientes al flujo de trabajo implementación obteniéndose ya en esta etapa resultados visibles para los usuarios a través del diagrama de componentes. El cual muestra los elementos concernientes a la implementación del sistema y las relaciones que existen entre cada uno de ellos. Permitiendo obtener por parte del usuario una primera valoración de lo que sería el resultado final. Además se obtuvo el modelado de los datos a partir del diagrama de clases persistentes en la solución, lo que permitió describir los elementos que intervienen en el problema dado y la forma en que se relacionan esos elementos entre sí.

CONCLUSIONES.

Durante la presente investigación se realizó un estudio de los asistentes matemáticos utilizados en el departamento de Ciencias Básicas de la facultad 7 de la UCI, para la consolidación del conocimiento referente a la representación gráfica de funciones lineales y cuadráticas por los estudiantes y se determinó la necesidad de desarrollar un sistema que les permitiera participar en la confección de dichas gráficas.

El sistema fue creado teniendo como base los requisitos obtenidos a partir de la observación de funcionalidades presentes en otros sistemas, los cuales cuentan con características similares a las perseguidas por la investigación.

Dicho proceso de desarrollo fue guiado por la metodología RUP, la cual admitió la generación de todos los artefactos asociados al desarrollo y permitió que los mismos quedaran documentados.

La aplicación desarrollada es el resultado del uso de una arquitectura en tres capas y orientado a objetos; fue definida a medidas que se realizaron las fases de la metodología y que permitió distribuir el trabajo de creación por niveles; de este modo, cada grupo de trabajo estuvo abstraído del resto de los niveles.

El sistema desarrollado cumple con los requisitos de software solicitados por el cliente y se ajusta a sus necesidades según aval emitido por el mismo.

RECOMENDACIONES.

Luego de haber cumplido con los objetivos planteados en la investigación, se considera que para lograr que el sistema creado aumente sus funcionalidades, el mismo debe permitir:

- Editar los contenidos teóricos y de ejercitación a través de la propia aplicación por un personal autorizado (profesor).
- Aumentar en el software la capacidad de realizar operaciones relacionadas con funciones trigonométricas, logarítmicas, polinomiales y exponenciales.

REFERENCIAS BIBLIOGRÁFICAS.

1. Revolución informática. *sites.google.com/site/revotica0622/primer-computadora*. [En línea] [Citado el: 6 de junio de 2013.] <https://sites.google.com/site/revotica0622/primer-computadora>.
2. Definición.org. *www.definicion.org/informatica*. [En línea] [Citado el: 6 de junio de 2013.] <http://www.definicion.org/informatica>.
3. DR. JUAN GUI TERAS GENE. Revista Médica Electrónica. *scielo.sld.cu/scielo.php*. [En línea] [Citado el: 6 de junio de 2013.] http://scielo.sld.cu/scielo.php?pid=S1684-18242009000300012&script=sci_arttext.
4. Las TICs en la educación. *boj.pntic.mec.es/jgomez46/ticedu.htm*. [En línea] [Citado el: 6 de junio de 2013.] <http://boj.pntic.mec.es/jgomez46/ticedu.htm>.
5. Software educativos en Cuba. *www.ecured.cu/index.php/Software_educativos_en_Cuba#Tipos_de_software_educativos*. [En línea] [Citado el: 6 de junio de 2013.] http://www.ecured.cu/index.php/Software_educativos_en_Cuba#Tipos_de_software_educativos.
6. Defición de... *definicion.de/fenomeno/*. [En línea] [Citado el: 14 de diciembre de 2012.] <http://definicion.de/fenomeno/>.
7. Informática Educativa en el Currículum de Enseñanza Media. *www.eduteka.org/pdfdir/ChileCurrículoMatemáticasTics.pdf*. [En línea] Red de Asistencia Técnica de Enlaces, Ministerio de Educación de Chile. [Citado el: 20 de febrero de 2013.] <http://www.eduteka.org/pdfdir/ChileCurrículoMatemáticasTics.pdf>.
8. Importancia De Las Matemáticas En La Carrera De Ingeniería De Sistemas e Informática. *www.buenastareas.com/ensayos/Importancia-De-Las-Matemáticas-En-La/2391461.html*. [En línea] [Citado el: 20 de marzo de 2013.] <http://www.buenastareas.com/ensayos/Importancia-De-Las-Matemáticas-En-La/2391461.html>.
9. Universidad Nacional Experimental del Táchira. Competencias del Ingeniero en Informática | Sitio de la Universidad Nacional Experimental del Táchira. *admission.unet.edu.ve/pages/main.php?id=8001*. [En línea] [Citado el: 12 de febrero de 2013.] <http://admission.unet.edu.ve/pages/main.php?id=8001>.
10. Buenas tareas. *www.buenastareas.com/ensayos/Áreas-Donde-Se-Aplican-Las-Matemáticas/1141084.html*. [En línea] noviembre de 2010. [Citado el: 4 de diciembre de 2012.] <http://www.buenastareas.com/ensayos/Áreas-Donde-Se-Aplican-Las-Matemáticas/1141084.html>.

11. Silva, Erika Cecilia Parra. USO DE HERRAMIENTA WEB EN EL APRENDIZAJE DE FUNCIONES. *revistas.upel.edu.ve/index.php/educare/article/view/59/59*. [En línea] [Citado el: 15 de febrero de 2013.] <http://revistas.upel.edu.ve/index.php/educare/article/view/59/59>.
12. Aedo, Ing. Sergio Fernández. Solución a Ejercicios de Matemáticas y Ejercicios de Física, Geometría. *www.jfinternational.com/funciones-matematicas.html*. [En línea] [Citado el: 11 de diciembre de 2012.] <http://www.jfinternational.com/funciones-matematicas.html>.
13. NUEVAS TECNOLOGIAS. *edutecusac.blogspot.com/2011/03/definiciones-de-tics.html*. [En línea] EDUTEC. [Citado el: 12 de diciembre de 2012.] <http://edutecusac.blogspot.com/2011/03/definiciones-de-tics.html>.
14. Tecnología al Instante. *www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=2060*. [En línea] Estudios Pro Nexus Radical. [Citado el: 12 de febrero de 2013.] http://www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=2060.
15. GÓMEZ, ENRIQUE HUAPAYA. *MODELACIÓN USANDO FUNCIÓN CUADRÁTICA: EXPERIMENTOS DE ENSEÑANZA CON ESTUDIANTES DE 5TO DE SECUNDARIA*. PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ. Lima : s.n., 2012. pág. 16, Tesis para optar el grado de Magister en Enseñanza de las Matemáticas.
16. Juega y Construye la Matemática. *sites.google.com/site/timatematicas/home/asistentes-matematicos*. [En línea] José L. Orozco Tróchez, noviembre de 2008. [Citado el: 10 de enero de 2013.] <https://sites.google.com/site/timatematicas/home/asistentes-matematicos>.
17. Richard, Lic. Cesar Nicolas. *Propuesta didáctica para el aprendizaje de la matemática numérica con la utilización de Entorno Virtual de Aprendizaje*. Ciencias Básicas, CUJAE. La Habana : s.n., 2007. pág. 43, Tesis de Maestría.
18. Sosa, José B. Rodríguez. *Una propuesta metodológica para la utilización de las tecnologías de la información y las comunicaciones en el proceso de enseñanza aprendizaje de las funciones matemáticas*. La Habana : s.n., 2003.
19. Lapuente, María Jesús Lamarca. Hipertexto, el nuevo concepto de documento en la cultura de la imagen. *www.hipertexto.info/documentos/interactiv.htm*. [En línea] 5 de mayo de 2011. [Citado el: 12 de diciembre de 2012.] <http://www.hipertexto.info/documentos/interactiv.htm>.
20. *Conceptualización de un nodo virtual de procesos*. Y. Gámez, V. Moreno and Y. Martínez. La Habana, : Informática, 2009.

21. Buenas Tareas. www.buenastareas.com/ensayos/Graficacion-Por-Computadora/110998.html. [En línea] febrero de 2010. [Citado el: 11 de diciembre de 2012.] <http://www.buenastareas.com/ensayos/Graficacion-Por-Computadora/110998.html>.
22. Tutorial sobre Matlab Capítulo 2. catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo2.pdf. [En línea] [Citado el: 10 de febrero de 2013.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo2.pdf.
23. Curso Rápido de Matlab. www.iit.upcomillas.es/palacios/matlab/curso_matlab.pdf. [En línea] [Citado el: 3 de febrero de 2013.] http://www.iit.upcomillas.es/palacios/matlab/curso_matlab.pdf.
24. Introducción a la programación con Matlab. mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf. [En línea] Departamento de Física y Matemática Aplicada, 2012. [Citado el: 14 de enero de 2013.] <http://mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf>.
25. Información sobre el programa Derive. www.upv.es/derive/general.htm. [En línea] Asociación de Usuarios de Derive de España, 2005. [Citado el: 7 de enero de 2013.] <http://www.upv.es/derive/general.htm>.
26. Derive 6. www.chartwellyorke.com/derive.html. [En línea] [Citado el: 21 de enero de 2013.] <http://www.chartwellyorke.com/derive.html>.
27. Softonic Descargas. derive.softonic.com. [En línea] Softonic, 2012. [Citado el: 17 de diciembre de 2012.] <http://derive.softonic.com>.
28. Computación Científica. www.scientific-computing.com/scwmarapr04derive6.html. [En línea] [Citado el: 21 de enero de 2013.] <http://www.scientific-computing.com/scwmarapr04derive6.html>.
29. Tangient LLC. Matemáticas: Integrando Tecnologías para una mejor Formación. vumatematicas.wikispaces.com/Informe+sobre+Winplot. [En línea] [Citado el: 2 de febrero de 2013.] <http://vumatematicas.wikispaces.com/Informe+sobre+Winplot>.
30. winplot. softonic. winplot.softonic.com/. [En línea] softonic, 2012. [Citado el: 14 de diciembre de 2012.] <http://winplot.softonic.com/>.
31. PI N° 686. Sistema AyD SEC CHA Depto. Informática- Fac. Ingeniería. http://www.ing.unp.edu.ar/enlaces_conceptuales/index.php?option=com_content&view=article&id=45:bienvenidos-al-sistema-ayd-sec-cha-depto-informatica-fac-ingenieria-unpsjb&catid=34:proyecto-investigacion. [En línea] [Citado el: 15 de febrero de 2013.]

- http://http://www.ing.unp.edu.ar/enlaces_conceptuales/index.php?option=com_content&view=article&id=45:bienv-enidos-al-sistema-ayd-sec-cha-depto-informatica-fac-ingenieria-unpsjb&catid=34:proyecto-investigacion.
32. Proyecto AYD SEC CHA. Informe Final GRAFIFUN. [www.ing.unp.edu.ar/asignaturas/progav/descgrafifun.pdf](http://www.ing.unp.edu.ar/ asignaturas/progav/descgrafifun.pdf). [En línea] [Citado el: 17 de diciembre de 2012.] <http://www.ing.unp.edu.ar/ asignaturas/progav/descgrafifun.pdf>.
33. Macarena Gómez Mármol y Manuel Luna Laynez. *Manual Básico de Octave y Qt octave..* Curso 2010/11.
34. K, Patricio Salinas Caro and Nancy Histchfeld. Tutorial de UML. users.dcc.uchile.cl/~psalinas/uml/introduccion.html. [En línea] [Citado el: 8 de enero de 2013.] <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
35. Booch, James Rumbaugh Ivar Jacoson and Grady. *El Lenguaje de Modelado*. 2000.
36. Monografias | Ingenieria de SoftwareUML. www.monografias.com/trabajos5/insof/insof.shtml. [En línea] Monografias.com S.A., 2011. [Citado el: 8 de enero de 2013.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.
37. Aris Community. www.omg.org/spec/BPMN/1.1. [En línea] 2010. [Citado el: 17 de febrero de 2013.] <http://www.omg.org/spec/BPMN/1.1>.
38. Aguilar, Maria de Jesús Cartelle Cruz y Denys Bárbaro Vega. *Sistema para la planificación quirúrgica ortopédica..* La Habana : s.n., 2011. s n.
39. Programación orientada a objetos. bragasofts.wordpress.com/paradigmas-de-programacion-orientacion-a-objetos/. [En línea] [Citado el: 9 de diciembre de 2012.] <http://bragasofts.wordpress.com/2008/07/01/paradigmas-de-programacion-orientacion-a-objetos/>.
40. Microsoft. Introducción al lenguaje C# y .NET Framework. msdn.microsoft.com/library/z1zx9t92. [En línea] [Citado el: 14 de diciembre de 2012.] <http://msdn.microsoft.com/library/z1zx9t92>..
41. Microsoft. Introducción al lenguaje C# y .Net Framework. msdn.microsoft.com/library/z1zx9t92. [En línea] [Citado el: 14 de enero de 2012.] <http://msdn.microsoft.com/library/z1zx9t92>..
42. Capítulo 2. IAGP 2005/06. Metodologías de desarrollo de software. www.um.es/docencia/barzana/IAGP/lagp2.html. [En línea] [Citado el: 14 de enero de 2013.] (<http://www.um.es/docencia/barzana/IAGP/lagp2.html>)..
43. Metodologías de desarrollo del software. latecladeescape.com/t/Metodolog%C3%ADas+de+desarrollo+del+software. [En línea] La tecla de escape. [Citado el: 13 de enero de 2013.] <http://latecladeescape.com/t/Metodolog%C3%ADas+de+desarrollo+del+software>.
44. *RUP Conferencia de Ingeniería de Software Introducción a la Ingeniería de Software*. 2005-2006.

45. *Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado (RUP)*.
46. Enterprise Architect. www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0. [En línea] [Citado el: 22 de abril de 2008.] <http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0>.
47. Sparx Systems Pty Ltd. www.sparxsystems.es/New/products/ea_features.html. [En línea] Sparx Systems, España. [Citado el: 14 de enero de 2013.] http://www.sparxsystems.es/New/products/ea_features.html.
48. Introducción a Visual Studio. msdn.microsoft.com/es-es/library/6b6b1f4.aspx. [En línea] [Citado el: 14 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/6b6b1f4.aspx>.
49. Microsoft. Información general y conceptual sobre .Net Framework. msdn.microsoft.com/library/zw4w595w.aspx. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/library/zw4w595w.aspx>.
50. Common Language Runtime (CLR). www.devjoker.com/contenidos/catss/89/Common-Language-Runtime-CLR.aspx. [En línea] [Citado el: 11 de enero de 2013.] <http://www.devjoker.com/contenidos/catss/89/Common-Language-Runtime-CLR.aspx>.
51. Información general de la biblioteca de clases de .Net Framework. msdn.microsoft.com/es-es/library/hfa3fa08%28v=vs.80%29.aspx. [En línea] [Citado el: 12 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/hfa3fa08%28v=vs.80%29.aspx>.
52. Funciones Matemáticas. www.monografias.com/trabajos75/funciones-matematicas/funciones-matematicas2.shtml. [En línea] [Citado el: 4 de diciembre de 2012.] <http://www.monografias.com/trabajos75/funciones-matematicas/funciones-matematicas2.shtml>.
53. Knowledge Management Made Simple. www.ofnisystems.com/services/validation/functional-requirements/. [En línea] 1 de agosto de 2001. [Citado el: 31 de marzo de 2013.] <http://www.ofnisystems.com/services/validation/functional-requirements/>.
54. www.architecting.co.uk/presentations/NFRs.pdf. [En línea] [Citado el: 31 de marzo de 2013.] <http://www.architecting.co.uk/presentations/NFRs.pdf>.
55. Kruchten, Philippe. *Architectural Blueprints--The 4+1 View Model of Software Architecture*. s.l. : Institute of Electrical and Electronics Engineers, 1995. IEEE.
56. Marcello Visconti and Hernán Astudillo. Patrones de diseño. www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf. [En línea] [Citado el: 12 de abril de 2013.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.

57. Software Factory. *www.newsoftwarefactory.com/implementacion.htm*. [En línea] [Citado el: 24 de abril de 2013.] <http://www.newsoftwarefactory.com/implementacion.htm>..
58. El arte de modelar. *fineans.usac.edu.gt:8001*. [En línea] [Citado el: 27 de abril de 2013.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>..
59. Diagramas de componente y Despliegue. *www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf*. [En línea] [Citado el: 24 de abril de 2013.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>..
60. Notación y estilos de programación. *www.bitriding.com/articulos/notacion-estilo-programacion.html*. [En línea] [Citado el: 23 de mayo de 2013.] <http://www.bitriding.com/articulos/notacion-estilo-programacion.html>.

BIBLIOGRAFÍA.

