Universidad de las Ciencias Informáticas

Facultad 6

Título: Software alasBioSyS v1.2: Módulo de Editor de Ecuaciones

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Ariel Pérez Rueda

Tutor: Ing. Carlos Gonzales Iglesias
Co-tutor: Ing. Edel Moreno Lemus

La Habana, junio de 2012.
"Año 54 de la Revolución"

La posibilidad de realizar un sueño es lo que hace que la vida sea interesante.
Pablo Coelho

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ariel Pérez Rueda

_____

Firma del Autor

Ing. Carlos González Iglesias

_____

Firma del Tutor

Ing. Edel Moreno Lemus

_____

Firma del Co-tutor

DATOS DE CONTACTO

Tutor:

Ing. Carlos González Iglesias
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: cgonzalez@uci.cu

Co-tutor:

Ing. Edel Moreno Lemus
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: emoreno@uci.cu

AGRADECIMIENTOS

A mi familia por haberme apoyado en toda mi carrera, en especial a mi mama.

A mi tutor Carlos por su apoyo e interés porque todo saliera bien.

A mis compañeros de aula que han sido muchos pero que su amistad incondicional siempre será valorada.

A mis compañeros de cuarto que han convivido conmigo por mucho tiempo.

A todas mis amistades que me apoyaron en todo momento en especial a aquellos que estuvieron tanto en los buenos como en los malos momentos.

En general a todas las personas que han apoyado de una forma u otra a mi formación como persona.

DEDICATORIA

A mis padres: María del Carmen y Eulises Fuentes, en especial a mi madre por ser la persona que más quiero, además por su confianza, dedicación y esfuerzo.
A toda mi familia que tanto quiero que de una forma u otra siempre tendré presente.
A todos mis amigos de la Universidad: Que me han apoyado, comprendido y aceptado durante estos 5 años de la carrera.
A todos mis amigos de la Isla de la Juventud José L, Lázaro, Yasmani.

RESUMEN

Con el gran auge de la bioinformática y el incremento del estudio de los sistemas biológicos en el mundo, han estado desarrollándose técnicas para comprender el comportamiento de los organismos dado un problema científico puntual. Esto ha traído consigo que varias ciencias estén implicadas en el estudio de estos sistemas. Una de estas ciencias es la matemática, la cual describe los sistemas biológicos mediante ecuaciones diferenciales ordinarias. Estos sistemas biológicos descritos por ecuaciones diferenciales ordinarias en muchos casos presentan una alta complejidad, por lo que se ha estado buscando una forma más fácil de entenderlos y comprobar su veracidad.

El análisis dimensional es una condición aplicable a estos sistemas descritos por ecuaciones diferenciales ordinarias, el cual utiliza un principio conocido como análisis de homogeneidad dimensional. Es por ello que esté presente trabajo tiene como objetivo llevar a todos los investigadores que utilicen el software alasBioSyS una herramienta que facilitará el trabajo y análisis de los sistemas biológicos descritos por ecuaciones diferenciales ordinarias.

PALABRAS CLAVES: Sistemas Biológicos, Análisis Dimensional, Ecuaciones Diferenciales Ordinarias, Homogeneidad Dimensional.

TABLA DE CONTENIDOS

AGRADECIMIENTOS IV

DEDICATORIA V

RESUMEN VI

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 4

1.1: Análisis dimensional. 4
1.2: Métodos de resolución de la homogeneidad dimensional. 5
1.3 Interfaz visual para el análisis dimensional. 7
1.3.1 Componentes de la interfaz visual para la homogeneidad dimensional en Java. 7
1.4 Herramientas y metodologías utilizadas. 8
1.4.1 Metodología: OpenUP 8
1.4.2 Lenguaje de modelado: UML 8
1.4.3 Herramientas Case: Visual Paradigm 8
1.4.4 Lenguaje de programación: Java 9
1.4.5 IDE de desarrollo: NetBeans 6.9 9
1.4.6 Asistentes Matemáticos 10
1.5 Conclusiones del capitulo 11

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA 12

2.1. Introducción 12
2.2. Breve descripción del sistema 12
2.3. Modelo de Dominio 12
2.3.1. Representación del Modelo de Dominio 12
Figura 1.1: Modelo de Dominio del análisis dimensional. 13
2.4. Especificación de los Requisitos del Sistema 14
2.4.1. Requisitos Funcionales 14
2.4.2. Requisitos No Funcionales 14
2.5. Definición de los Casos de Uso del Sistema 16
2.5.1. Actores del Sistema 16
2.5.2. Diagrama de Casos de Uso del Sistema 16
2.5.3. Descripción de los Casos de Uso del Sistema 17
2.6. Conclusiones 20

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA 21

3.1. Introducción 21
3.2. Patrón Arquitectónico Utilizado 21
3.3 Patrones de Diseño Utilizados 23
3.4 Diagramas de Clases del Diseño 26
3.5 Diagramas de Secuencia. 28
3.6. Modelo de Despliegue 30
3.7. Conclusiones 30

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA 31

4.1. Introducción 31
4.2. Implementación 31

4.3. Pruebas del Sistema 36
4.4. Conclusiones 44
CONCLUSIONES 45
RECOMENDACIONES 46
BIBLIOGRAFÍA 47
ANEXOS 50

ÍNDICE DE FIGURAS

Figura 1.1: Modelo de Dominio del análisis dimensional. 13
Figura 2.1: Diagrama de Casos de Uso del Sistema del editor de ecuaciones del software alasBioSyS v1.2..... 17
Figura 3.1: Esquema del patrón de Modelo-Vista-Controlador del editor de ecuaciones del software alasBioSyS v1.2 21
Figura 4.1: Patrón Modelo-Vista-Controlador para el editor de ecuaciones del software alasBioSyS v1.2..... 23
Figura 5.1: Ejemplo del Patrón Creador aplicado en el editor de ecuaciones del software alasBioSyS v1.2..... 24
Figura 6.1: Ejemplo del Patrón Experto aplicado en el editor de ecuaciones del software alasBioSyS v1.2. 24
Figura 7.1: Ejemplo del Patrón Controlador aplicado en el editor de ecuaciones del software alasBioSyS v1.2. 25
Figura 8.1: Diagrama de Clases del Caso de Uso Asignar Dimensión del editor de ecuaciones del software alasBioSyS v1.2..... 26
Figura 9.1: Diagrama de Clases del Caso de Uso Verificar Homogeneidad Dimensional del editor de ecuaciones del software alasBioSyS v1.2..... 27
Figura 10.1: Diagrama de Clases del Caso de Uso Modificar Fuente del editor de ecuaciones del software alasBioSyS v1.2..... 27
Figura 11.1: Diagrama de Secuencia del Caso de Uso Asignar Dimensión del editor de ecuaciones del software alasBioSyS v1.2. 28
Figura 12.1: Diagrama de Secuencia del Caso de Uso Verificar Homogeneidad Dimensional del editor de ecuaciones del software alasBioSyS v1.2. 29
Figura 13.1: Diagrama de Secuencia del Caso de Uso Modificar Fuente del editor de ecuaciones del software alasBioSyS v1.2. 29
Figura 14.1: Diagrama de despliegue del software alasBioSyS..... 30
Figura 15.1: Diagrama de componentes del editor de ecuaciones del software alasBioSyS v1.2 32
Figura 16.1: Editor de Ecuaciones 34

ÍNDICE DE FIGURAS

Figura 17.1:gg Visual para asignar las dimensiones..... 34
Figura 18.1: Visual para seleccionar las dimensiones que posterior mente se asignaran. 35
Figura 19.1: Visual mostrando el proceso del análisis de homogeneidad dimensional. 35
Figura 20.1: Reporte para visualizar donde existen errores en el sistema. 36

INTRODUCCIÓN

La biología de sistemas es un término muy utilizado en la actualidad por muchos científicos ya que utiliza como unidad fundamental un modelo computacional que es creado a raíz de componentes celulares y sus mecanismos de comunicación, brindando la oportunidad de combinar moléculas y sus mensajes mediante estos aparatos computacionales, probándolos con numerosas condiciones y combinaciones. El modelo de un sistema biológico puede ser representado por un sistema de ecuaciones diferenciales ordinarias el cual explica mediante simulaciones cómo se comporta este sistema en diferentes ámbitos. Esta ayuda de la matemática a la biología influye mucho en el proceso de modelación de estos sistemas biológicos debido al artificio de técnicas y herramientas matemáticas. (1)

Hoy en la actualidad estos modelos han crecido considerablemente tanto en tamaño como en complejidad lo cual ha traído consigo un mejor desarrollo de las herramientas y técnicas para la creación de estos modelos mediantes ecuaciones matemáticas y con ello un mayor esfuerzo para la resolución de estos modelos debido a la complejidad y crecimiento de la información generada por dichos sistemas. Por esto se hace necesario el uso de sistemas computacionales que ayuden con la simulación, análisis y modelación de los sistemas biológicos mediante sistemas de ecuaciones diferenciales ordinarias.

Estas herramientas o software creados para facilitar el trabajo de científicos en muchos casos no son gratuitos o asequibles para todas las personas de la comunidad científica, además de que algunos de estos programas o herramientas solo se crean para resolver algún problema en específico.

El análisis dimensional es uno de los problemas que pudiera presentarse en el análisis de los sistemas biológicos pero para resolver este problema hasta el momento según bibliografía consultada no ha sido resuelta por ningún software. Este análisis presenta una potente ayuda que permite simplificar el estudio de cualquier fenómeno en el que estén involucradas muchas magnitudes físicas en forma de variables independientes. En general, cualquier ecuación física ha de relacionar términos, en relaciones de igualdad o de suma, que pertenezcan a la misma magnitud o que tengan las mismas dimensiones, esta propiedad recibe el nombre de condición de homogeneidad dimensional (2), (3). En este proceso se pueden admitir términos o componentes de un sistema biológico que sean adimensionales los cuales van a ser necesarios para estudiar cada sistema. (4)

Estos términos adimensionales de cada sistema nos van a permitir:

- Analizar con mayor facilidad el sistema que estamos estudiando.
- Reducir considerablemente el número de veces que se debe analizar cualquier sistema para averiguar el comportamiento o respuesta de este.