- Aedo, Ing. Sergio Fernández.** Solución a Ejercicios de Matemáticas y Ejercicios de Física, Geometría. www.jfinternational.com/funciones-matematicas.html. [En línea] [Citado el: 11 de diciembre de 2012.] <http://www.jfinternational.com/funciones-matematicas.html>.
- Aguilar, María de Jesús Cartelle Cruz y Denys Bárbaro Vega. 2011.** *Sistema para la planificación quirúrgica ortopédica*. La Habana : s.n., 2011. s n.
- 2010.** Aris Community. www.omg.org/spec/BPMN/1.1. [En línea] 2010. [Citado el: 17 de febrero de 2013.] <http://www.omg.org/spec/BPMN/1.1>.
- Astudillo, Marcello Visconti y Hernán.** *Fundamentos de la Ingeniería de Software*. [PDF] Chile : Universidad Técnica Federico Santa María.
- Booch, James Rumbaugh Ivar Jacoson and Grady. 2000.** *El Lenguaje de Modelado*. 2000.
- 2010.** Buenas tareas. www.buenastareas.com/ensayos/Areas-Donde-Se-Aplican-Las-Matematicas/1141084.html. [En línea] noviembre de 2010. [Citado el: 4 de diciembre de 2012.] <http://www.buenastareas.com/ensayos/Areas-Donde-Se-Aplican-Las-Matematicas/1141084.html>.
- 2010.** Buenas Tareas. www.buenastareas.com/ensayos/Graficacion-Par-Computadora/110998.html. [En línea] febrero de 2010. [Citado el: 11 de diciembre de 2012.] <http://www.buenastareas.com/ensayos/Graficacion-Par-Computadora/110998.html>.
- Capítulo 2. IAGP 2005/06. Metodologías de desarrollo de software. www.um.es/docencia/barzana/IAGP/lagp2.html. [En línea] [Citado el: 14 de enero de 2013.] (<http://www.um.es/docencia/barzana/IAGP/lagp2.html>).
- Common Language Runtime (CLR). www.devjoker.com/contenidos/catss/89/Common-Language-Runtime-CLR.aspx. [En línea] [Citado el: 11 de enero de 2013.] <http://www.devjoker.com/contenidos/catss/89/Common-Language-Runtime-CLR.aspx>.
- Conceptualización de un nodo virtual de procesos*. **Y. Gámez, V. Moreno and Y. Martínez. 2009.** La Habana, : Informática, 2009.
- Computación Científica. www.scientific-computing.com/scwmarapr04derive6.html. [En línea] [Citado el: 21 de enero de 2013.] <http://www.scientific-computing.com/scwmarapr04derive6.html>.
- Curso Rápido de Matlab. www.iit.upcomillas.es/palacios/matlab/curso_matlab.pdf. [En línea] [Citado el: 3 de febrero de 2013.] http://www.iit.upcomillas.es/palacios/matlab/curso_matlab.pdf.

Definición de... *definicion.de/fenomeno/*. [En línea] [Citado el: 14 de diciembre de 2012.] <http://definicion.de/fenomeno/>.

Definición.org. *www.definicion.org/informatica*. [En línea] [Citado el: 6 de junio de 2013.] <http://www.definicion.org/informatica>.

Derive 6. *www.chartwellyorke.com/derive.html*. [En línea] [Citado el: 21 de enero de 2013.] <http://www.chartwellyorke.com/derive.html>.

Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado (RUP).

Diagramas de componente y Despliegue. *www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf*. [En línea] [Citado el: 24 de abril de 2013.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.

DISEÑO ARQUITECTONICO. *docs.google.com/document*. [En línea] [Citado el: 21 de abril de 2013.] <https://docs.google.com/document>.

DME. Departamento de Matemática Educativa. *www.matedu.cinvestav.mx/~ccuevas/SoftwareEducativo.htm*. [En línea] [Citado el: 12 de febrero de 2013.] <http://www.matedu.cinvestav.mx/~ccuevas/SoftwareEducativo.htm>.

DR. JUAN GUITERAS GENE. Revista Médica Electrónica. *scielo.sld.cu/scielo.php*. [En línea] [Citado el: 6 de junio de 2013.] http://scielo.sld.cu/scielo.php?pid=S1684-18242009000300012&script=sci_arttext.

El arte de modelar. *fineans.usac.edu.gt:8001*. [En línea] [Citado el: 27 de abril de 2013.] <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.

Enterprise Architect. *www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0*. [En línea] [Citado el: 22 de abril de 2008.] <http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0>.

2010. Entorno Visual de Aprendizaje. *eva.uci.cu/course/view.php?id=170*. [En línea] Universidad de las Ciencias Informáticas, 2010. [Citado el: 7 de enero de 2013.] <http://eva.uci.cu/course/view.php?id=170>.

Funciones Matemáticas. *www.monografias.com/trabajos75/funciones-matematicas/funciones-matematicas2.shtml*. [En línea] [Citado el: 4 de diciembre de 2012.] <http://www.monografias.com/trabajos75/funciones-matematicas/funciones-matematicas2.shtml>.

García, A. *Matemática con Derive*.

—. *Prácticas de matemática con DERIVE*. 1994 : Alfonso García.

GÓMEZ, ENRIQUE HUAPAYA. 2012. *MODELACIÓN USANDO FUNCIÓN CUADRÁTICA: EXPERIMENTOS DE ENSEÑANZA CON ESTUDIANTES DE 5TO DE SECUNDARIA*. PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ. Lima : s.n., 2012. pág. 16, Tesis para optar el grado de Magister en Enseñanza de las Matemáticas.

- Importancia De Las Matematicas En La Carrera De Ingenieria De Sistemas e Informatica. www.buenastareas.com/ensayos/Importancia-De-Las-Matematicas-En-La/2391461.html. [En línea] [Citado el: 20 de marzo de 2013.] <http://www.buenastareas.com/ensayos/Importancia-De-Las-Matematicas-En-La/2391461.html>.
- Información general de la biblioteca de clases de .Net Framework. msdn.microsoft.com/es-es/library/hfa3fa08%28v=vs.80%29.aspx. [En línea] [Citado el: 12 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/hfa3fa08%28v=vs.80%29.aspx>.
- 2005.** Información sobre el programa Derive. www.upv.es/derive/general.htm. [En línea] Asociación de Usuarios de Derive de España, 2005. [Citado el: 7 de enero de 2013.] <http://www.upv.es/derive/general.htm>.
- Informática Educativa en el Currículum de Enseñanza Media. www.eduteka.org/pdfdir/ChileCurriculoMatematicasTics.pdf. [En línea] Red de Asistencia Técnica de Enlaces, Ministerio de Educación de Chile. [Citado el: 20 de febrero de 2013.] <http://www.eduteka.org/pdfdir/ChileCurriculoMatematicasTics.pdf>.
- 2012.** Introducción a la programación con Matlab. mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf. [En línea] Departamento de Física y Matemática Aplicada, 2012. [Citado el: 14 de enero de 2013.] <http://mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf>.
- Introducción a Visual Studio. msdn.microsoft.com/es-es/library/6x6bk1f4.aspx. [En línea] [Citado el: 14 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/6x6bk1f4.aspx>.
- 2008.** Juega y Construye la Matemática. sites.google.com/site/timatematicas/home/asistentes-matematicos. [En línea] José L. Orozco Tróchez, noviembre de 2008. [Citado el: 10 de enero de 2013.] <https://sites.google.com/site/timatematicas/home/asistentes-matematicos>.
- K, Patricio Salinas Caro and Nancy Histchfeld.** Tutorial de UML. users.dcc.uchile.cl/~psalinas/uml/introduccion.html. [En línea] [Citado el: 8 de enero de 2013.] <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
- 2001.** Knowledge Management Made Simple. www.ofnisystems.com/services/validation/functional-requirements/. [En línea] 1 de agosto de 2001. [Citado el: 31 de marzo de 2013.] <http://www.ofnisystems.com/services/validation/functional-requirements/>.
- Kruchten, Philippe. 1995.** *Architectural Blueprints--The 4+1 View Model of Software Architecture*. s.l. : Institute of Electrical and Electronics Engineers, 1995. IEEE.

- Lapiente, María Jesús Lamarca. 2011.** Hipertexto, el nuevo concepto de documento en la cultura de la imagen. *www.hipertexto.info/documentos/interactiv.htm*. [En línea] 5 de mayo de 2011. [Citado el: 12 de diciembre de 2012.] <http://www.hipertexto.info/documentos/interactiv.htm>.
- Las TICs en la educación. *boj.pntic.mec.es/jgomez46/ticedu.htm*. [En línea] [Citado el: 6 de junio de 2013.] <http://boj.pntic.mec.es/jgomez46/ticedu.htm>.
- Macarena Gómez Mármol y Manuel Luna Laynez. Curso 2010/11..** *Manual Básico de Octave y Qt octave..* Curso 2010/11.
- Marcello Visconti and Hernán Astudillo.** Patrones de diseño. *www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf*. [En línea] [Citado el: 12 de abril de 2013.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
- matemática, Departamento de.** *Manual de introducción al uso del programa DERIVE.* s.l. : EII EUX. Memorias | El segundo Congreso Internacional de Enseñanza de la Matemática (CIEM). [En línea] Metodologías de desarrollo del software. *latecladeescape.com/t/Metodolog%C3%ADas+de+desarrollo+del+software*. [En línea] La tecla de escape. [Citado el: 13 de enero de 2013.] <http://latecladeescape.com/t/Metodolog%C3%ADas+de+desarrollo+del+software>.
- Microsoft. Información general y conceptual sobre .Net Framework. *msdn.microsoft.com/library/zw4w595w.aspx*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/library/zw4w595w.aspx>.
- Microsoft. Introducción al lenguaje C# y .NET Framework. *msdn.microsoft.com/library/z1zx9t92*. [En línea] [Citado el: 14 de diciembre de 2012.] <http://msdn.microsoft.com/library/z1zx9t92>.
- Microsoft. Introducción al lenguaje C# y .Net Framework. *msdn.microsoft.com/library/z1zx9t92*. [En línea] [Citado el: 14 de enero de 2012.] <http://msdn.microsoft.com/library/z1zx9t92>.
- 2012.** Misión | Portal de la Universidad de Ciencias Informáticas. *www.uci.cu/mision*. [En línea] Universidad de Ciencias Informáticas, 2012. [Citado el: 6 de diciembre de 2012.] <http://www.uci.cu/mision>.
- 2011.** Monografías | Ingeniería de Software UML. *www.monografias.com/trabajos5/insof/insof.shtml*. [En línea] Monografías.com S.A., 2011. [Citado el: 8 de enero de 2013.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.
- Notación y estilos de programación. *www.bitriding.com/articulos/notacion-estilo-programacion.html*. [En línea] [Citado el: 23 de mayo de 2013.] <http://www.bitriding.com/articulos/notacion-estilo-programacion.html>.
- NUEVAS TECNOLOGIAS. *edutecusac.blogspot.com/2011/03/definiciones-de-tics.html*. [En línea] EDUTEC. [Citado el: 12 de diciembre de 2012.] <http://edutecusac.blogspot.com/2011/03/definiciones-de-tics.html>.

- PI N° 686.** Sistema AyD SEC CHA Depto. Informática- Fac. Ingeniería. http://www.ing.unp.edu.ar/enlaces_conceptuales/index.php?option=com_content&view=article&id=45:bienvenidos-al-sistema-ayd-sec-cha-depto-informatica-fac-ingenieria-unpsjb&catid=34:proyecto-investigacion. [En línea] [Citado el: 15 de febrero de 2013.] http://www.ing.unp.edu.ar/enlaces_conceptuales/index.php?option=com_content&view=article&id=45:bienvenidos-al-sistema-ayd-sec-cha-depto-informatica-fac-ingenieria-unpsjb&catid=34:proyecto-investigacion.
- 2012.** Pregrado | Portal de la Universidad de Ciencias Informáticas. www.uci.cu/pregrado#estudiantes. [En línea] Universidad de Ciencias Informáticas, 2012. [Citado el: 6 de diciembre de 2012.] <http://www.uci.cu/pregrado#estudiantes>.
- 2012.** Pregrado | Portal de la Universidad de Ciencias Informáticas. www.uci.cu/pregrado. [En línea] Universidad de Ciencias Informáticas, 2012. [Citado el: 6 de diciembre de 2012.] <http://www.uci.cu>.
- Programación orientada a objetos. bragasofts.wordpress.com/paradigmas-de-programacion-orientacion-a-objetos/. [En línea] [Citado el: 9 de diciembre de 2012.] <http://bragasofts.wordpress.com/2008/07/01/paradigmas-de-programacion-orientacion-a-objetos/>.
- Proyecto AYD SEC CHA.** Informe Final GRAFIFUN. www.ing.unp.edu.ar/asignaturas/progav/descgrafifun.pdf. [En línea] [Citado el: 17 de diciembre de 2012.] <http://www.ing.unp.edu.ar/asignaturas/progav/descgrafifun.pdf>.
- Revolución informática. sites.google.com/site/revotica0622/primera-computadora. [En línea] [Citado el: 6 de junio de 2013.] <https://sites.google.com/site/revotica0622/primera-computadora>.
- Richard, Lic. Cesar Nicolas. 2007.** *Propuesta didáctica para el aprendizaje de la matemática numérica con la utilización de Entorno Virtual de Aprendizaje*. Ciencias Básicas, CUJAE. La Habana : s.n., 2007. pág. 43, Tesis de Maestría.
- Richard, Lic. Cesar Nicolás. 2007.** *Propuesta didáctica para el aprendizaje de la matemática numérica con la utilización de Entorno Virtual de Aprendizaje*. La Habana : s.n., 2007. pág. 47, Tesis de Maestría.
- 2005-2006.** RUP Conferencia de Ingeniería de Software Introducción a la Ingeniería de Software. 2005-2006.
- Salgado, Zully Lucía Alfonzo. 2012.** *DIDÁCTICA DE LAS FUNCIONES LINEALES Y CUADRÁTICAS ASISTIDA CON COMPUTADORA*. Cumaná : CEDUT- Las Tunas y CEdeG-Granma, CUBA, 2012.
- Silva, Erika Cecilia Parra.** USO DE HERRAMIENTA WEB EN EL APRENDIZAJE DE FUNCIONES. revistas.upel.edu.ve/index.php/educare/article/view/59/59. [En línea] [Citado el: 15 de febrero de 2013.] <http://revistas.upel.edu.ve/index.php/educare/article/view/59/59>.

- 2012.** Softonic Descargas. *derive.softonic.com*. [En línea] Softonic, 2012. [Citado el: 17 de diciembre de 2012.] <http://derive.softonic.com>.
- Software educativos en Cuba. *www.ecured.cu/index.php/Software_educativos_en_Cuba#Tipos_de_software_educativos*. [En línea] [Citado el: 6 de junio de 2013.] http://www.ecured.cu/index.php/Software_educativos_en_Cuba#Tipos_de_software_educativos.
- Software Factory. *www.newsoftwarefactory.com/implementacion.htm*. [En línea] [Citado el: 24 de abril de 2013.] <http://www.newsoftwarefactory.com/implementacion.htm>.
- Sosa, José B. Rodríguez. 2003.** *Una propuesta metodológica para la utilización de las tecnologías de la información y las comunicaciones en el proceso de enseñanza aprendizaje de las funciones matemáticas*. La Habana : s.n., 2003.
- Sparx Systems Pty Ltd. *www.sparxsystems.es/New/products/ea_features.html*. [En línea] Sparx Systems, España. [Citado el: 14 de enero de 2013.] http://www.sparxsystems.es/New/products/ea_features.html.
- System, Eastern Software. 2008.** *Arquitectura de tres capas*. 2008.
- Tangent LLC.** Matemáticas: Integrando Tecnologías para una mejor Formación. *vumatematicas.wikispaces.com/Informe+sobre+Winplot*. [En línea] [Citado el: 2 de febrero de 2013.] <http://vumatematicas.wikispaces.com/Informe+sobre+Winplot>.
- Tecnología al Instante. *www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=2060*. [En línea] Estudios Pro Nexus Radical. [Citado el: 12 de febrero de 2013.] http://www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=2060.
- Tutorial sobre Matlab Capítulo 2. *catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo2.pdf*. [En línea] [Citado el: 10 de febrero de 2013.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo2.pdf.
- 2012.** Universia | España: noticias. *noticias.universia.es/ciencia-nn-tt/noticia/2005/09/05/606111/importancia-matematicas.html*. [En línea] Universia España., 2012. [Citado el: 6 de diciembre de 2012.] <http://noticias.universia.es/ciencia-nn-tt/noticia/2005/09/05/606111/importancia-matematicas.html>.
- Universidad Nacional Experimental del Táchira.** Competencias del Ingeniero en Informática | Sitio de la Universidad Nacional Experimental del Táchira. *admission.unet.edu.ve/pages/main.php?id=8001*. [En línea] [Citado el: 12 de febrero de 2013.] <http://admission.unet.edu.ve/pages/main.php?id=8001>.

- 2012.** winplot. softonic. *winplot.softonic.com/*. [En línea] softonic, 2012. [Citado el: 14 de diciembre de 2012.] <http://winplot.softonic.com/>.
- www.architecting.co.uk/presentations/NFRs.pdf. [En línea] [Citado el: 31 de marzo de 2013.] <http://www.architecting.co.uk/presentations/NFRs.pdf>.

ANEXOS.

Anexo 1 Prefijos de los requerimientos no funcionales por categoría.

Categoría	Prefijo	Ejemplo
Usabilidad	RNU	RNU 1. Requisito de usabilidad A RNU 2. Requisito de usabilidad B
Diseño e Implementación	RNDI	RNDI 1. Requisito de diseño e implementación
Legal	RNL	RNL 1. Requisito legal
Fiabilidad	RNF	RNF 1. Requisito de fiabilidad
Seguridad	RNS	RNS 1. Requisito de seguridad
Eficiencia	RNE	RNE 1. Requisito de eficiencia
Funcionamiento	RNFO	RNFO 1. Requisito de funcionamiento
Soporte	RNSO	RNSO 1. Requisito de soporte
Interfaz de usuario	RNIU	RNIU 1. Requisito de interfaz de usuario
Interconexión	RNI	RNI 1. Requisito de interconexión
Componentes Comprados	RNCC	RNCC 1. Requisito de componentes comprados
Licencia	RNLI	RNLI 1. Requisito de licencia

Anexo 2 Descripción de los casos de uso.

- Caso de uso #1.1 – Graficar rectas y parábolas.

Objetivo	El objetivo que persigue dicho CU es mostrar la gráfica que le corresponde a la ecuación entrada por parámetro.	
Actores	Usuario (Inicia).	
Resumen	El caso de uso inicia cuando el usuario hace clic en el botón “Graficar” o presiona la tecla “Enter” y el sistema representa la gráfica que se corresponde con la ecuación entrada por parámetros.	
Complejidad	Alta.	
Prioridad	Crítica.	
Referencias	RF 2	
Precondiciones	El usuario introduce la ecuación que va a graficar.	
Postcondiciones	Permite graficar la ecuación entrada por parámetros.	
Flujo de eventos		
Flujo básico “Graficar ecuación.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Graficar” o presionar la tecla “Enter”. 2. El sistema muestra la gráfica de la ecuación. 3. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	Insertar ecuación
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes	No aplicable	

- Caso de uso #1.2 – Mostrar contenidos teóricos de las ecuaciones lineales y cuadráticas

Objetivo	El objetivo que persigue el actor al realizar este CU es mostrar vistas con definiciones, teoremas y conceptos sobre ecuaciones lineales y cuadráticas, el cual podrá consultar para evacuar dudas.	
Actores	Usuario (Inicia).	
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Contenidos teóricos”.	
Complejidad	Alta.	
Media.	Crítica.	
Referencias	RF 3.	
Precondiciones	El usuario ha encontrado el fichero con contenidos teóricos relacionado con las ecuaciones lineales y cuadráticas.	
Postcondiciones	Permite mostrar ficheros con contenidos teóricos.	
Flujo de eventos		
Flujo básico “Mostrar aspectos teóricos de las ecuaciones lineales y cuadráticas.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Contenidos teóricos”. 2. El sistema muestra un fichero con contenidos. 3. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes	No aplicable	

- Caso de uso #1.3 – Representar rectas y parábolas en el plano.

Objetivo	El objetivo que persigue este CU es representar de forma manual rectas y parábolas en el plano coordenado
Actores	Usuario (Inicia).

Resumen	El caso de uso inicia cuando el usuario realiza la representación gráfica de forma manual.	
Complejidad	Alta.	
Prioridad	Crítica.	
Referencias	RF 4.	
Precondiciones	El usuario ha seleccionado la opción “Graficar y Revisar” he introducido la ecuación.	
Postcondiciones	Permite representar rectas y parábolas en el plano de forma manual.	
Flujo de eventos		
Flujo básico “Trazar la recta o parábola.”		
<ol style="list-style-type: none"> 1. El usuario traza la recta o parábola que considera que corresponde con la ecuación entrada por parámetros (utilizando el botón que lleva por nombre “lápiz”). 2. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	Insertar ecuación.
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.4 – Editar configuración.

Objetivo	El objetivo que persigue dicho CU es permitir modificar los campos que se corresponden con el sistema coordinado
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Configuración” y modifica los campos.
Complejidad	Medio.

Prioridad	Medio.	
Referencias	RF 9.1, RF 9.2, RF 9.3.	
Precondiciones	El usuario ha seleccionado un sistema coordinado.	
Postcondiciones	Permite seleccionar la configuración.	
Flujo de eventos		
Flujo básico “Editar configuración.”		
Escenario “Seleccionar la Configuración.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Configuración”. 2. El sistema muestra una ventana que lleva por nombre Configuración. 3. El usuario selecciona la configuración que desea (manual o predeterminada). 4. El sistema realiza los cambios realizados en el plano coordinado. 5. Termina el caso de uso. 		
Escenario “Seleccionar la vista.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Configuración”. 2. El sistema muestra una ventana que lleva por nombre Configuración. 3. El usuario selecciona la configuración manual y llena los campos mostrados. 4. El sistema realiza los cambios realizados en el plano coordinado. 5. Termina el caso de uso. 		
Escenario “Cambiar idioma.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Lenguaje”. 2. El sistema muestra los lenguajes que puedes escoger (Español o Inglés). 3. El usuario selecciona el lenguaje. 4. El sistema realiza los cambios realizados por el usuario. 5. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable

Asuntos pendientes	No aplicable
---------------------------	--------------

- Caso de uso #1.5 – Eliminar.

Objetivo	El objetivo que persigue este CU es eliminar las gráficas realizadas (gráfica del usuario o gráfica del sistema).
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Eliminar gráfica del sistema” o “Eliminar gráfica del usuario”.
Complejidad	Medio.
Prioridad	Medio.
Referencias	RF 11.1, RF 11.2, RF 10.
Precondiciones	Se ha realizado la representación gráfica de una función.
Postcondiciones	Permite eliminar la gráfica del usuario. Permite eliminar la gráfica del sistema.
Flujo de eventos	
Flujo básico “Eliminar.”	
Escenario “Eliminar gráfica del usuario.”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Eliminar gráfica del usuario”. 2. El sistema borra en el plano coordenado la gráfica realizada por el usuario. 3. Termina el caso de uso. 	
Escenario “Eliminar gráfica del sistema.”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Eliminar gráfica del sistema”. 2. El sistema borra en el plano coordenado la gráfica realizada por el sistema. 3. Termina el caso de uso. 	
Escenario “Limpiar”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Todo”. 2. El sistema borra en el plano coordenado las gráficas realizadas. 	

3. Termina el caso de uso.		
Flujos alternos		
Relaciones	CU Incluidos	Graficar rectas y parábolas. Representar rectas y parábolas en el plano.
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.6 – Guardar.

Objetivo	El objetivo que persigue dicho CU es permitir guardar la interfaz en la cual se está trabajando como una imagen o fichero. También brinda la opción de guardar los cambios efectuados en el archivo abierto sobrescribiendo su última versión. Este comando permite actualizar el archivo activo en cualquier momento.
Actores	Usuario (Inicia).
Resumen	El caso de uso inicia cuando el usuario selecciona la opción “Guardar” o “Guardar como”.
Complejidad	Medio.
Prioridad	Medio.
Referencias	RF 6.1, RF 6.2.
Precondiciones	Se ha realizado la representación gráfica de una función.
Postcondiciones	Permitir guardar la interfaz en la cual se está trabajando como una imagen o fichero y guardar los cambios efectuados.
Flujo de eventos	
Flujo básico “Guardar.”	
Escenario “Guardar como”	

<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Guardar como”. 2. El sistema muestra una ventana donde puedes seleccionar la dirección donde desea guardar la interfaz en la cual se está trabajando (como una imagen o fichero). 3. El usuario selecciona la dirección donde desea guardar la imagen, lo nombra y hace clic en el botón “Guardar”. 4. El sistema lo guarda en el lugar seleccionado por el usuario. 5. Termina el caso de uso. 		
Escenario “Guardar.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Guardar.” 2. El sistema guardar los cambios efectuados en el archivo abierto sobrescribiendo su última versión (Este comando permite actualizar el archivo activo en cualquier momento). 3. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	Graficar rectas y parábolas. Representar rectas y parábolas en el plano.
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.7 – Revisar.

Objetivo	El objetivo que persigue el actor al realizar este CU es mostrar los resultados de la revisión.
Actores	Usuario (Inicia).
Resumen	Consiste en mostrar los resultados de la revisión donde realiza una comparación de la gráfica realizada por el usuario con la que realice el sistema emitiendo según el color de la gráfica una evaluación.

Complejidad	Media.	
Prioridad	Crítica.	
Referencias	RF 5.	
Precondiciones	El usuario ha representado en el plano coordenado la gráfica que considere que se corresponde con la ecuación entrada por parámetro.	
Postcondiciones	Permite realizar la revisión correctamente.	
Flujo de eventos		
Flujo básico “Revisar.”		
<ol style="list-style-type: none"> 1. El usuario selecciona el botón “Revisar” que se encuentra en el formulario “Graficar y Revisar”. 2. El sistema realiza la revisión comparando la gráfica realizada por el usuario con la que realice el sistema emitiendo según el color de la gráfica una evaluación y lo muestra en el formulario “Principal”. 3. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	Representar rectas y parábolas en el plano.
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.8 – Evaluar habilidades del estudiante para la representación gráfica de funciones.

Objetivo	El objetivo que persigue el actor al realizar este CU es evaluar a los estudiantes mediante un conjunto de ejercicios en el cual se podrá avanzar de manera gestionada.
-----------------	---

Actores	Usuario (Inicia).	
Resumen	Consiste en seleccionar la opción “Ejercitación” y muestra un conjunto de ejercicios en el cual se podrá avanzar de manera gestionada.	
Complejidad	Alta.	
Prioridad	Crítica.	
Referencias	RF 12	
Precondiciones	El usuario ha encontrado el fichero con los ejercicios.	
Postcondiciones	Permite evaluar las habilidades del estudiante.	
Flujo de eventos		
Flujo básico “Evaluar habilidades del estudiante para la representación gráfica de funciones.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Ejercitación”. 2. El sistema muestra un conjunto de ejercicios. 3. El usuario realiza los ejercicios de forma ordenada (de uno en uno). 4. El sistema evalúa el ejercicio y si esta correcto pasa para el otro nivel. 5. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	Representar rectas y parábolas en el plano.
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.9 – Abrir ficheros.

Objetivo	El objetivo que persigue el actor al realizar este CU es abrir un sistema de coordenadas ya guardado como archivo.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Abrir”.

Complejidad	Medio.	
Prioridad	Medio.	
Referencias	RF 8	
Precondiciones	Tiene que existir el fichero.	
Postcondiciones	Permite abrir los ficheros.	
Flujo de eventos		
Flujo básico “Abrir ficheros.”		
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Abrir” o presiona las teclas Ctrl + A o el botón que lleva el mismo nombre que se encuentra en la parte superior de la aplicación. 2. El sistema muestra una ventana donde busca el lugar donde está el proyecto que desea abrir. 3. El usuario selecciona el proyecto y da clic en el botón “abrir”. 4. El sistema abre el fichero seleccionado. 5. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes	No aplicable	

- Caso de uso#1.10– Modificar escala del eje coordenado.

Objetivo	El objetivo que persigue este CU es permitir aumentar o disminuir la escaladel eje coordenado.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Zoom”.
Complejidad	Baja.
Prioridad	Baja.

Referencias	RF 7.1, RF 7.2	
Precondiciones	Tiene que existir un sistema coordenado.	
Postcondiciones	Permite aumentar o disminuir la escala del eje coordenado.	
Flujo de eventos		
Flujo básico “Modificar escala del eje coordenado.”		
Escenario “Aumentar escala del eje coordenado.”		
	<ol style="list-style-type: none"> 1. El usuario selecciona el botón “Zoom (+)” que se encuentra en la parte superior de la aplicación. 2. El sistema aumenta el tamaño original del eje coordenado. 3. Termina el caso de uso. 	
Escenario “Disminuir escala del eje coordenado.”		
	<ol style="list-style-type: none"> 1. El usuario selecciona el botón “Zoom (-)” que se encuentra en la parte superior de la aplicación. 2. El sistema disminuye el tamaño original del eje coordenado. 3. Termina el caso de uso. 	
Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.11 – Mover sistema coordenado.

Objetivo	El objetivo que persigue este CU es mover el sistema coordenado hacia cualquier lugar de la pantalla de la aplicación para lograr una mejor visualización de la gráfica.
Actores	Usuario (Inicia).

Resumen	Consiste en seleccionar el botón “Mover” e ir al sistema coordenado y desplazarlo hacia el lugar que se desee.	
Complejidad	Baja.	
Prioridad	Baja.	
Referencias	RF 14.	
Precondiciones	El usuario desea cambiar la posición del sistema coordenado.	
Postcondiciones	Permitemover el sistema coordenado.	
Flujo de eventos		
Flujo básico “Mover sistema coordenado.”		
<ol style="list-style-type: none"> 1. El usuario selecciona el botón “Mover” y da clic en el sistema coordenado donde lo desplaza hacia el lugar donde que desea. 2. El sistema realiza los cambios. 3. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.12 – Mostrar propiedades de la ecuación.

Objetivo	El objetivo que persigue este CU es mostrar las propiedades de la gráfica representada en el plano coordenado.
Actores	Usuario (Inicia).
Resumen	Consiste en mostrar en el lado derecho de la aplicación las propiedades de la gráfica representada en el plano coordenado.
Complejidad	Medio.
Prioridad	Medio.

Referencias	RF 15.	
Precondiciones	El usuario ha realizado la representación gráfica de la función.	
Postcondiciones	Permite mostrar las propiedades de la gráfica.	
Flujo de eventos		
Flujo básico “Mostrar propiedades de la ecuación.”		
<ol style="list-style-type: none"> 1. El sistema muestra las propiedades de la gráfica representada en el plano coordenado, en el lado derecho de la aplicación del formulario principal. 2. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales no	No aplicable	
Asuntos pendientes	No aplicable	

- Caso de uso #1.13 – Insertar ecuación.

Objetivo	El objetivo que persigue dicho CU es permitir al usuario introducir la ecuación de la recta o parábola que desea graficar.
Actores	Usuario (Inicia).
Resumen	Consiste en introducir la ecuación de la recta o parábola que desea graficar.
Complejidad	Alta.
Prioridad	Crítica.
Referencias	RF 1, RF 16.
Precondiciones	El usuario ha seleccionado la función.
Postcondiciones	El sistema permite introducir la función.
Flujo de eventos	

Flujo básico “Insertar ecuación.”		
<ol style="list-style-type: none"> 1. El usuario introduce la ecuación. 2. El sistema verifica que la ecuación entrada sea de primer grado o de segundo grado en dependencia si es una recta o parábola (recta tiene que ser de primer grado y si es una parábola de segundo grado). 3. Termina el caso de uso. 		
Flujos alternos		
Relaciones	CU Incluidos	Validar función.
	CU Extendidos	No aplicable
Requisitos funcionales	no	No aplicable
Asuntos pendientes		No aplicable

- Caso de uso #1.14– Salir.

Objetivo	El objetivo que persigue este CU es permitir al usuario salir de la aplicación.
Actores	Usuario (Inicia).
Resumen	Consiste en seleccionar la opción “Salir”.
Complejidad	Baja.
Prioridad	Baja.
Referencias	RF 17.
Precondiciones	Tiene que haber abierto la aplicación.
Postcondiciones	Permite salir de la aplicación.
Flujo de eventos	
Flujo básico “Salir.”	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Salir” o presiona las teclas Alt + F4. 2. El sistema cierra la aplicación. 3. Termina el caso de uso. 	

Flujos alternos		
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos funcionales no	No aplicable	
Asuntos pendientes	No aplicable	

Anexo 3 Diagramas de clases del diseño.

Diagrama de Clases del CU Abrir ficheros.