El análisis dimensional tiene muchas aplicaciones las cuales ayudan con el entendimiento de un sistema, algunas de estas aplicaciones son:

- Detección de errores de cálculo.
- Resolución de problemas cuya solución directa conlleva dificultades matemáticas insalvables.
- Creación y estudio de modelos reducidos.
- Consideraciones sobre la influencia de posibles cambios en los modelos, etc.

Además de esto el análisis dimensional sirve para comprobar la veracidad o falsedad de las fórmulas físicas o ecuaciones matemáticas, haciendo uso del principio de homogeneidad dimensional. La homogeneidad dimensional garantiza o verifica que las dimensiones del término de la izquierda deben ser iguales a las de la derecha y lo mismo debe ocurrir cuando hay más de un miembro a la derecha o a la izquierda. En cuanto a los sistemas biológicos el análisis de la homogeneidad dimensional representa una parte fundamental e importante ya que ayuda en el análisis del sistema en estudio atendiendo a parámetros o propiedades que se quieran analizar en el sistema como por ejemplo el crecimiento de una población de *n* células en el tiempo. (4)

La Universidad de Ciencias Informáticas (UCI) en conjunto con el Centro de Inmunología Molecular (CIM) se dieron la tarea de crear un software el cual ayuda con la simulación, modelación y análisis de los sistemas biológicos, el cual presenta una interfaz muy amigable para todo aquel que lo utilice y en específico para aquellos científicos o investigadores que necesiten analizar sistemas biológicos.

De esta forma surge alasBioSyS el cual representa una potente herramienta para el trabajo con los sistemas biológicos, ya que permite simular, analizar y modelar estos sistemas. En cuanto al análisis y simulación de los sistemas biológicos son condiciones que este software realiza mediante un editor de ecuaciones en el cual se describen los sistemas biológicos mediante ecuaciones diferenciales ordinarias. Este editor en su versión actual 1.1 no cuenta con varias funcionalidades y una de ellas es el análisis dimensional.

Por lo que esta necesidad debido a su importancia en el análisis y simulación de los sistemas biológicos descritos por ecuaciones diferenciales ordinarias se convierte en el **problema a resolver** en este trabajo: *¿Cómo contribuir con el trabajo del investigador para realizar el análisis dimensional de los sistemas biológicos descritos por ecuaciones diferenciales ordinarias en el editor de ecuaciones del software alasBioSyS.?*

Se plantea como objeto de estudio: El análisis dimensional.
Se plantea como campo de acción: El análisis dimensional en los sistemas biológicos descritos por ecuaciones diferenciales ordinarias.

Para resolver el problema anterior se plantea como objetivos:

Objetivo general:

Desarrollar la versión 1.2 del editor de ecuaciones del software alasBioSyS que permita el análisis dimensional de los sistemas biológicos descritos por ecuaciones diferenciales ordinarias.

Objetivos específicos:

- Definir el procedimiento para realizar el análisis dimensional.
- Diseñar las funcionalidades para la versión 1.2 del software alasBioSyS.
- Implementar funcionalidades para la versión 1.2 del software alasBioSyS.
- Validar las funcionalidades implementadas en la versión 1.2 del software alasBioSyS.

Tareas de la investigación:

- Estudio del estado del arte de los software que implementan análisis dimensional.
- Definición de un procedimiento para realizar el análisis dimensional.
- Definición de las funcionalidades a agregar para el análisis dimensional en el Editor de Ecuaciones.
- Realización del diseño del análisis dimensional.
- Implementación del análisis dimensional.
- Implementación del cambio de fuente del editor.
- Realización de pruebas de caja negra para validar el correcto funcionamiento del análisis dimensional.

Estructuración del contenido con una breve explicación de sus partes:

Capítulo 1: Fundamentación teórica. En este capítulo se presenta el marco teórico estudiado y seleccionado para desarrollar nuestro objeto de estudio, orientado al análisis de homogeneidad dimensional de los sistemas biológicos descritos por sistemas de ecuaciones diferenciales ordinarias.

Capítulo 2: Características del sistema. En este capítulo se describen los casos de uso encontrados. Se hará una descripción del sistema a automatizar y se mostrará un diagrama de casos de uso del sistema para un mejor entendimiento del mismo. Se definirán los actores del sistema y los requerimientos funcionales del sistema. Se expondrán los requerimientos mínimos que debe tener el sistema de cómputo donde se vaya a instalar la aplicación.

Capítulo 3: Análisis y diseño del sistema. En este capítulo se realiza el diagrama de clases del diseño dando respuesta a la solución que se propone. Se describen los principios de diseño que se tuvieron en cuenta durante el desarrollo de la aplicación.

Capítulo 4: Implementación y Prueba. En este capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes, para esto se muestra el diagrama de componentes. Además se muestran los casos de prueba de caja negra para cada caso de uso con el objetivo de validar el correcto funcionamiento de la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se describe el estado del arte de la investigación desarrollada, además de dar una panorámica de los principales conceptos utilizados en la investigación. Además se explican las herramientas y tecnologías utilizadas para dar respuesta al problema planteado.

1.1: Análisis dimensional.

El análisis dimensional aplicado a los sistemas biológicos descritos por ecuaciones diferenciales ordinarias representa una potente herramienta para el entendimiento y comprensión de propiedades y comportamientos, estas ecuaciones pueden ser modeladas, simuladas y analizadas computacionalmente reduciendo el número y el tiempo de las pruebas aplicadas a estos sistemas. Su resultado fundamental, el teorema de Vaschy-Buckingham (más conocido por teorema Π) permite cambiar el conjunto original de parámetros de entrada dimensionales de un problema físico por otro conjunto de parámetros de entrada adimensionales más reducido. Así, para este tipo de cálculos, se utilizan ecuaciones dimensionales, que son expresiones algebraicas que tienen como variables a las unidades físicas fundamentales y derivadas, las cuales se usan para demostrar fórmulas, equivalencias o para dar unidades a una respuesta. En el mundo existen programas computacionales que pueden ser utilizados como complemento para realizar este análisis, como el Máxima, Mathematic y Maple los cuales son asistentes matemáticos muy potentes que presentan funcionalidades que apoyan al cálculo matemático. Según el estudio realizado y la bibliografía consultada, este análisis aplicado a la biología de sistemas no se encuentra desarrollado por ningún programa de cómputo.

1.1.1 Magnitudes adimensionales.

En física, química, ingeniería y otras ciencias aplicadas se denomina magnitud adimensional a toda aquella magnitud que carece de una magnitud física asociada. Así, serían magnitudes adimensionales todas aquellas que no tienen unidades, o cuyas unidades pueden expresarse como relaciones matemáticas puras. Algunos ejemplos de magnitudes adimensionales son:

- La cantidad de objetos de un conjunto.
- Las razones de proporcionalidad.

- Los ángulos: a pesar que pueden expresarse en grados, radianes, etc., sus unidades se pueden definir de forma puramente matemática, sin necesidad de definir una unidad física, simplemente expresándolos como fracción de una circunferencia.
- Algunos números usados en ingeniería como el número de Mach, el número de Reynolds, etc.
- Un punto en el plano o el espacio.

1.2: Métodos de resolución de la homogeneidad dimensional.

La resolución de la homogeneidad dimensional se basa en relación de suma y multiplicación de las dimensiones de los términos de una ecuación. Estas dimensiones son descritas mediante la física y se clasifican en 7 magnitudes básicas, estas magnitudes son Longitud (L), Tiempo (T), Masa (M), Temperatura (Θ), Intensidad de corriente eléctrica (I), Cantidad de sustancia (N) e Intensidad luminosa (J), pudiendo analizar la homogeneidad dimensional de cualquier ecuación que exprese magnitudes físicas como:

$$e = s + vt + at^2$$

Donde e va a representar una longitud L, s va a representar una distancia L, vt va a representar otra longitud expresada en $(m/s)*s$, convirtiéndolo quedaría $L*T^{-1}*T$ y at^2 va representar otra longitud $(m/s^2)*s^2$, sustituyendo y resolviendo la ecuación para comprobar si son homogéneos dimensionalmente quedaría así:

$$L = L + \left(L*T^{-1}*T\right) + L*T^{-2}*T^2$$

$$L = L = L = L$$

Aplicando esto a un sistema biológico conocido como Presa-Depredador o Lotka quedaría de la siguiente forma:

$$\frac{\partial x}{\partial t} = x * (\alpha - \beta * y)$$
$$\frac{\partial y}{\partial t} = -y * (\gamma - \delta * x)$$

Donde x y y representan cantidades de unidades representadas por N , t representa el tiempo T , y $\alpha, \beta, \gamma, \delta$ son términos adimensionales que se pueden representar por U . Sustituyendo estas magnitudes la ecuación quedaría:

$$\frac{\partial N}{\partial T} = N * (U - U * N)$$
$$\frac{\partial N}{\partial T} = -N * (U - U * N)$$

Simplificando la ecuación llegaríamos al siguiente resultado:

$$N = N * U - N^2 * U$$
$$N = -N * U - N^2 * U$$

Eliminando los términos adimensionales de la ecuación ya que representan términos constante o numéricos el resultado quedaría:

$$N = N = N$$
$$N = N = N$$

Debido a que la cantidad de unidades representan un número.

1.3 Interfaz visual para el análisis dimensional.

La interfaz visual es una potente herramienta para el entendimiento y manejo del análisis de homogeneidad dimensional ya que una interfaz gráfica está compuesta por principios básicos para tratar de ayudar a los usuarios y en estos casos a los investigadores y científicos que utilicen el software, algunos de estos principios son:

- **Autonomía:** Se debe dar un ambiente flexible para que se pueda aprender rápidamente a usar la aplicación.
- **Eficiencia del Usuario:** Se debe priorizar la productividad del usuario antes que la productividad de la máquina.
- **Interfaces Explorables:** Siempre que sea posible se debe permitir que el usuario pueda salir ágilmente, dejando una marca que indique el estado de avance de su trabajo, con vistas a continuarlo en otro momento.
- **Curva de Aprendizaje:** El aprendizaje de un producto y su usabilidad no son mutuamente excluyentes. Lo ideal es que la curva de aprendizaje sea nula y que el usuario principiante pueda alcanzar el dominio total de la aplicación sin esfuerzo.
- **Protección del Trabajo:** Se debe asegurar que el usuario nunca pierda su trabajo, ya sea por error de su parte, problemas de transmisión de datos, de energía o alguna otra razón inevitable.

1.3.1 Componentes de la interfaz visual para la homogeneidad dimensional en Java.

La plataforma Java está creada para ser independiente del sistema operativo que se utilice, por lo que las aplicaciones creadas en ella no deben ser dependientes de algún sistema operativo. Esta plataforma brinda herramientas y componentes para la generación de software basándose en la utilización de librerías que integran todas estas herramientas y componentes. Estas librerías estándares ayudan al programador a desarrollar software, algunas de ellas son:

- **(Abstract Window Toolkit):** la cual permite al programador crear interfaces gráficas en las que se pueden manejar imágenes, botones, menús y otros elementos básicos para interactuar con el usuario.
- **Librería gráfica SWING:** El paquete Swing es parte de la JFC (Java Foundation Classes) en la plataforma Java. La JFC provee facilidades para ayudar a construir GUIs y abarca componentes tales como: botones, tablas o marcos. (5)

1.4 Herramientas y metodologías utilizadas.

1.4.1 Metodología: OpenUP

OpenUP es un proceso unificado que aplica acercamientos iterativos e incrementales dentro de un ciclo de vida estructurado. Además abraza una filosofía pragmática, ágil y enfocada en la naturaleza de colaboración del desarrollo del software. Es una herramienta que se puede ampliar a una gran variedad de tipos del proyecto.

El esfuerzo personal en un proyecto de OpenUP se organiza en micro-incrementos, los cuales representan unidades cortas de trabajo que producen un paso constante y mensurable en el progreso del proyecto. El ciclo de vida de un proyecto en OpenUP está estructurado en cuatro fases: inicio, elaboración, construcción y transición. (6)

1.4.2 Lenguaje de modelado: UML

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Desde su creación se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. (7)

1.4.3 Herramientas Case: Visual Paradigm

Como herramienta CASE se empleó Visual Paradigm para el trabajo con UML. La herramienta está diseñada para una amplia gama de usuarios que incluye a: ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, interesados en la creación de Grandes Sistemas de Software de manera confiable, a través del paradigma Orientado a Objetos. Esta herramienta soporta los últimos estándares de anotaciones de JAVA y UML y provee soporte para la generación de código y la ingeniería inversa para Java. Las transiciones del análisis al diseño y de éste a la implementación, están adecuadamente integradas dentro de la herramienta CASE, de manera que reduce significativamente los esfuerzos de todas las etapas del ciclo de desarrollo de software. (8)

1.4.4 Lenguaje de programación: Java

El lenguaje de programación Java es un lenguaje de propósito general, concurrente, basado en clases y orientado a objetos. Su diseño fue concebido para que los programadores puedan lograr fluidez con el lenguaje. Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello, Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje, además, el mismo elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica o recolector de basura). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread (hilo) de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de esta. Una de las características más importante de Java es que posee una arquitectura neutral, es decir el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. (9), (10)

1.4.5 IDE de desarrollo: NetBeans 6.9

Netbeans se refiere a una plataforma para el desarrollo de aplicaciones usando Java. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden extenderse agregándole nuevos módulos. Debido a que los módulos pueden desarrollarse independientemente, las aplicaciones basadas en la plataforma NetBeans pueden extenderse por otros desarrolladores de software. (11) Algunas de sus características fundamentales son:

- Es un proyecto de código abierto.
- El IDE es desarrollado para distintas plataformas como Linux, MacOS X, Solaris y también Windows.
- Trae incorporados todos los plug-ins que se necesitan para trabajar, además, es configurable en su instalación, o sea, el usuario escoge los plug-ins con los que quiere instalar el IDE.

1.4.6 Asistentes Matemáticos

Los asistentes matemáticos ofrecen a sus usuarios una herramienta interactiva de cálculo y un versátil lenguaje de programación para una rápida y precisa solución a problemas técnicos. Representan programas que ayudan a realizar cálculos numéricos, analizando y visualizando los datos para resolver problemas matemáticos, físicos, etc.

Se describen como sistemas de gestión de datos y análisis estadísticos en entorno gráficos los cuales reciben datos desde cualquier fichero y generan informes, tablas, gráficos de distribución, moda estadísticas descriptivas y análisis estadísticos complejos pudiendo mejorar los resultados obtenidos con la metodología tradicional y además ser utilizados en la enseñanza del Algebra Lineal y en el Cálculo diferencial e integral.

Estos asistentes permiten un ambiente para resoluciones de problemas matemáticos complejos que involucren expresiones algebraicas, simbólicas, cálculos numéricos de alta precisión, etc. Entre los asistentes matemáticos más utilizados se encuentran los siguientes:

- Matlab
- Derive
- Maple
- Mathematica
- Octave
- Maxima
- Y otros que se especializan en visualizar graficas dando respuesta a las ecuaciones que en ellos se resuelven.

Estos asistentes matemáticos antes mencionados se pueden clasificar en tres grupos según la salida de los datos. Estos grupos son: simbólicos-numéricos, numéricos y gráficos.

Para la realización del análisis dimensional utilizaremos asistentes que su salida se exprese en un resultado simbólico-numérico ya que facilitaría el trabajo con las variables de una ecuación sin importar el resultado número. Entre estos asistentes se encuentran el Maple, Mathematica y Maxima, sin embargo Maxima es bastante ligero en comparación con los dos antes mencionados, además de presentar las funcionalidades suficientes y necesarias para realizar los análisis que se necesitan. Después de analizar estos asistentes se decidió utilizar el Maxima (15) para la realización del análisis dimensional.

1.5 Conclusiones del capitulo

En este capítulo se definieron conceptos importantes para la comprensión de la investigación, se profundizó en el análisis dimensional y se explicaron la metodología y las herramientas utilizadas para llegar a la solución propuesta.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En el presente capítulo se brinda una breve descripción de la solución propuesta, sus requisitos funcionales y no funcionales, así como los actores que intervienen en ella. Se presenta el diagrama de casos de uso del sistema, así como la descripción de los mismos.

2.2. Breve descripción del sistema

La versión 1.2 del editor de ecuaciones del software alasBioSyS constara con funcionalidades las cuales ayudaran con la realización del análisis dimensional. Estas funcionalidades estarán enmarcadas en asignarles dimensiones a los términos de una ecuación y verificar su homogeneidad dimensional.

2.3. Modelo de Dominio

Para poder entender el contexto en que se enmarca el sistema se describe el funcionamiento de la aplicación mediante una serie de conceptos, entidades y sus relaciones, agrupándolos en un modelo de dominio.

El modelo de dominio es una representación de los conceptos u objetos del mundo real, significativos para un problema. Tiene como objetivo fundamental la descripción de las clases más importantes en el sistema y representa conceptos del mundo real, no de los componentes de software. (12)

2.3.1. Representación del Modelo de Dominio

El diagrama de clases del modelo de dominio del presente trabajo se muestra en la figura 1.1, donde se aprecia que el especialista o investigador inserta los sistemas biológicos, los cuales están descritos por uno o varios modelos matemáticos, a los cuales se le asignan dimensiones a sus variables, esto generaría un modelo dimensional en el cual se verificaría la homogeneidad dimensional.

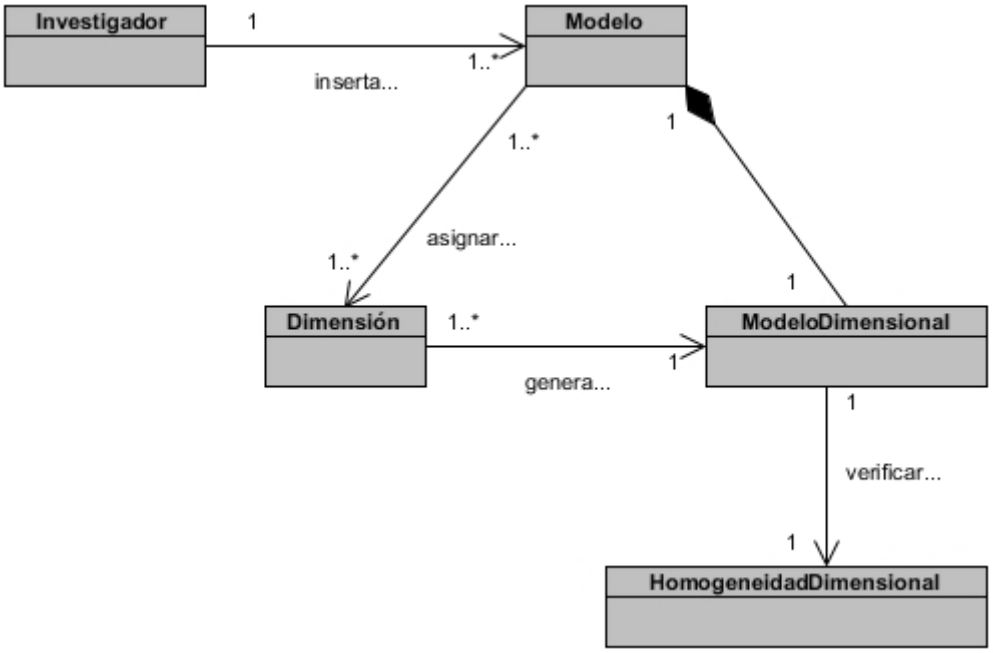

Figura 1.1: Modelo de Dominio del análisis dimensional.