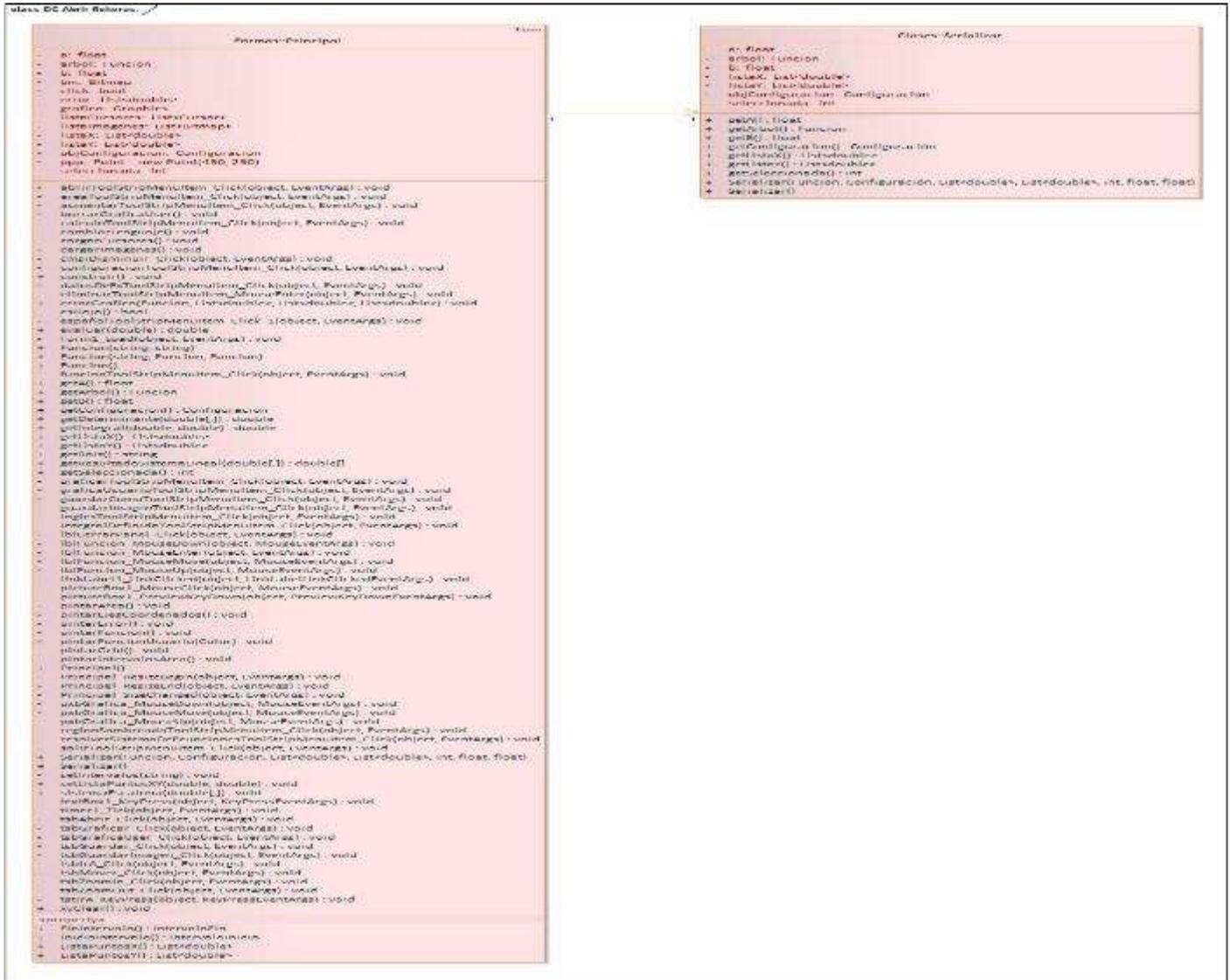


Diagrama de Clases del CU Mostrar contenidos teóricos de las ecuaciones lineales y cuadráticas.



Diagrama de Clases del CU Editar configuración.

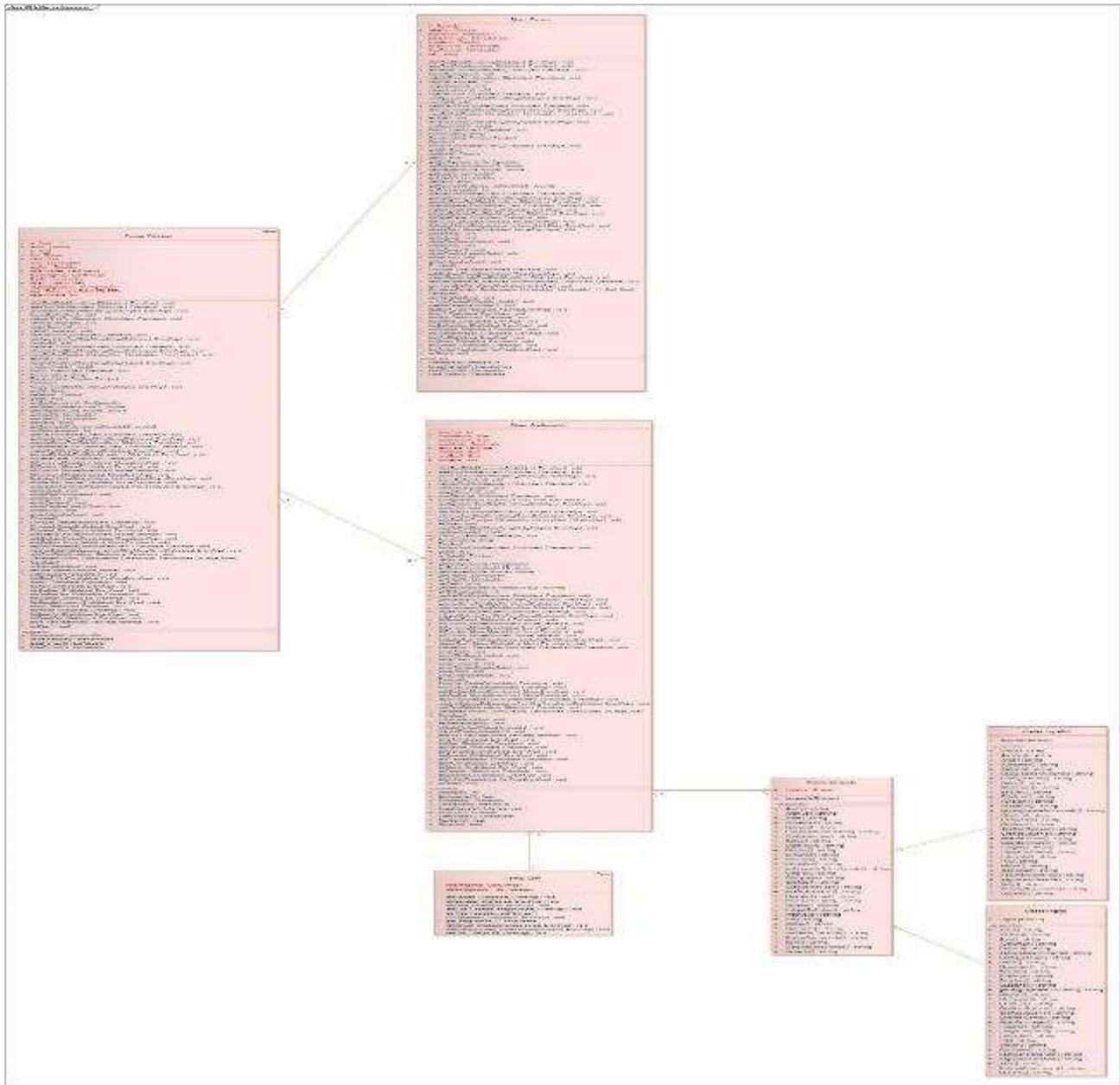
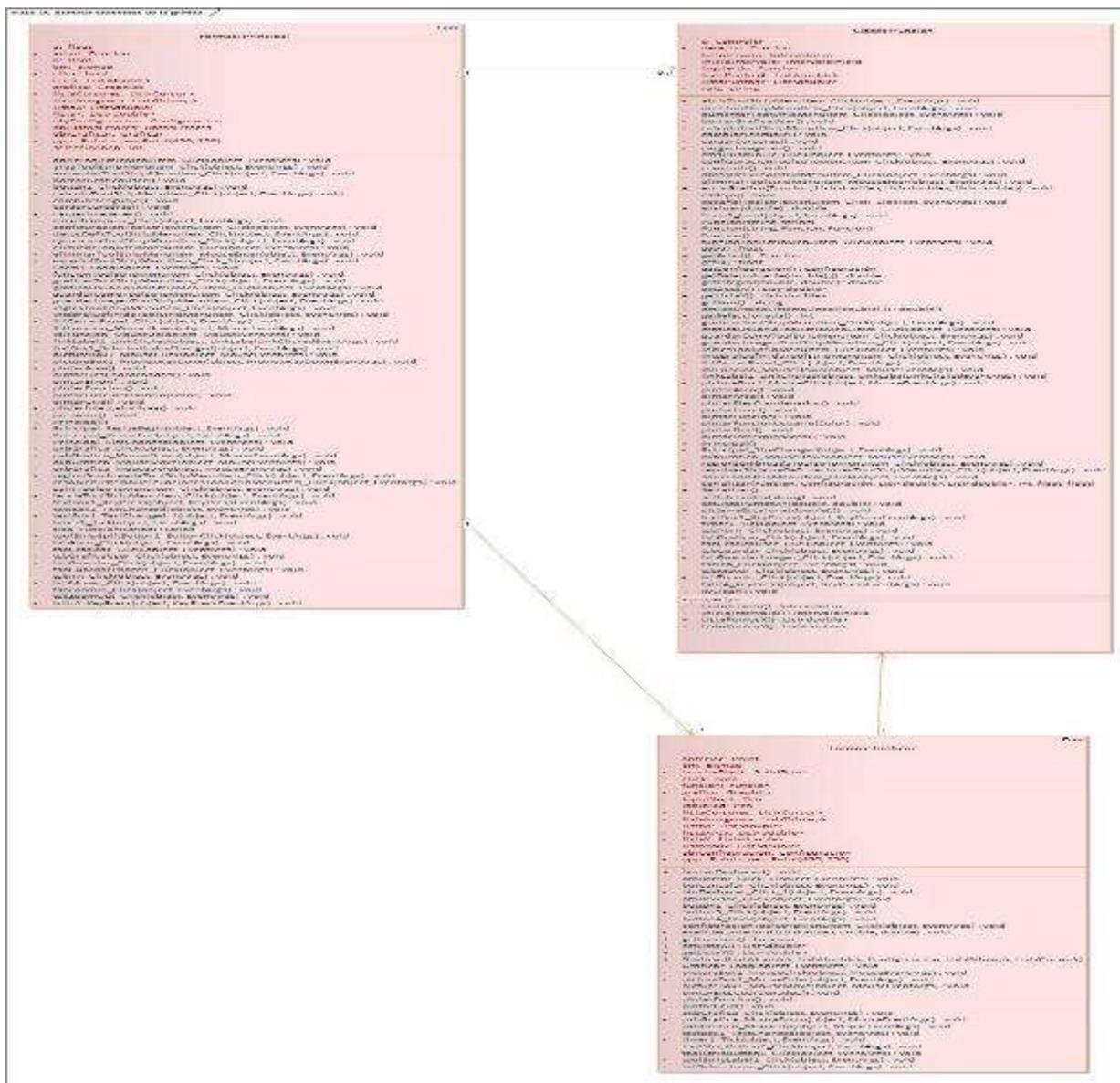


Diagrama de Clases del CU Eliminar.



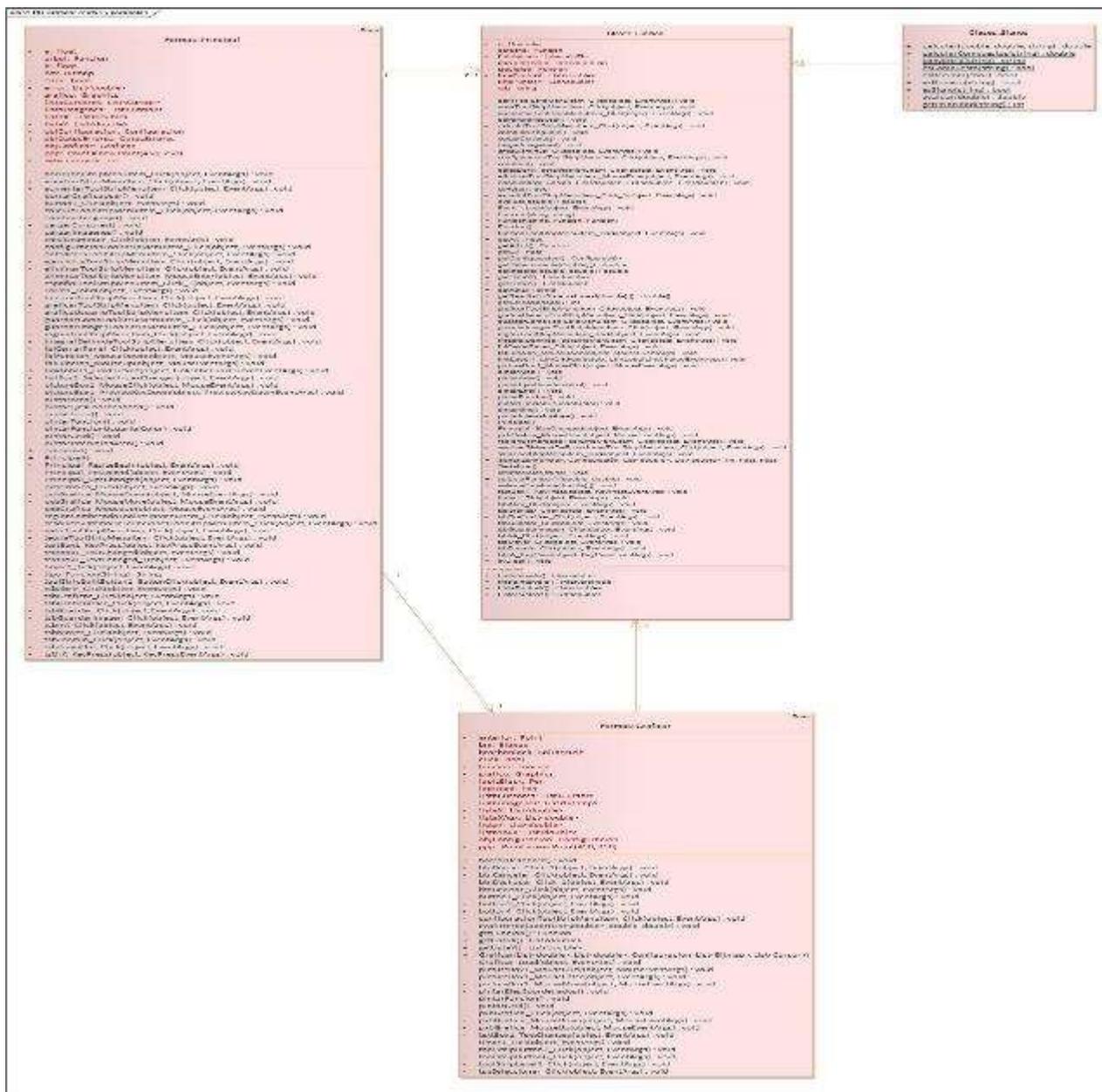


Diagrama de Clases del CU Guardar.

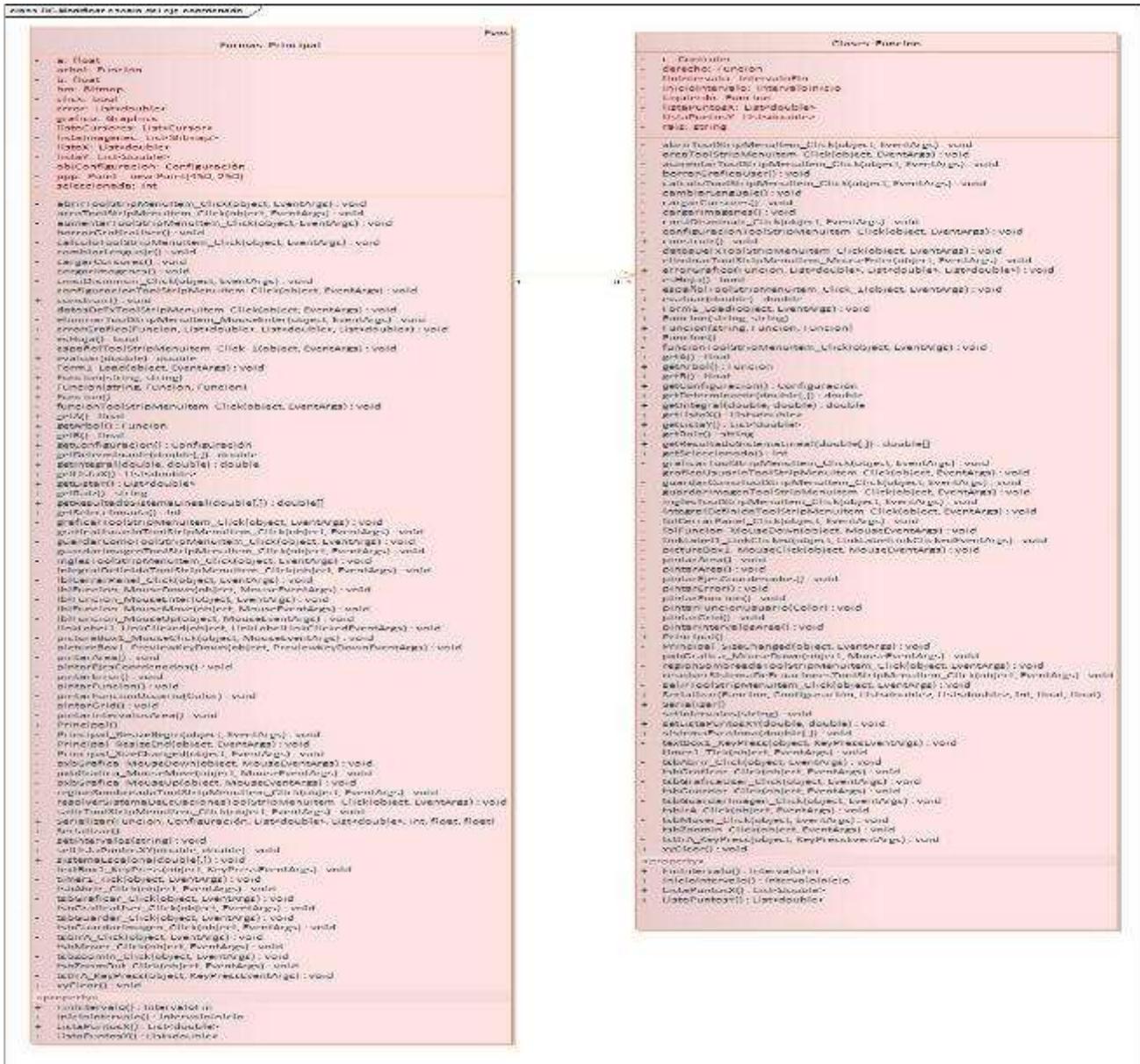


Diagrama de Clases del CU Mostrando propiedades de la ecuación.

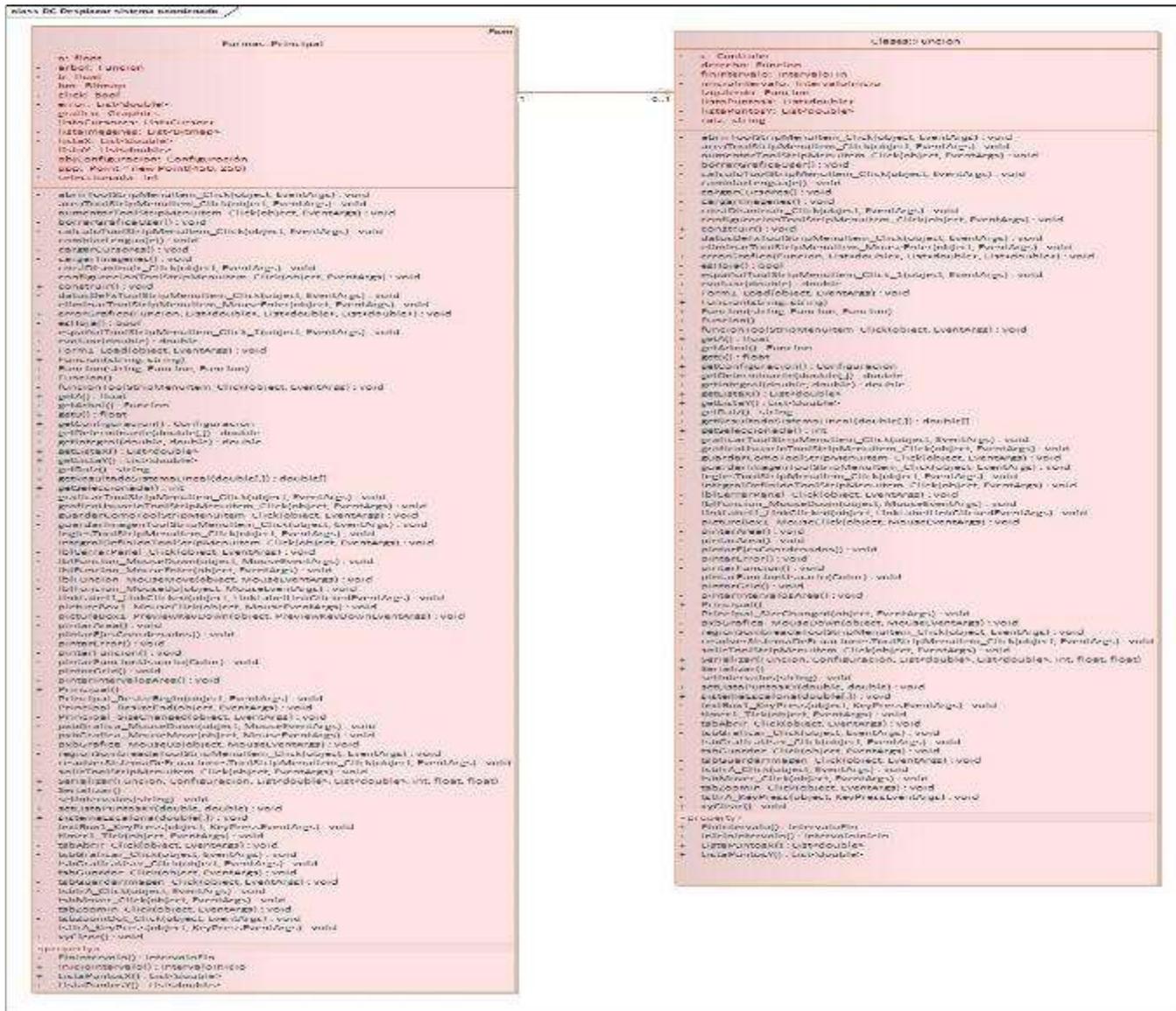


Diagrama de Clases del CU Salir.

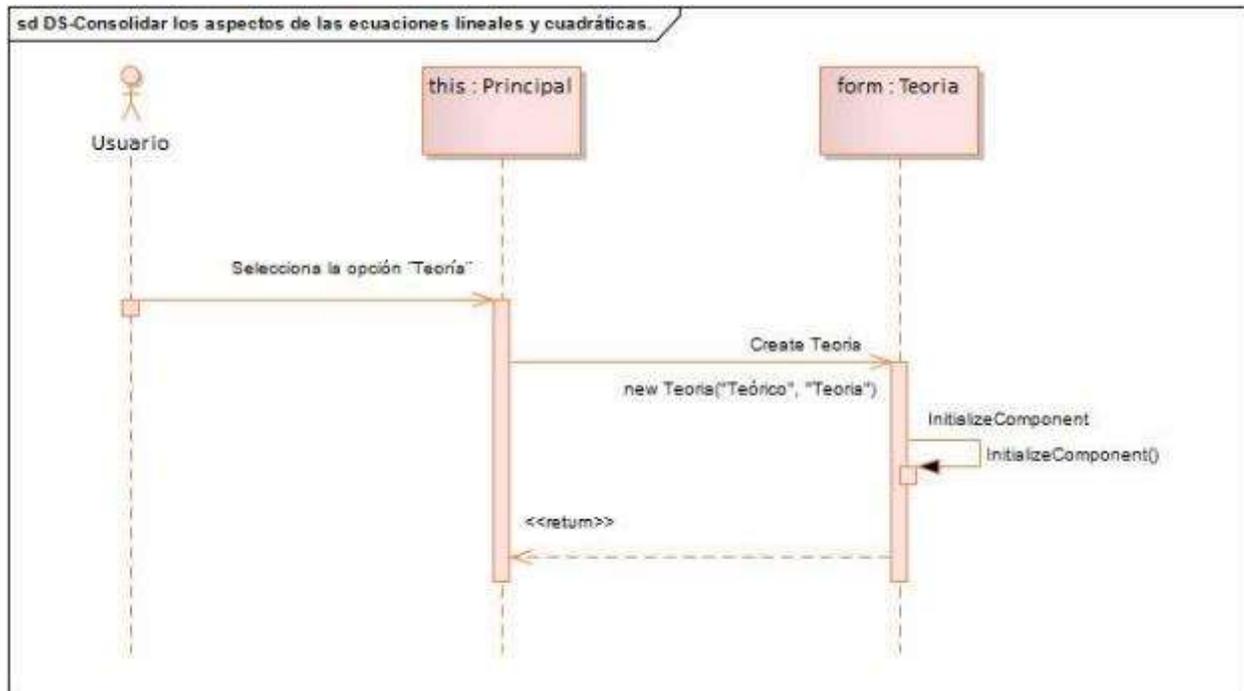


Diagrama de secuencia del CU Abrir ficheros.

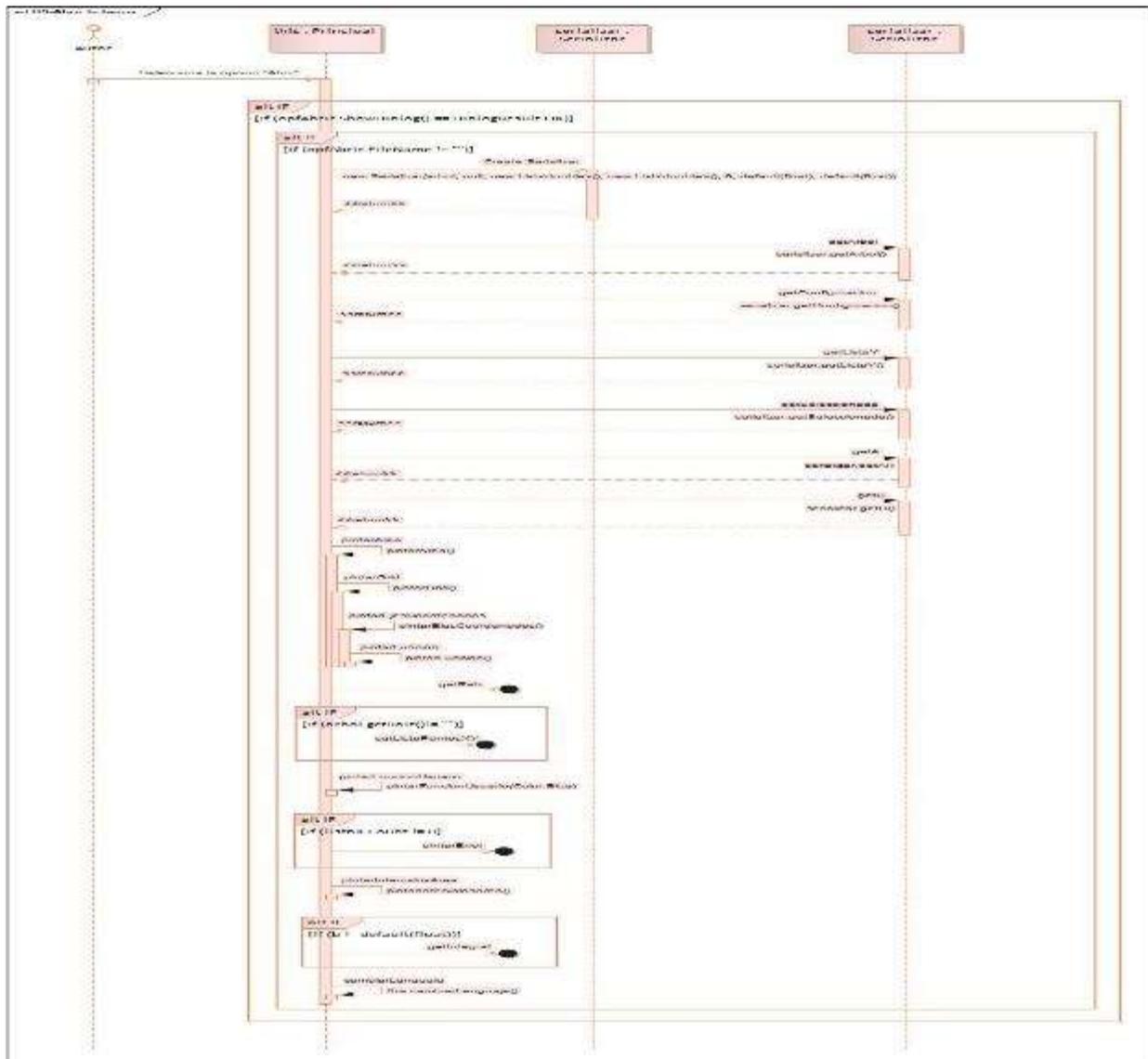
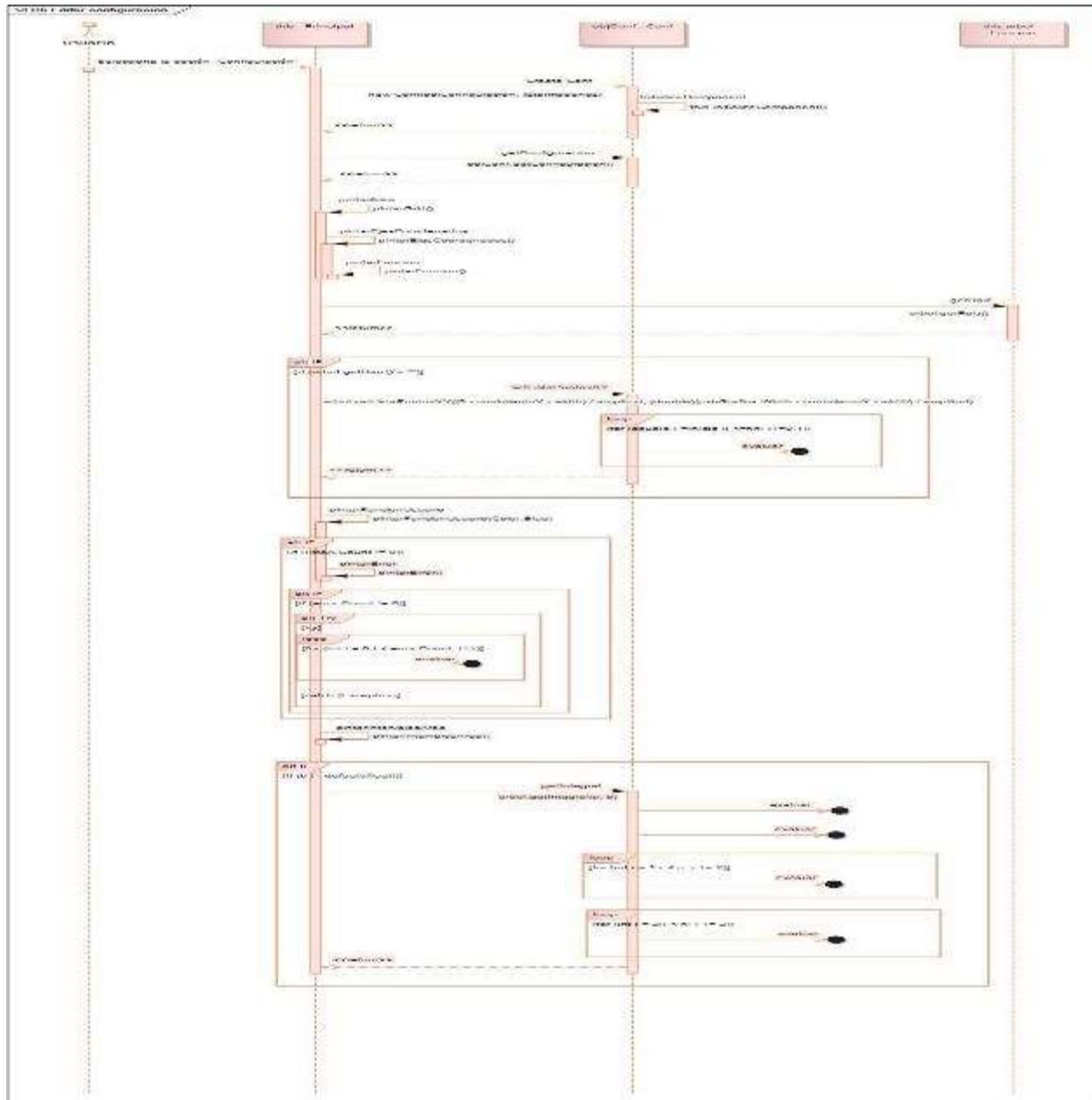


Diagrama de secuencia del CU Editar configuración.