A continuación se describen cada una de las clases mostradas en la Figura 1.1.

Investigador: Persona capacitada para insertar el modelo matemático de un sistema biológico en el editor de ecuaciones del software alasBioSyS.

Modelo: Conjunto de ecuaciones diferenciales que describen a un Sistema Biológico.

Dimensión: Clase que permitirá asignarle dimensiones a las variables del modelo.

Modelo Dimensional: Conjunto de ecuaciones generada a partir de la asignación de las dimensiones a las variables del modelo.

Homogeneidad Dimensional: Técnica que permitirá verificar si el modelo dimensional está bien generado.

2.4. Especificación de los Requisitos del Sistema

Los requisitos son condiciones o capacidades que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Esta definición se extiende y se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes, para satisfacer un contrato, una norma o una especificación. (13)

2.4.1. Requisitos Funcionales

Los requisitos funcionales de un sistema describen su funcionalidad, los servicios que de él se esperan, o los que proveerá, entre ellos: sus entradas, salidas y excepciones.

- RF1 – Asignar dimensiones al modelo descrito.
    - RF1.1 – Asignar dimensión a los términos del miembro derecho de las ecuaciones del modelo.
    - RF1.2 – Asignar dimensión a los términos del miembro izquierdo de las ecuaciones del modelo.
- RF2 – Verificar la homogeneidad dimensional del sistema descrito.
    - RF2.1 – Crear el modelo dimensional a partir del modelo antes descrito.
    - RF2.2 – Simplificar el modelo dimensional.
    - RF2.3 – Comprobar homogeneidad dimensional.
- RF3 – Modificar Fuente en el editor de ecuaciones.

2.4.2. Requisitos No Funcionales

Los requisitos no funcionales son propiedades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado. Normalmente están

vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. (14)

2.4.2.1. Apariencia o interfaz externa

- La aplicación deberá tener una interfaz externa amigable, que sea sencilla y fácil de entender por el usuario para facilitar el trabajo del usuario en la aplicación.

2.4.2.2. Software

- En cuanto a restricciones de sistemas operativos la aplicación actualmente funciona en Linux y Windows.
- La máquina virtual de Java 1.6.
- Maxima como asistente matemático. (15)

2.4.2.3. Hardware

- 512 MB de memoria RAM como mínimo, aunque lo ideal sería 1 GB.
- Procesadores Pentium IV o superiores.

2.4.2.4. Requerimientos de implementación

- Lenguaje de programación Java.

2.4.2.5. Usabilidad

- El sistema podrá ser usado por aquellos usuarios que posean conocimientos básicos en el campo de la modelación de sistemas biológicos.
- El sistema le ofrecerá al investigador la posibilidad de realizar un análisis más profundo de los sistemas biológicos.

2.4.2.6. Requisitos Legales y Derecho de Autor

- Los derechos de autor serán registrados por la UCI.

2.5. Definición de los Casos de Uso del Sistema

Los casos de uso son descripciones de las funcionalidades del sistema y se utilizan para obtener información de cómo debe trabajar este. Independientemente de la implementación, describen bajo la forma de acciones y reacciones, el comportamiento de un sistema desde el punto de vista del usuario. (16)

2.5.1. Actores del Sistema

Se define Actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos, así como a entidades abstractas como el tiempo. (17)

Actor	Descripción
Investigador	Representa al usuario que va a hacer uso del sistema y quien tiene la posibilidad de interactuar con todas las funcionalidades de este.

2.5.2. Diagrama de Casos de Uso del Sistema

El diagrama de casos de uso del sistema sirve para especificar la comunicación y el comportamiento de un sistema, mediante su interacción con los usuarios y/u otros sistemas (18). A continuación se muestra el diagrama de casos de uso del presente trabajo.

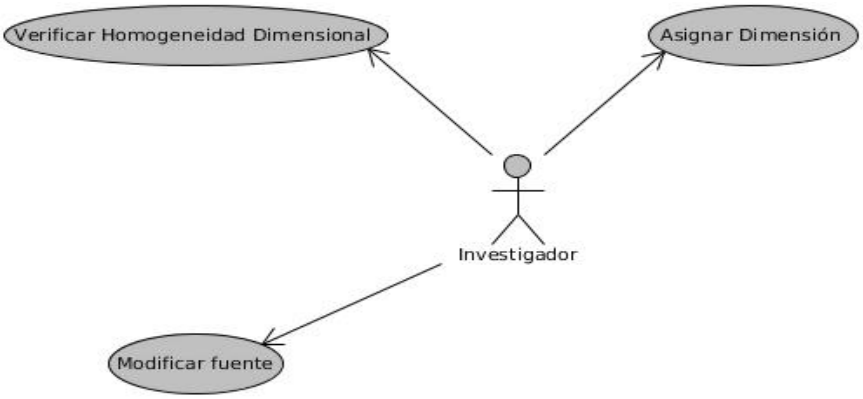

Figura 2.1: Diagrama de Casos de Uso del Sistema del editor de ecuaciones del software alasBioSyS v1.2

2.5.3. Descripción de los Casos de Uso del Sistema

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario, o con otro sistema, para conseguir un objetivo específico. A continuación se muestra la descripción de los casos de uso del sistema del presente trabajo.

2.5.3.1. Caso de Uso Asignar Dimensión

Caso de Uso	**Asignar Dimensión**
Actores	Investigador
Propósito	Asignarle dimensiones a los términos del modelo matemático descrito.
Resumen	El caso de uso se inicia cuando el investigador selecciona la opción asignar dimensión.
Referencia	RF1
Precondiciones	Debe existir algún modelo matemático en el editor de ecuaciones.

Prioridad	Crítico
Flujo Normal de Eventos	
Acción de Actor	**Respuesta del Sistema**
1. - El investigador selecciona la opción asignar dimensión.	1.1 – El sistema muestra una ventana de interfaz que le permitirá al investigador asignarle las dimensiones a los términos del modelo matemático.
2. - El investigador selecciona el término y la dimensión a asignar y presiona el botón Add.	2.1 – El sistema asigna la dimensión seleccionada al término correspondiente.
3. - El investigador al asignarle dimensiones a todos los términos presiona el botón Aceptar.	3.1 – El sistema finaliza la asignación de dimensión.
Flujos Alternos	
Acción de Actor	**Respuesta del Sistema**
1. - El investigador selecciona la opción asignar dimensión.	1.2 – El sistema le muestra una ventana de error porque debe existir ecuaciones diferenciales en el editor de ecuaciones que describan un sistema biológico.

2.5.3.2. Caso de Uso Verificar Homogeneidad Dimensional

Caso de Uso	**Verificar Homogeneidad Dimensional**
Actores	Investigador
Propósito	Verificar la homogeneidad dimensional en las ecuaciones descritas por el sistema biológico.
Resumen	El caso de uso se inicia cuando el investigador

	selecciona la opción verificar dimensión.
Referencia	RF2
Precondiciones	Debe existir algún modelo matemático en el editor de ecuaciones. Debe habérsele asignado dimensiones al modelo matemático en el editor de ecuaciones.
Prioridad	Crítico
Flujo Normal de Eventos	
Acción de Actor	**Respuesta del Sistema**
1. - El investigador selecciona la opción verificar homogeneidad dimensional.	1.1 - El sistema identifica en que sistema operativo está trabajando. 1.2 – El sistema busca en el sistema operativo el asistente matemático Maxima . 1.3 – El sistema realiza el análisis dimensional y muestra una ventana de diálogo con un mensaje de que el modelo no tiene errores dimensionalmente.
Flujos Alternos	
Acción de Actor	**Respuesta del Sistema**
1. - El investigador selecciona la opción verificar homogeneidad dimensional.	1.2 – El sistema muestra un mensaje de error debido a que el máxima no está instalado. 1.3 – El sistema le muestra una ventana de error mostrando en que ecuación es donde existe el error.

2.5.3.2. Caso de Uso Modificar Fuente

Caso de Uso	**Modificar Fuente**
Actores	Investigador
Propósito	Modificar la fuente del editor de ecuaciones brindándole una vista al usuario más amigable.
Resumen	El caso de uso se inicia cuando el investigador selecciona la opción change font.
Referencia	RF3
Precondiciones	Ninguna
Prioridad	Normal
Flujo Normal de Eventos	
Acción de Actor	**Respuesta del Sistema**
1. - El investigador selecciona la opción change font.	1.1 – El sistema muestra una ventana donde le permite al investigador cambiar la fuente del editor.
2. - El investigador modifica la fuente a su criterio y presiona el botón “Aceptar”.	2.1 - El sistema cambia la fuente del editor.

2.6. Conclusiones

- Se brinda una clara definición de los requisitos que debe cumplir el sistema, seleccionando los actores, así como los casos de uso del sistema.

- Se ganó en claridad en cuánto al sistema a construir. También se sentaron las bases para las restantes fases del proceso de diseño e implementación, a través de la descripción de los casos de uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En el presente capítulo se dará una descripción del estilo arquitectónico utilizado para el desarrollo de la herramienta, se describirán los patrones de diseño empleados así como los diagramas de clases del diseño.