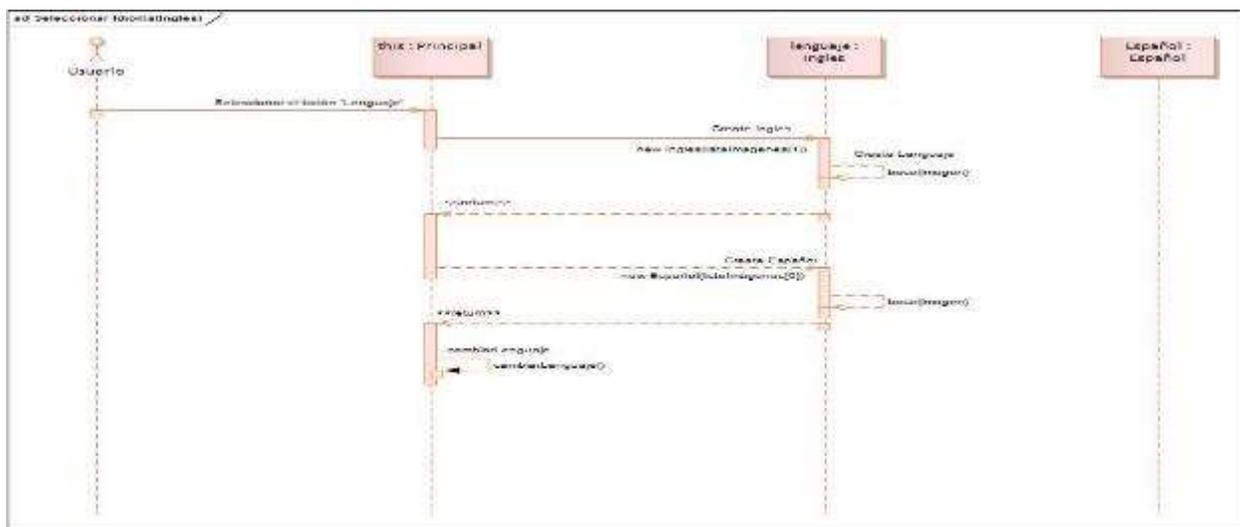
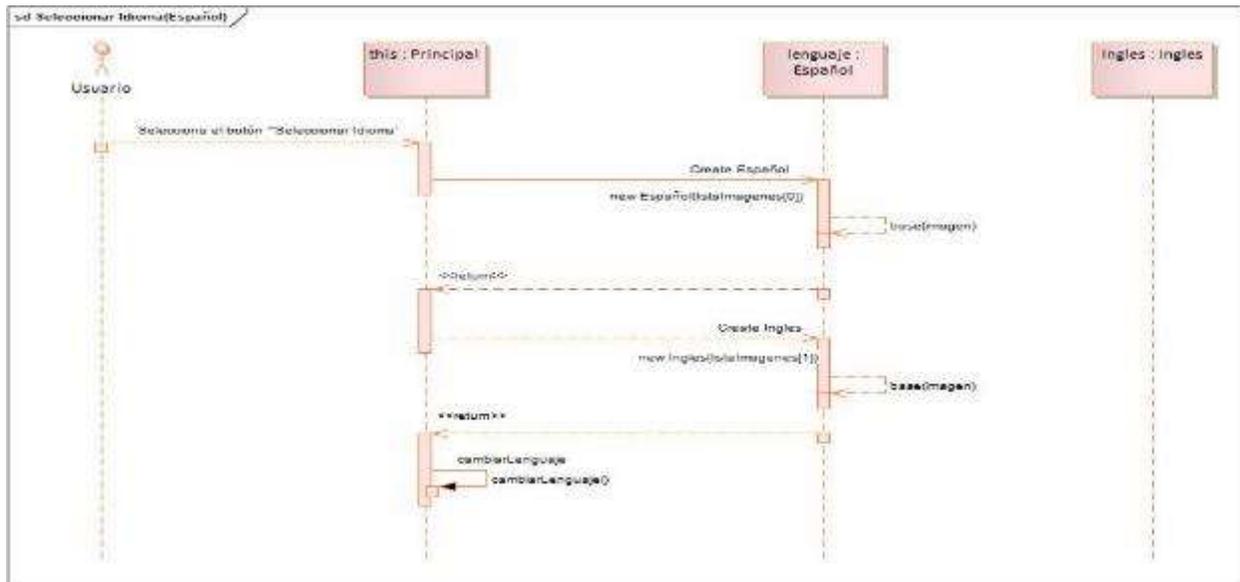


Diagrama de secuencia del CU Eliminar.

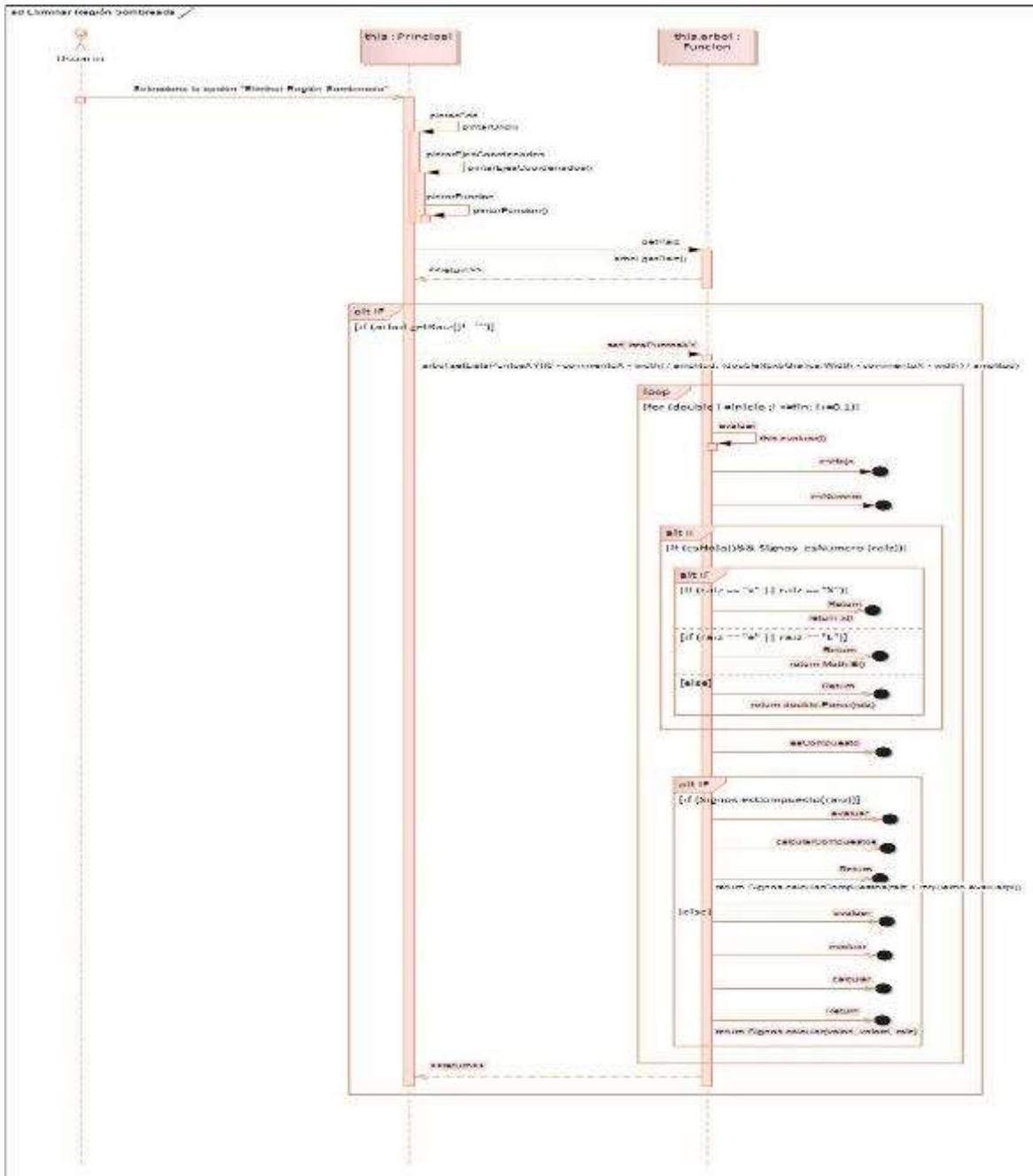


Diagrama de secuencia del CU Evaluar habilidades del estudiante para la representación gráfica de funciones.

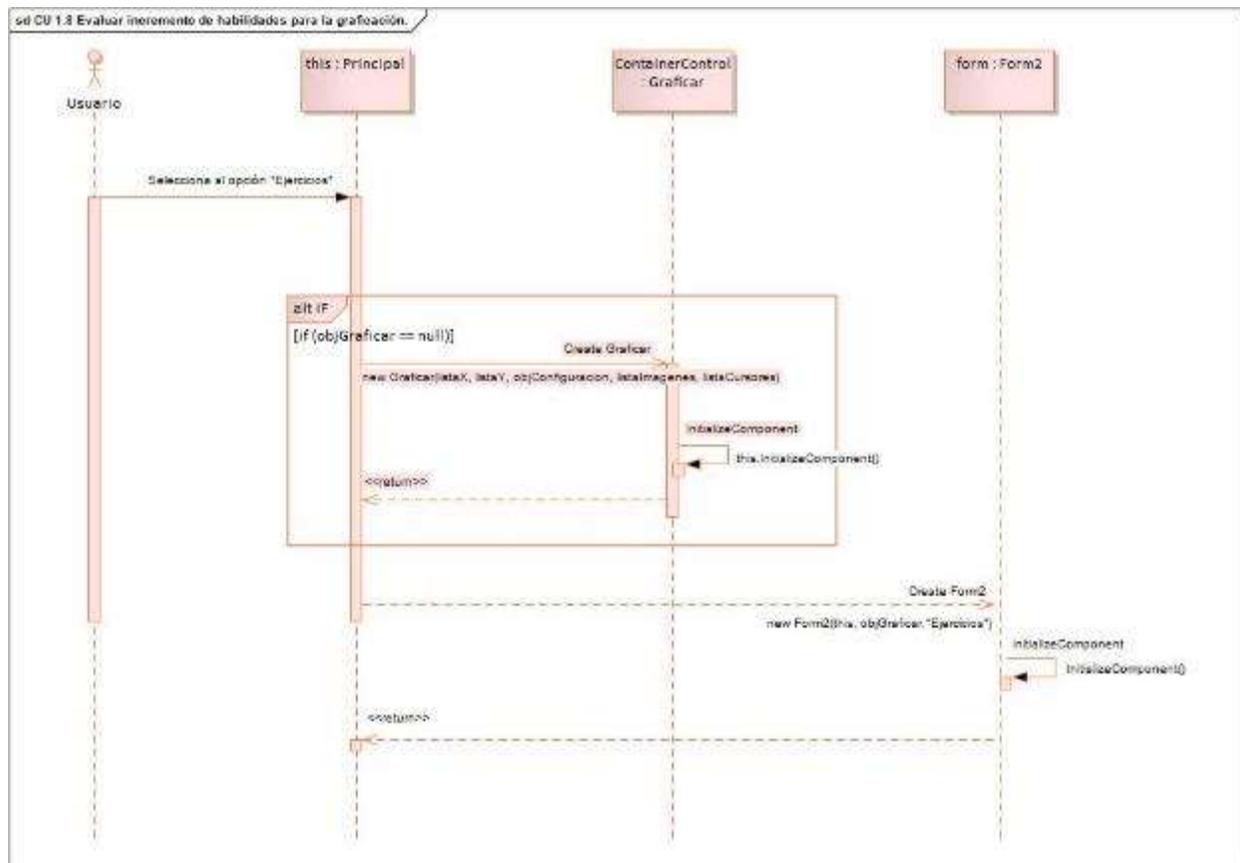


Diagrama de secuencia del CU Graficar rectas y parábolas.

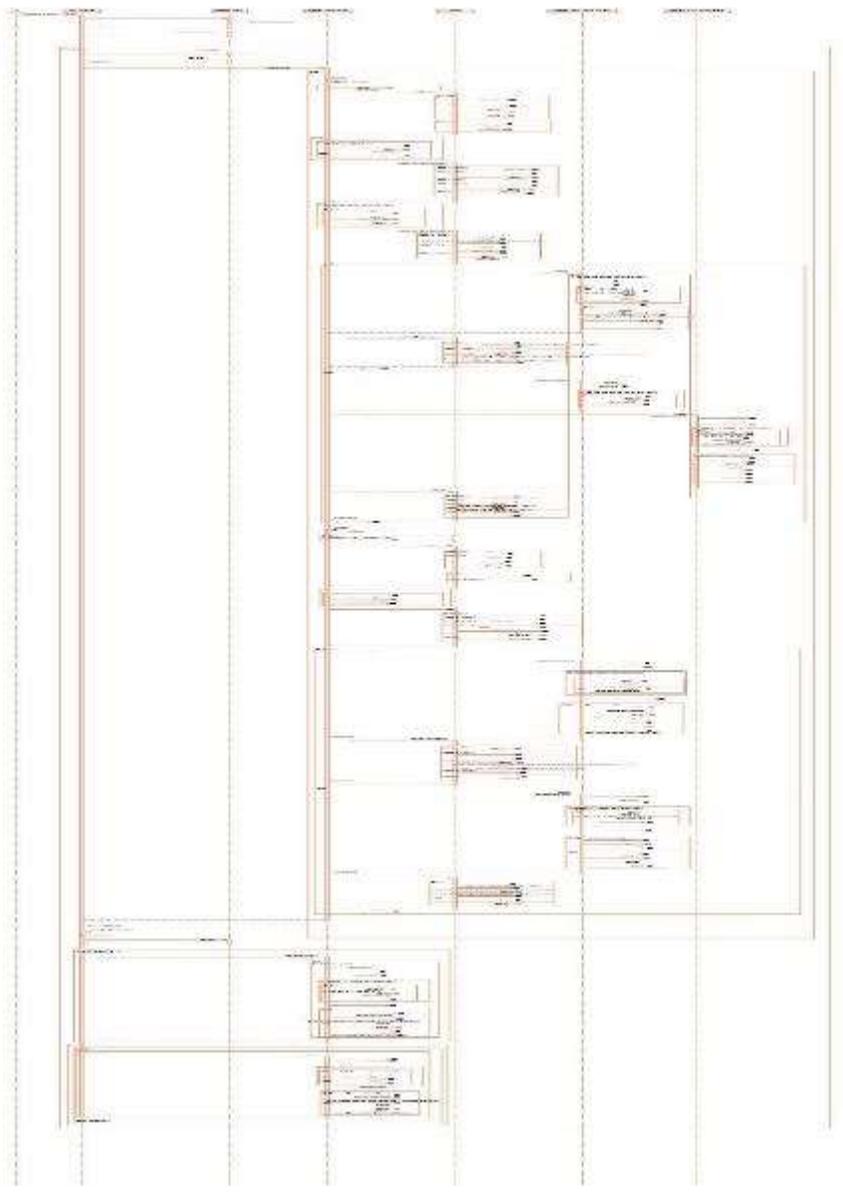
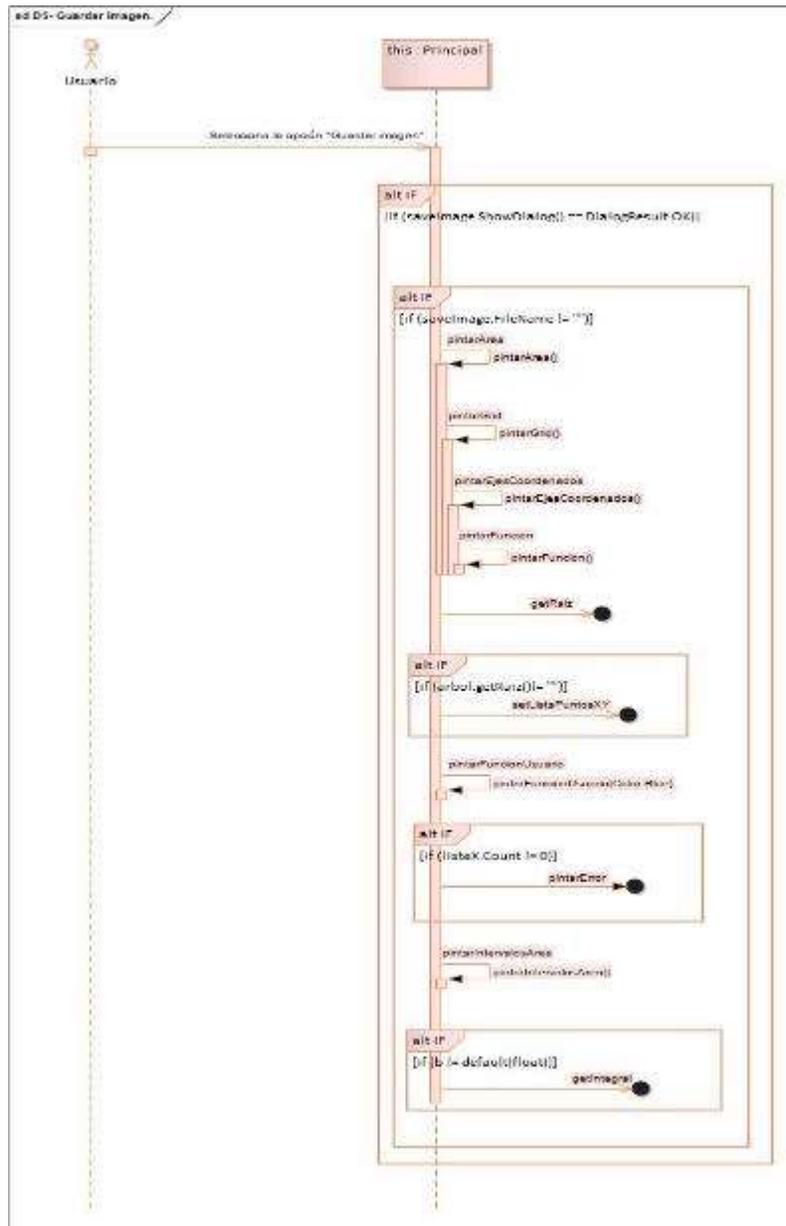


Diagrama de secuencia del CU Guardar.