3.2. Patrón Arquitectónico Utilizado

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones, para organizar los distintos componentes.

Para el desarrollo de la herramienta se utilizó el patrón Modelo Vista Controlador (Model-View-Controller-MVC), un patrón que muestra la relación entre el modelo, la vista y el controlador.

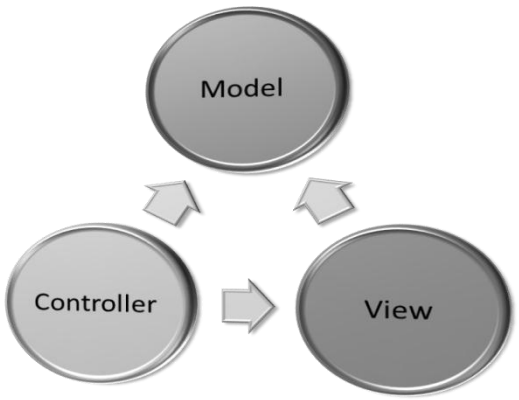

Figura 3.1: Esquema del patrón de Modelo-Vista-Controlador del editor de ecuaciones del software alasBioSyS v1.2

El patrón Modelo – Vista – Controlador mostrado en la figura 3.1, está catalogado como un patrón de arquitectura de software donde:

Modelo: El modelo incluye los datos de estado para cada componente. Define las reglas de negocio (la funcionalidad del sistema). Un modelo está destinado a servir como una aproximación computacional o abstracción de algún proceso o sistema de mundo real.

Vista: La vista se refiere a cómo se ve el componente o aplicación en la pantalla. Es una forma de visualizar el estado del modelo. Recibe datos del modelo y los muestra al usuario.

Controlador: El controlador es la porción de la interfaz de usuario que dicta cómo la aplicación interactúa con los acontecimientos. Ofrece facilidades para cambiar el estado del modelo. Es el medio por el cual el usuario interactúa con la aplicación aceptando la entrada del usuario y ordenando el modelo y la vista para realizar acciones sobre la base de esa entrada y es responsable de asignar al usuario final la acción de respuesta de la aplicación. (19)

Entre las principales ventajas que presenta este patrón arquitectónico se encuentran:

1. Es posible tener diferentes vistas para un mismo modelo (ej. representación de un conjunto de datos como una tabla o como un diagrama de barras).
2. Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.
3. Proporciona un mecanismo de configuración a componentes complejos muchos más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción).

Aplicando este patrón arquitectónico a la aplicación, el diseño quedó de la siguiente manera:

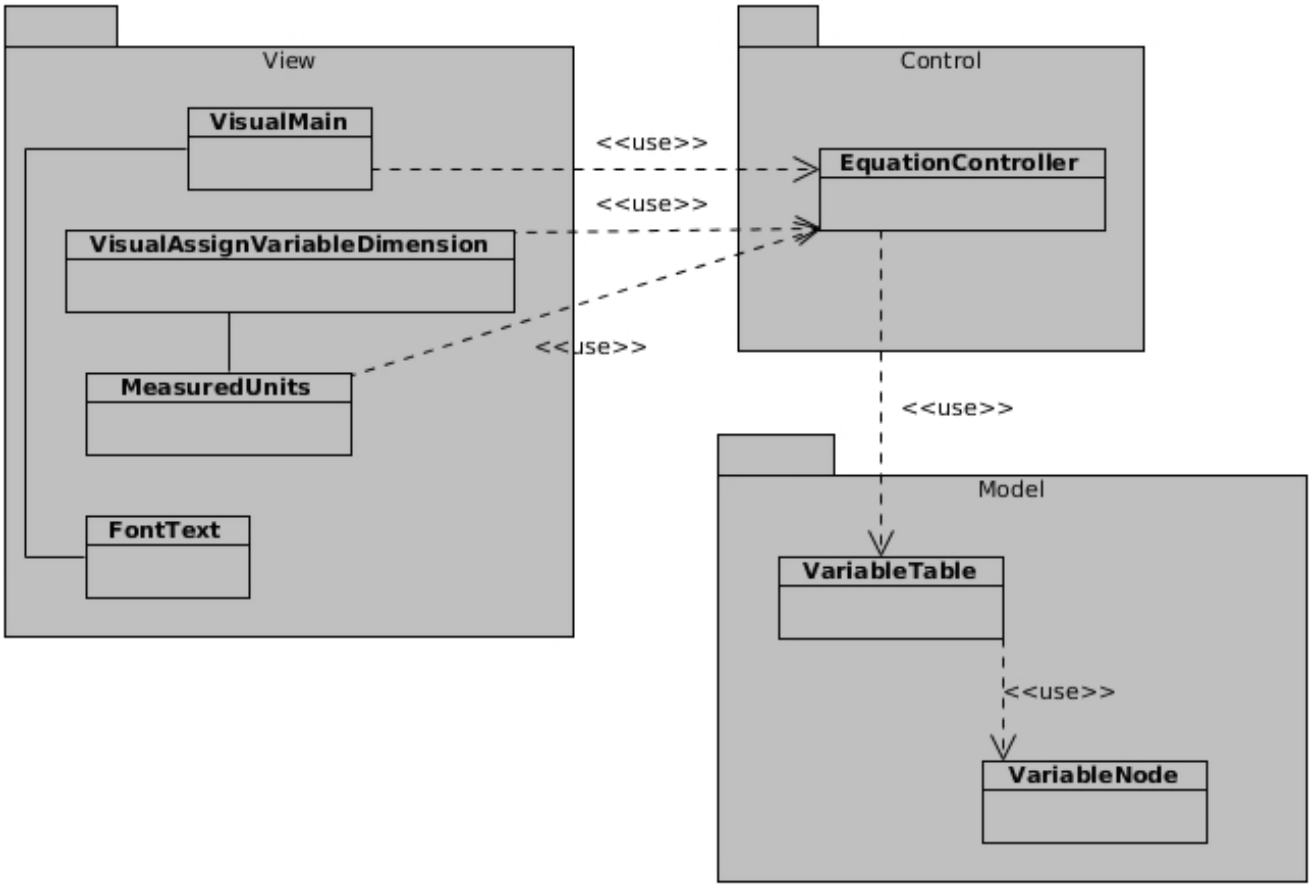

Figura 4.1: Patrón Modelo-Vista-Controlador para el editor de ecuaciones del software alasBioSyS v1.2

3.3 Patrones de Diseño Utilizados

3.3.1 Patrón Creador

Los patrones de creación muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades. La valía de los patrones de creación nos dice como estructurar y encapsular estas decisiones. (20)

- La clase B contenga a la clase A
- B sea una agregación (o composición) de A
- B almacene a A
- B tenga los datos de inicialización de A(datos que requiere su constructor)
- B use a A

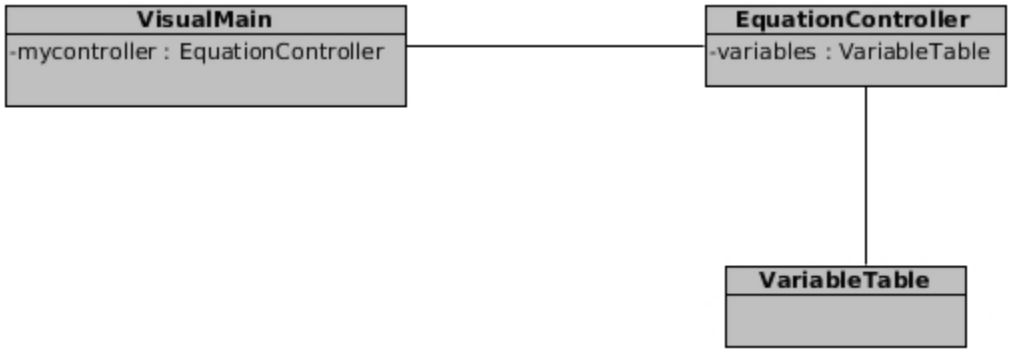

Figura 5.1: Ejemplo del Patrón Creador aplicado en el editor de ecuaciones del software alasBioSyS v1.2

3.3.2 Patrón Experto

Nos indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo y así tendremos una mayor cohesión y la información se mantendrá encapsulada atendiendo al bajo acoplamiento. (21)

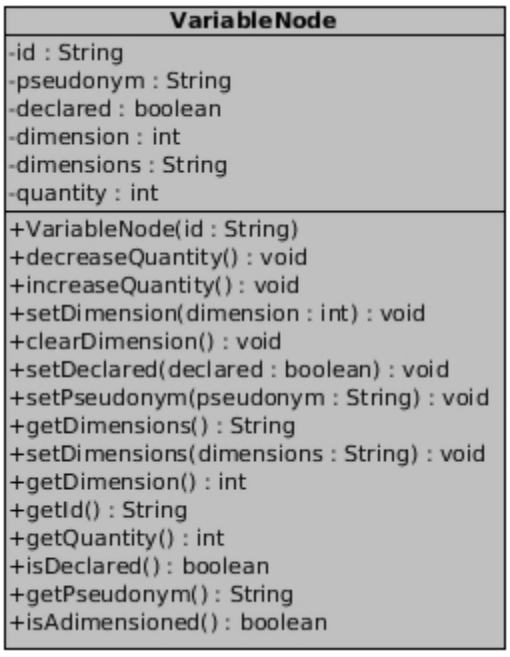

Figura 6.1: Ejemplo del Patrón Experto aplicado en el editor de ecuaciones del software alasBioSyS v1.2.

3.3.3 Patrón Alta Cohesión

El patrón Alta Cohesión nos dice que la información que almacena una clase debe de ser coherente y debe estar relacionada con la clase tanto como sea posible.

3.3.4 Patrón Bajo Acoplamiento.

El patrón Bajo Acoplamiento es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases.

3.3.5 Patrón Controlador

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. (20)

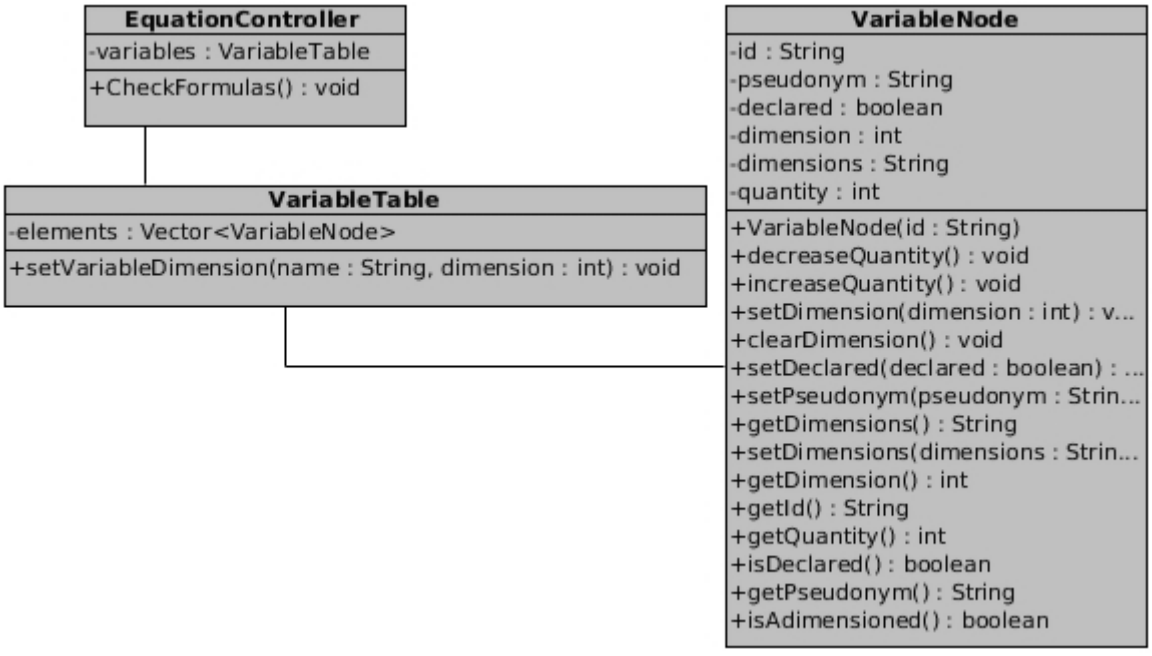

Figura 7.1: Ejemplo del Patrón Controlador aplicado en el editor de ecuaciones del software alasBioSyS v1.2.

3.4 Diagramas de Clases del Diseño

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. Nos sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento. A continuación se muestran los diagramas de clase:

3.4.1 Caso de Uso Asignar Dimensión

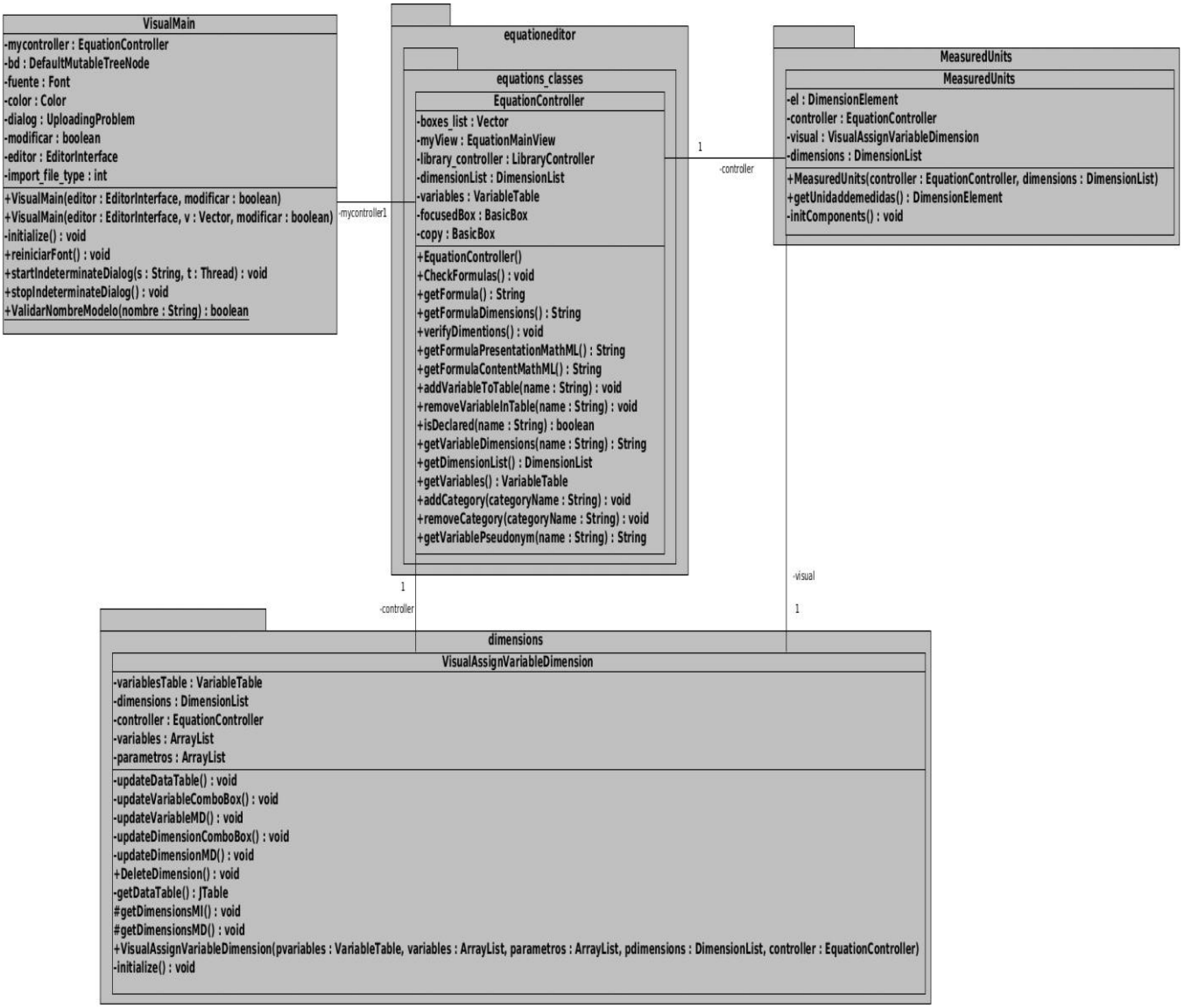

Figura 8.1: Diagrama de Clases del Caso de Uso Asignar Dimensión del editor de ecuaciones del software alasBioSyS v1.2

3.4.2 Caso de Uso Verificar Homogeneidad Dimensional

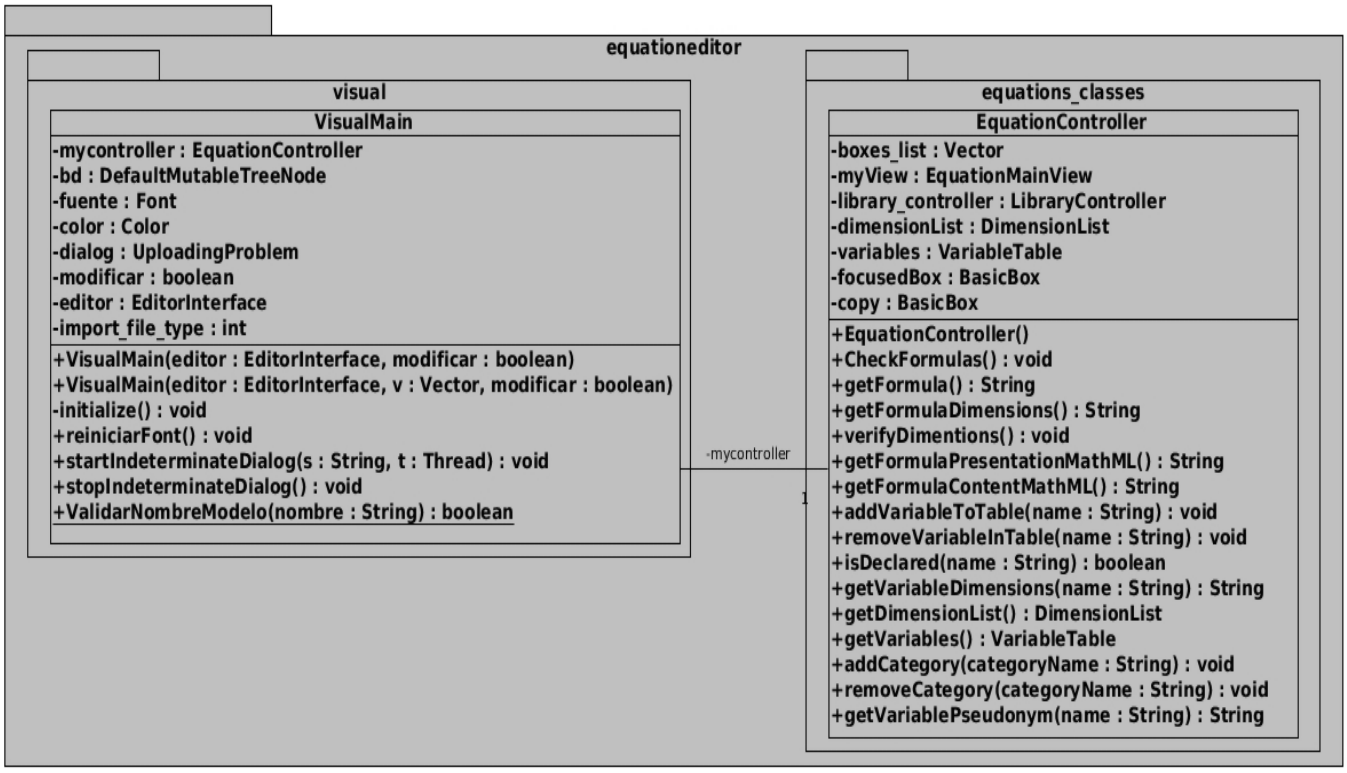

Figura 9.1: Diagrama de Clases del Caso de Uso Verificar Homogeneidad Dimensional del editor de ecuaciones del software alasBioSyS v1.2