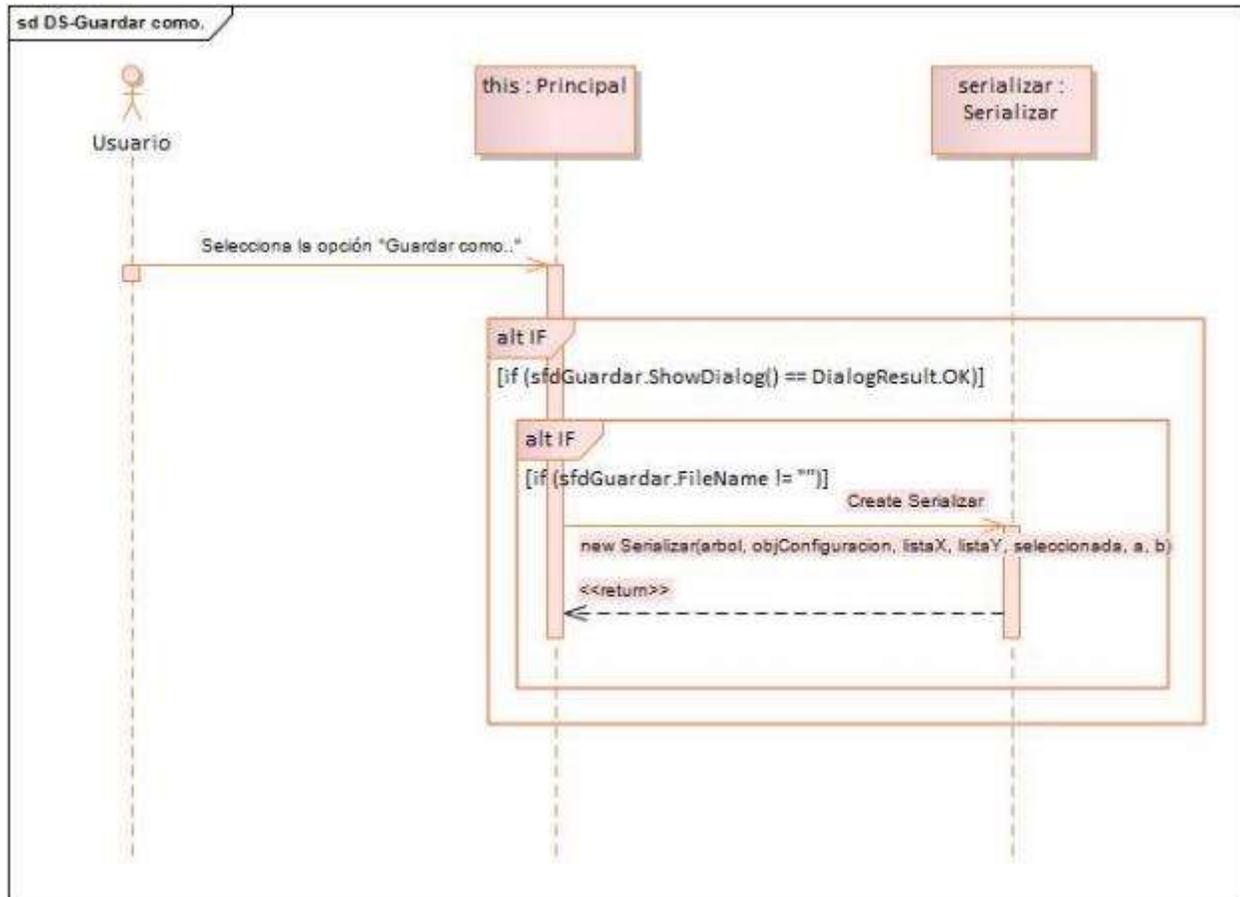
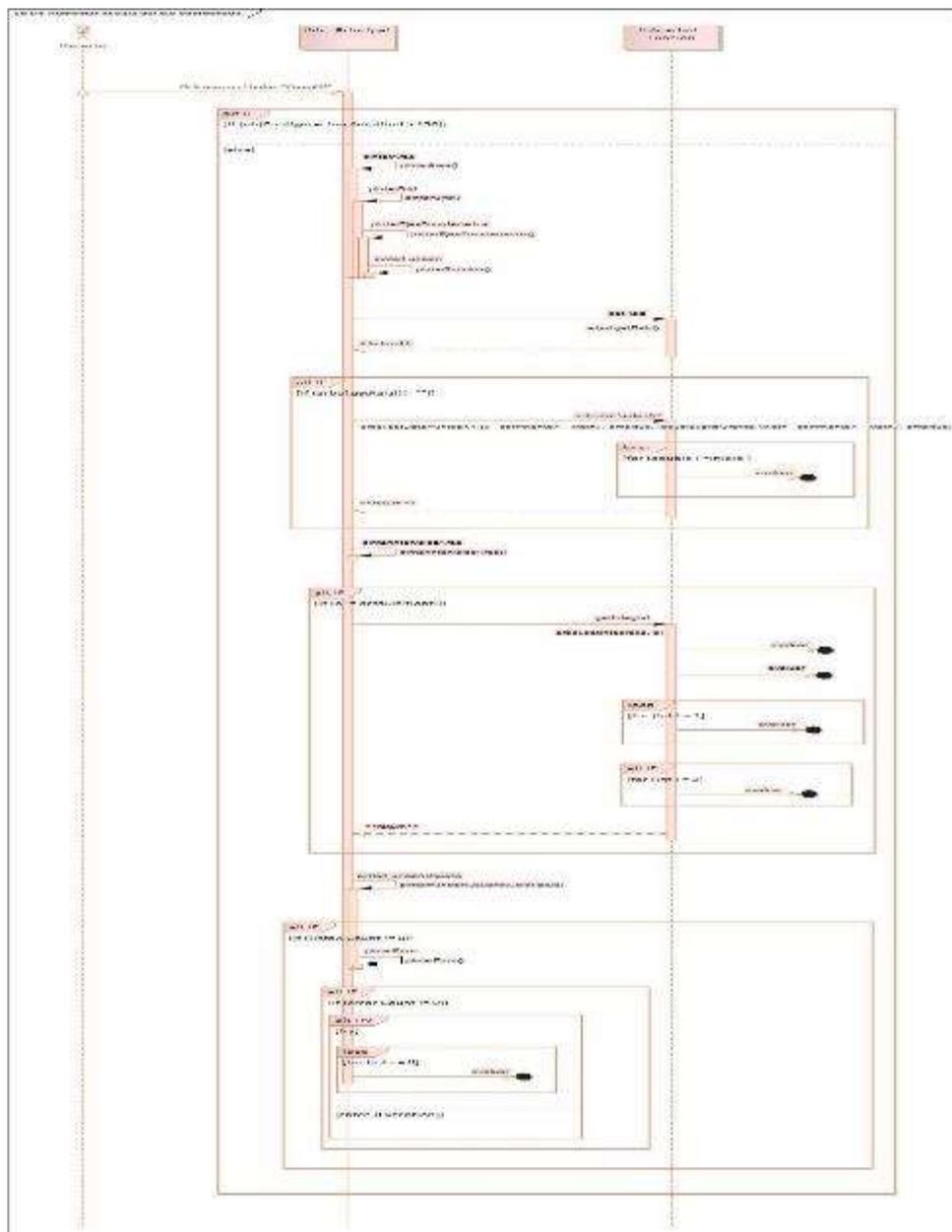


Diagrama de secuencia del CU Modificar escala del eje coordenado.



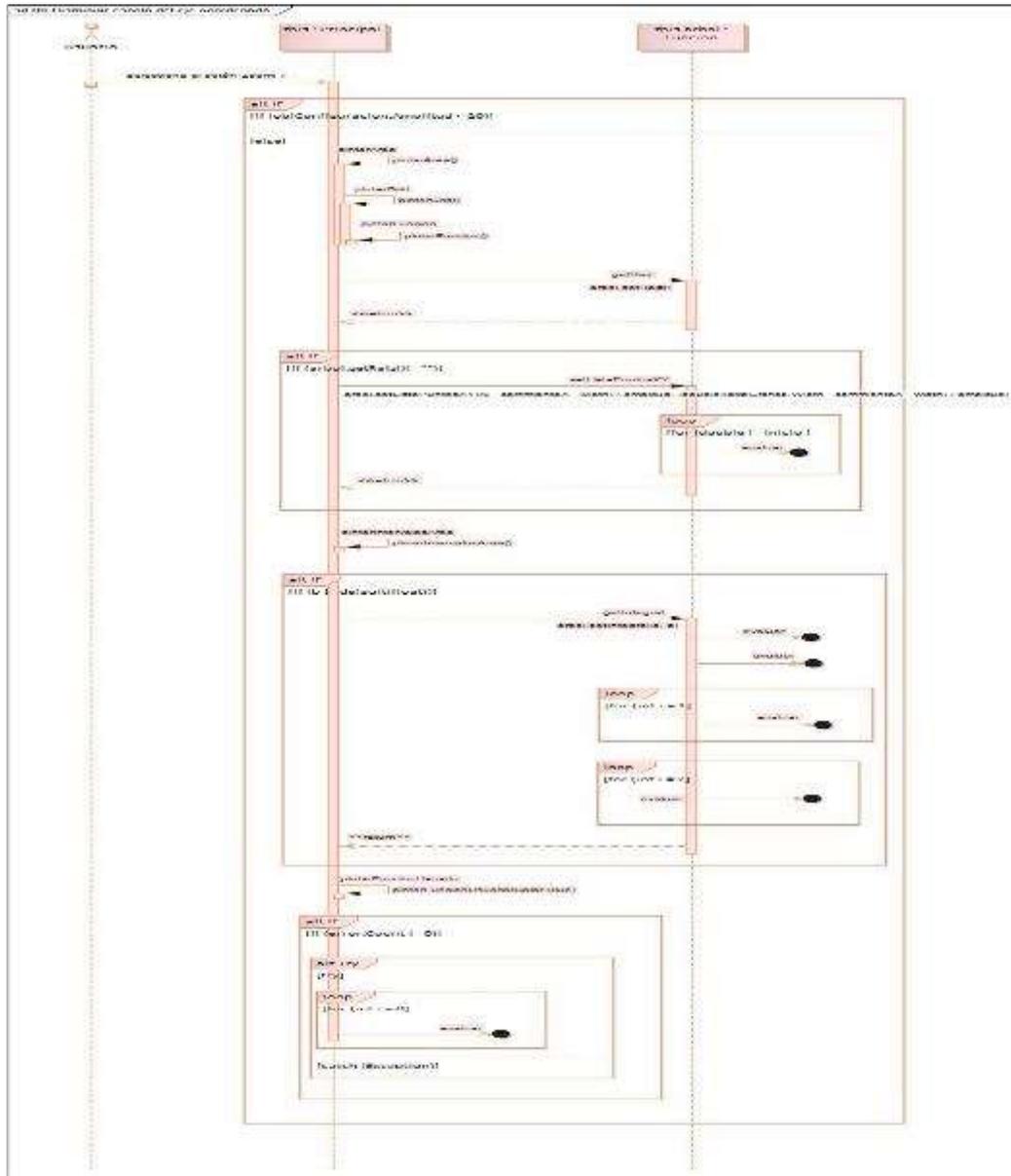


Diagrama de secuencia del CU Mostrar propiedades de la ecuación.

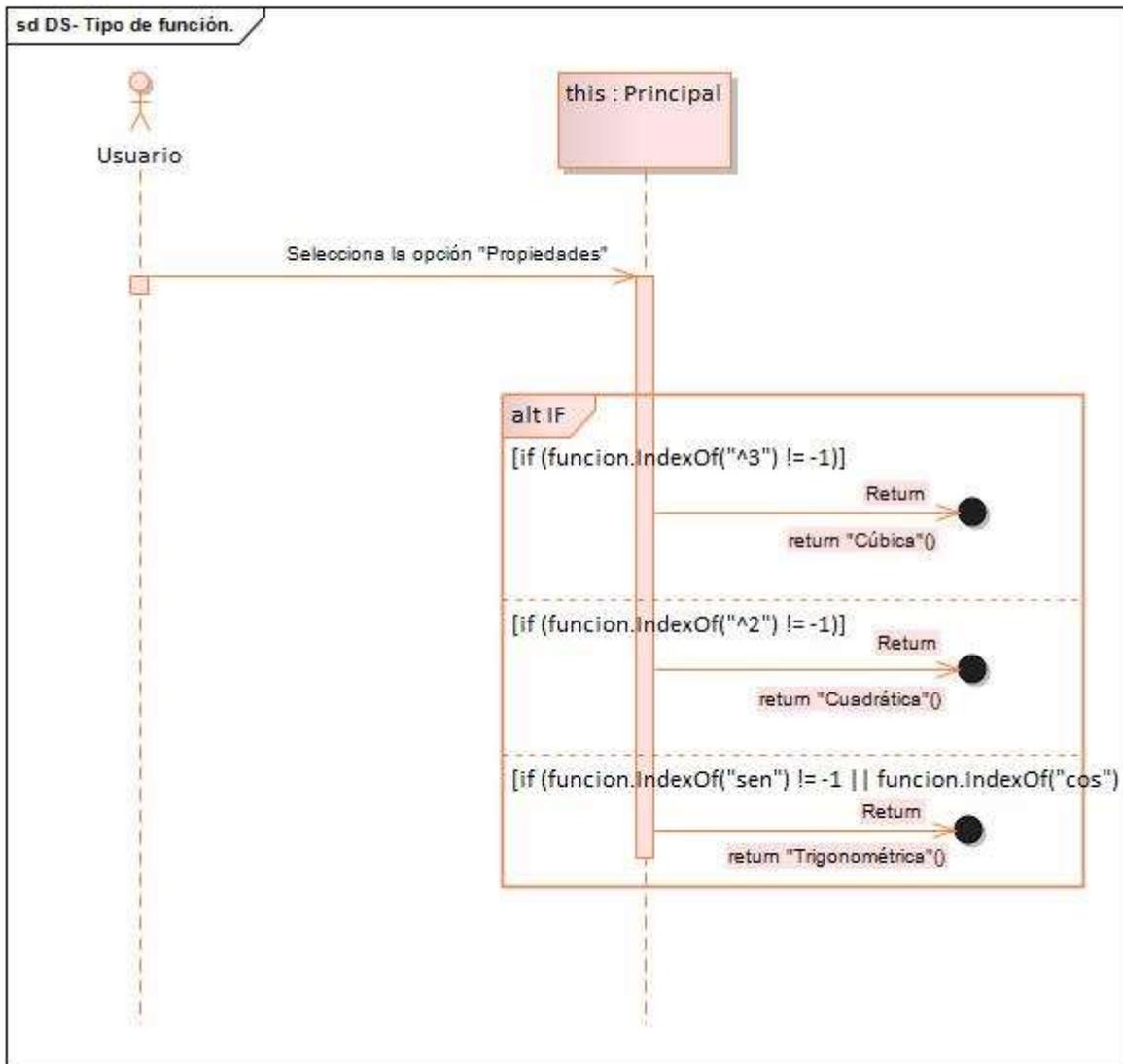


Diagrama de secuencia del CU Mover sistema coordenado.