3.4.3 Caso de Uso Modificar Fuente

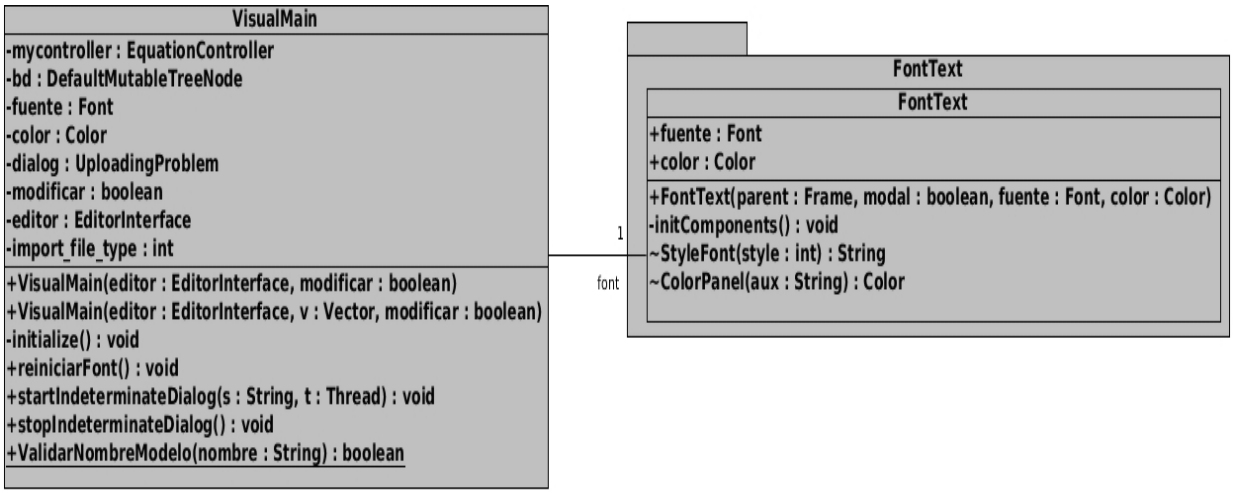

Figura 10.1: Diagrama de Clases del Caso de Uso Modificar Fuente del editor de ecuaciones del software alasBioSyS v1.2.

3.5 Diagramas de Secuencia.

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema.

3.5.1 Diagrama de Secuencia del Caso de uso Asignar Dimensión

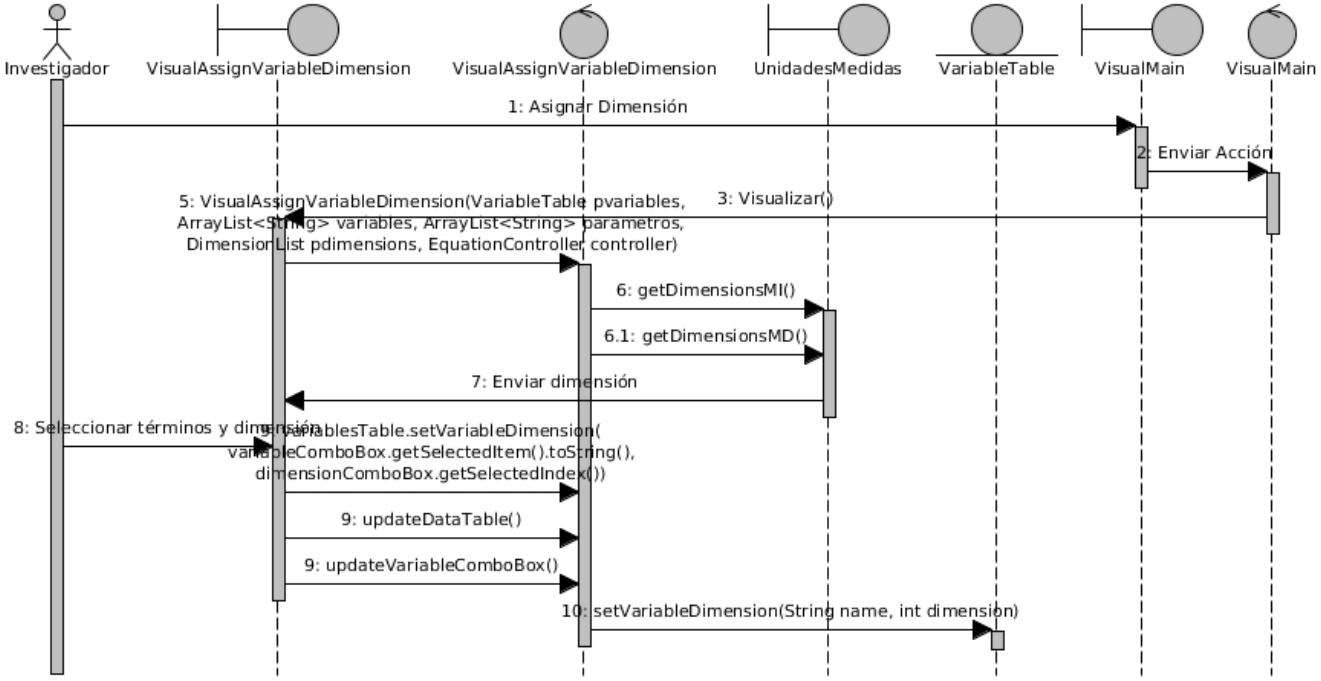

Figura 11.1: Diagrama de Secuencia del Caso de Uso Asignar Dimensión del editor de ecuaciones del software alasBioSyS v1.2.

3.5.2 Diagrama de Secuencia del Caso de Uso Verificar Homogeneidad Dimensional

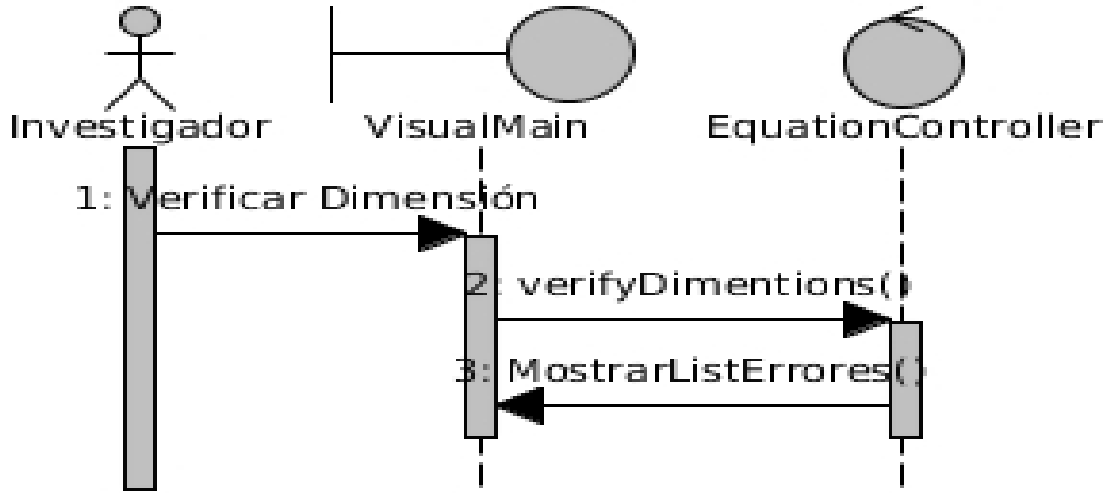

Figura 12.1: Diagrama de Secuencia del Caso de Uso Verificar Homogeneidad Dimensional del editor de ecuaciones del software alasBioSyS v1.2.

3.5.3 Diagrama de Secuencia del Caso de Uso Modificar Fuente

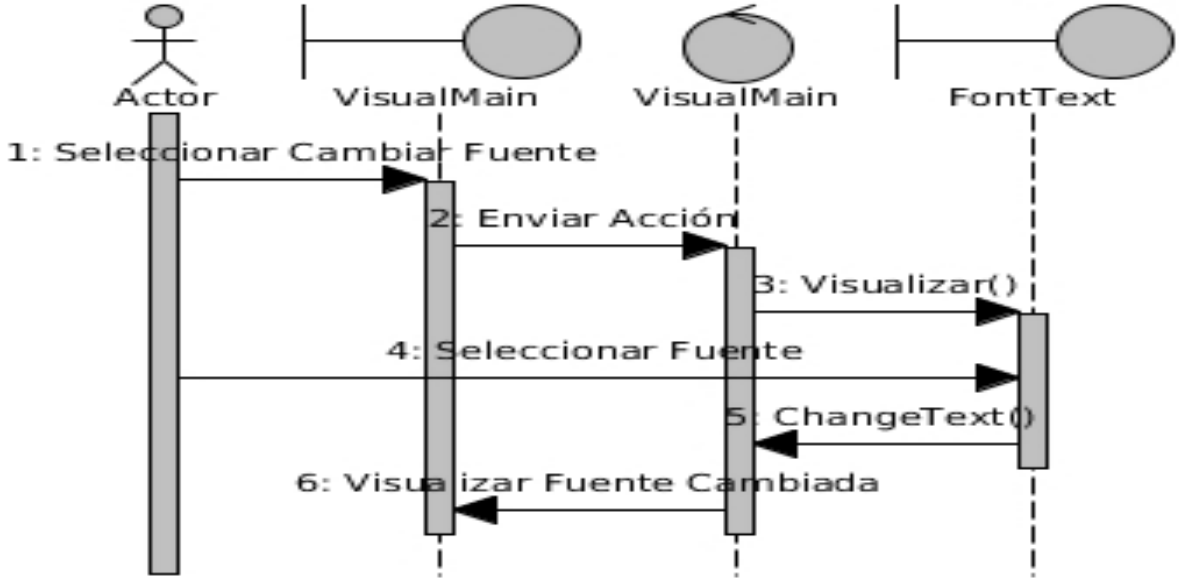

Figura 13.1: Diagrama de Secuencia del Caso de Uso Modificar Fuente del editor de ecuaciones del software alasBioSyS v1.2.