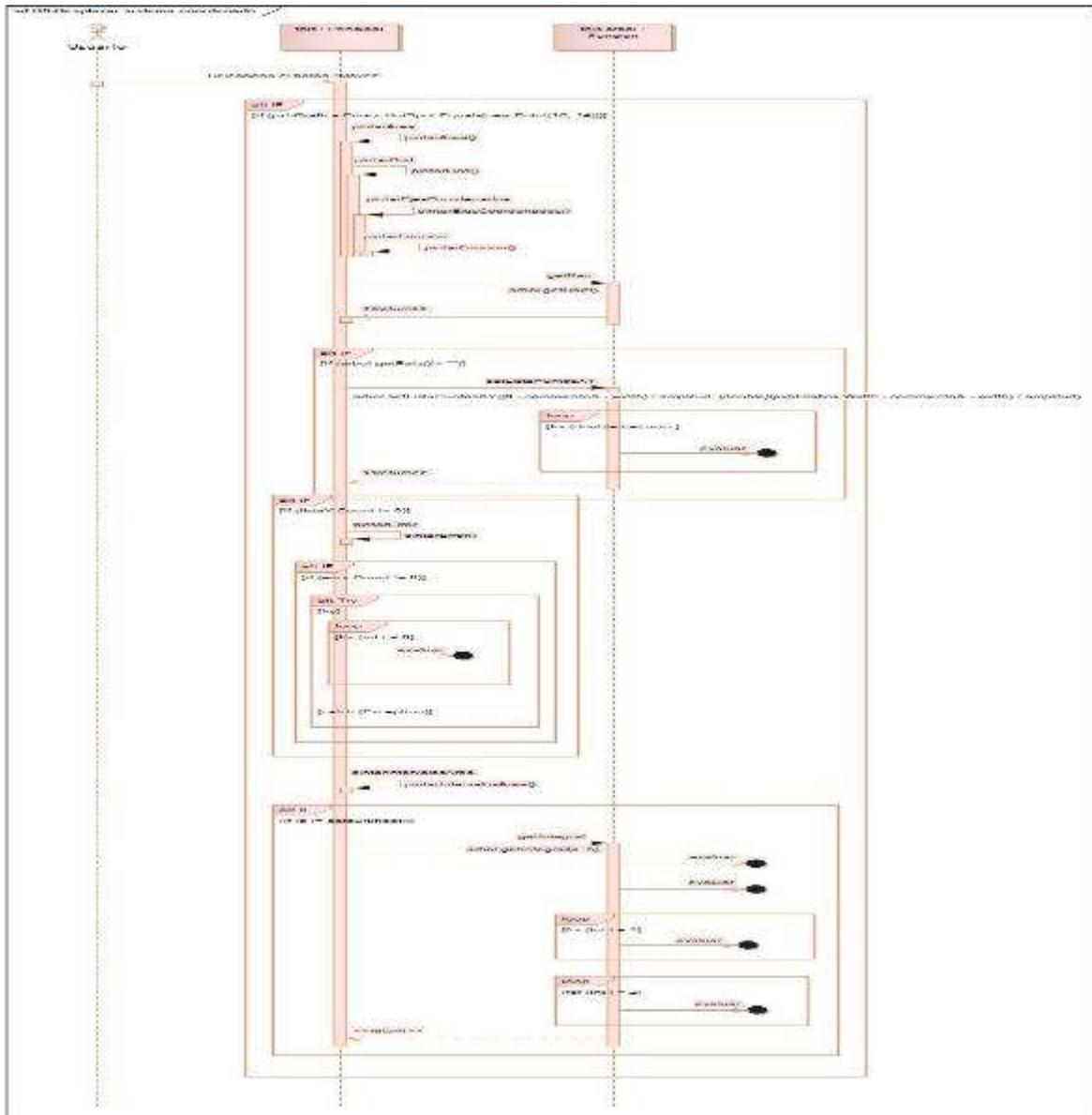
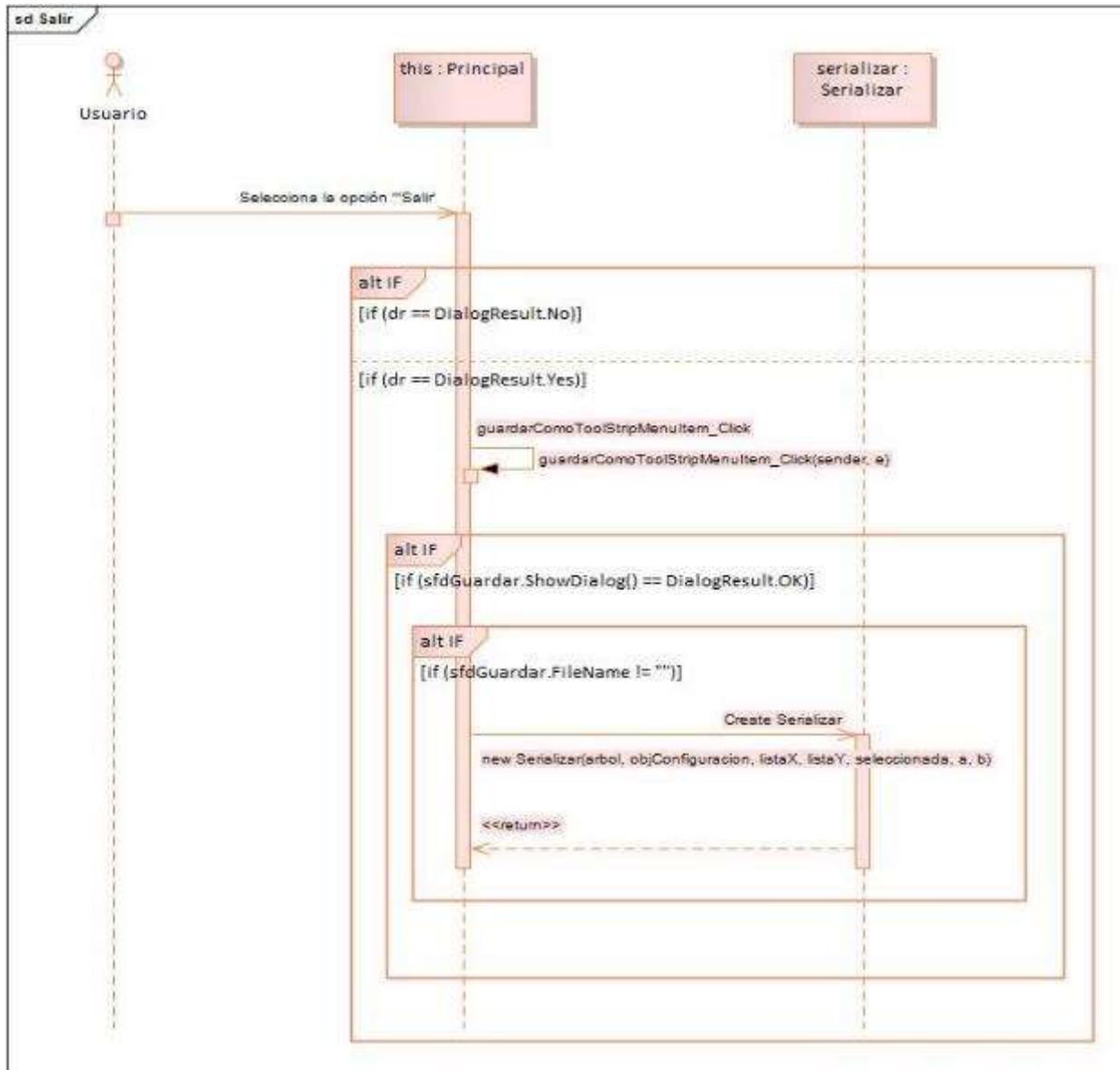


Diagrama de secuencia del CU Salir



Anexo 5 Descripción de las clases del sistema.

- **Clase:** Controler

Propósito: Es la clase encargada de la generación del árbol de expresiones a partir de una función.

Descripción:

Nombre: Controler	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	getArbol(string expresión)
Descripción:	Obtiene el árbol de expresiones.
Nombre:	getArbolCompuesto(string expresión)
Descripción:	Obtiene el árbol de expresiones compuestas.
Nombre:	filtro(string expresión)
Descripción:	Realiza el filtro a una expresión matemática.

- **Clase:** IntervaloInicio

Propósito: Se encarga de definir el inicio del intervalo para la representación gráfica.

Descripción

Nombre: IntervaloInicio	
Tipo de clase: entidad	
Atributo	Tipo
_valor	double
_incluido	bool
Para cada responsabilidad:	
Nombre:	IntervaloInicio()
Descripción:	Constructor de la clase.
Nombre:	Valor
Descripción:	Obtiene el valor para el inicio del intervalo.
Nombre:	Incluido

Descripción:	Dice verdadero o falso a la inclusión.
--------------	--

- **Clase:** IntervaloFin

Propósito: Se encarga de definir el fin del intervalo para la representación gráfica.

Descripción

Nombre: IntervaloFin	
Tipo de clase: entidad	
Atributo	Tipo
_valor	double
_incluido	bool
Para cada responsabilidad:	
Nombre:	IntervaloFin ()
Descripción:	Constructor de la clase.
Nombre:	Valor
Descripción:	Obtiene el valor para el fin del intervalo.
Nombre:	Incluido
Descripción:	Dice verdadero o falso a la inclusión.

- **Clase:** Signos

Propósito: Se encarga de analizar los elementos de la función y si son signos le otorga un orden de prioridad.

Descripción

Nombre: Signos	
Tipo de clase: Entidad	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	esSigno(string signo)
Descripción:	Dado un valor devuelve si es o no un signo.
Nombre:	esNumero(string número)
Descripción:	Dado un valor devuelve si es o no un número.

Nombre:	esDecimal(char valor)
Descripción:	Dado un valor devuelve si es o no decimal.
Nombre:	getPrioridad(string signo)
Descripción:	Según el signo dice la prioridad que tiene.
Nombre:	calcular(double val1, double val2, string signo)
Descripción:	Realiza el cálculo según el signo en cuestión.
Nombre:	calcularCompuestos(string comp)
Descripción:	Realiza cálculos entre valores decimales.

- **Clase:** Funcion

Propósito: Se encarga de realizar el análisis de la función.

Descripción

Nombre: Funcion.	
Tipo de clase: entidad	
Atributo	Tipo
_c	Controler
_derecho	Funcion
_finIntervalo	IntervaloFin
_inicioIntervalo	IntervaloInicio
_izquierdo	Funcion
_listaPuntosX	List<double>
_listaPuntosY	List<double>
_raíz	string
_maxmin	string
_a	int
_b	int
Para cada responsabilidad:	
Nombre: Funcion()	

Descripción: Constructor de la clase.	
Nombre:	getRaiz()
Descripción:	Obtiene la raíz del árbol.
Nombre:	construir()
Descripción:	Construye el árbol de expresiones.
Nombre:	evaluar(double x)
Descripción:	Dado un valor el determina si es la raíz o es un signo para proceder a efectuar las operaciones con la función.
Nombre:	esHoja()
Descripción:	Dice si es Hoja.
Nombre:	FinIntervalo
Descripción:	Devuelve el fin del intervalo.
Nombre:	InicioIntervalo
Descripción:	Devuelve el inicio del intervalo.
Nombre:	getMaxMin()
Descripción:	Obtiene los valores máximos y mínimos de la función.
Nombre:	setIntervalos(string intervalos)
Descripción:	Modifica el intervalo.
Nombre:	setListaPuntosXY(double inicio, double fin)
Descripción:	Modifica la lista de puntos de los ejes.
Nombre:	xyClear()
Descripción:	Limpia los ejes de coordenadas.
Nombre:	ListaPuntosY
Descripción:	Contiene la lista de puntos del eje y.
Nombre:	ListaPuntosX
Descripción:	Contiene la lista de puntos del eje x.
Nombre:	getResultadoSistemaLineal(double[,] matriz)
Descripción:	Obtiene el resultado de la ecuación lineal.
Nombre:	errorGrafico(Funcion, List<double> listaX, List<double> listaY, List<double> error)

Descripción:	Devuelve el error de la gráfica.
--------------	----------------------------------

- **Clase:** Calculos

Propósito: Realiza cálculos entre polinomios.

Descripción

Nombre: Calculos	
Tipo de clase: Entidad	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	multiplicaEscalarPolinomio (polinomio: List<double>, escalar: double)
Descripción:	Multiplica un valor escalar por un polinomio.
Nombre:	multiplicaPolinomio (polinomio1: List<double>, polinomio2: List<double>)
Descripción:	Multiplica dos polinomios.
Nombre:	sumaPolinomios (polinomio1: List<double>, polinomio2: List<double>)
Descripción:	Suma polinomios.

- **Clase:** Configuracion

Propósito: Editar los principales campos visuales de la aplicación, díganse color y estilo de la grilla, entre otras.

Descripción

Nombre: Configuracion	
Tipo de clase: entidad	
Atributo	Tipo
_amplitud	int
_numeros	bool
_corrimientoX	float
_corrimientoY	float
_newCorrX	float

_newCorrY	float
_cuadrícula	Cuadrícula
_lenguaje	Lenguaje
Para cada responsabilidad:	
Nombre:	Configuracion()
Descripción:	Constructor de la clase.
Nombre:	Amplitud
Descripción:	Obtiene amplitud.
Nombre:	Numeros
Descripción:	Obtiene números
Nombre:	CorrimientoX
Descripción:	Obtiene corrimiento de X.
Nombre:	CorrimientoY
Descripción:	Obtiene corrimiento de Y.
Nombre:	NewCorrX
Descripción:	Obtiene nuevo corrimiento de X.
Nombre:	NewCorrY
Descripción:	Obtiene nuevo corrimiento de Y.
Nombre:	Cuadrícula
Descripción:	Obtiene cuadrículas.
Nombre:	Lenguaje
Descripción:	Obtiene lenguajes.

- **Clase:** Lenguaje

Propósito: Permite modificar el idioma dentro de la aplicación.

Descripción

Nombre: Lenguaje	
Tipo de clase: entidad	
Atributo	Tipo

_imagen	Bitmap
Para cada responsabilidad:	
Nombre:	Lenguaje()
Descripción:	Constructor de la clase.
Nombre:	getLenguajeSeleccionado
Descripción:	Obtiene el lenguaje seleccionado.
Nombre:	Imagen
Descripción:	Obtiene la imagen.

- **Clase:** Ingles

Propósito: Muestra los textos dentro de la aplicación en Inglés.

Descripción

Nombre: Ingles	
Tipo de clase: entidad	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Ingles
Descripción:	Constructor de la clase.
Nombre:	getLenguajeSeleccionado
Descripción:	Obtiene el lenguaje seleccionado.
Nombre:	Imagen
Descripción:	Obtiene la imagen.

- **Clase:** Español

Propósito: Muestra los textos dentro de la aplicación en Español.

Descripción

Nombre: Español	
Tipo de clase: entidad	
Atributo	Tipo

Para cada responsabilidad:	
Nombre:	Español()
Descripción:	Constructor de la clase.
Nombre:	getLenguajeSeleccionado
Descripción:	Obtiene el lenguaje seleccionado.
Nombre:	Imagen
Descripción:	Obtiene la imagen.

- **Clase:** Cuadrícula

Propósito: Modificar el tamaño y el color de las cuadrículas dentro del eje de coordenada.

Descripción

Nombre: Cuadrícula	
Tipo de clase: entidad	
Atributo	Tipo
color	Color
tipo	Tipo
grid	Boolean
Para cada responsabilidad:	
Nombre:	Cuadrícula()
Descripción:	Constructor de la clase.
Nombre:	Color
Descripción:	Obtiene el color de las cuadrículas.
Nombre:	Tipo
Descripción:	Obtiene el tipo de las cuadrículas.
Nombre:	GetGrid
Descripción:	Obtiene las grillas.

Anexo 6 Aval de aprobación del cliente.