3.6. Modelo de Despliegue

El modelo de despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

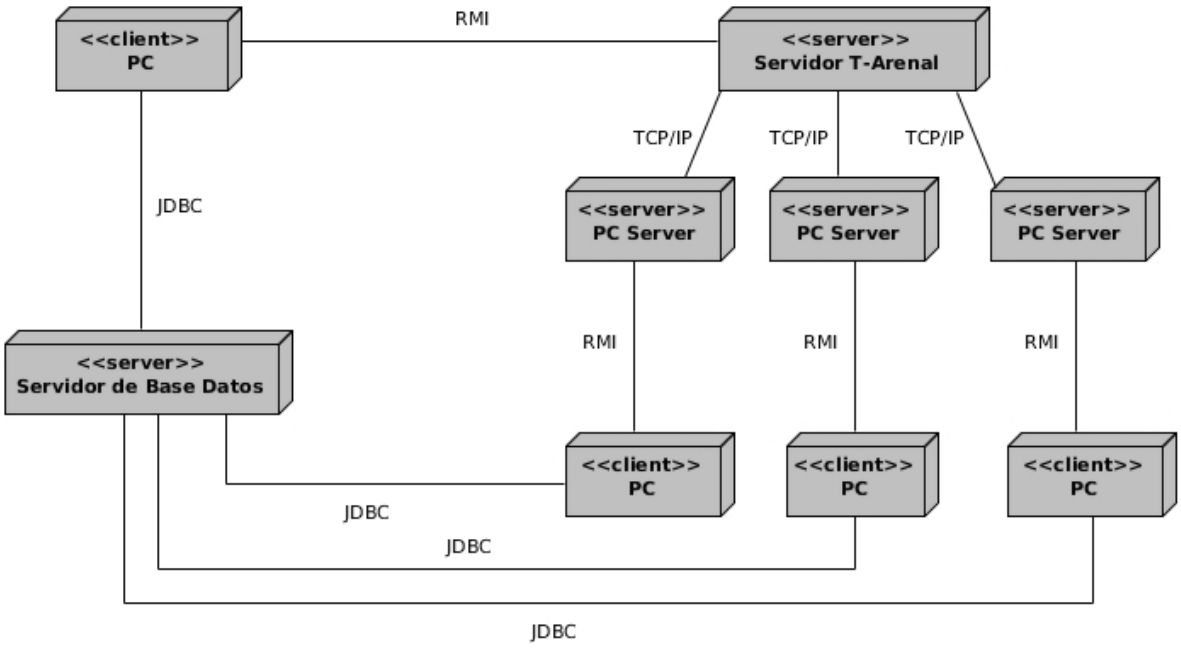

Figura 14.1: Diagrama de despliegue del software alasBioSyS

En la figura 3.8, se observa un modelo en el cual una PC cliente se conecta utilizando el protocolo JDBC al servidor de Base de Datos, además se conecta con la plataforma de cálculo distribuido T-arenal por el protocolo RMI para realizar las simulaciones de forma distribuidas. A su vez esta plataforma se conecta con varias PC para distribuir las tareas que se le asignen, las cuales están conectadas al servidor de Base de Datos por el protocolo JDBC.

3.7. Conclusiones

- Para la implementación de la aplicación se escogió el patrón arquitectónico Modelo-Vista-Controlador.
- En el diseño de la aplicación se utilizaron diferentes patrones de diseño como: Creador, Alta Cohesión, Bajo Acoplamiento.
- Se diseñaron las clases y prototipos correspondientes para dar paso a la implementación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

En este capítulo se describe la implementación del sistema, mostrando los diagramas de componentes de las funcionalidades desarrolladas. Además se presenta todo lo relacionado con las pruebas realizadas al sistema.

4.2. Implementación

4.2.1 Diagramas de componentes

En los diagramas de componentes se muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Se puede usar un diagrama de componentes para describir un diseño que se implemente en cualquier lenguaje o estilo. Solo es necesario identificar los elementos del diseño que interactúan con otros elementos del diseño a través de un conjunto restringido de entradas y salidas. Los componentes pueden tener cualquier escala y pueden estar interconectados de cualquier manera. (22)

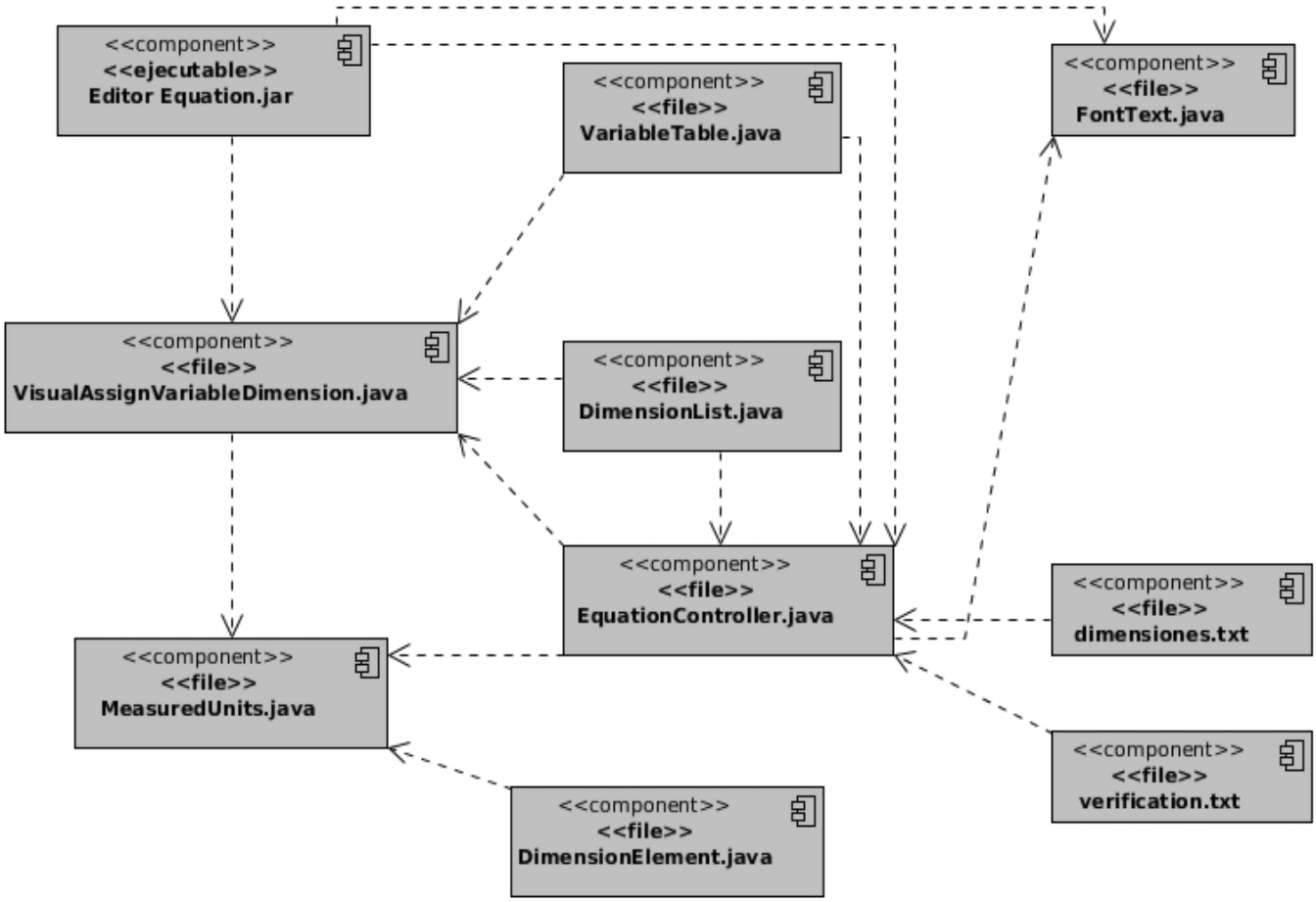

Figura 15.1: Diagrama de componentes del editor de ecuaciones del software alasBioSyS v1.2

A continuación se describen cada uno de los componentes mostrados en la Figura 15.1.

Editor Equation: Representa el ejecutable de la aplicación.
VisualAssignVariableDimension: Representa una clase interfaz en la cual el investigador selecciona la variable a la cual desea asignarle alguna dimensión.

MeasuredUnits: Representa una clase interfaz con las magnitudes definidas internacionalmente donde el investigador selecciona la dimensión que desea para posteriormente asignarla a una variable en la clase VisualAssignVariableDimension.

VariableTable: Representa la clase que se encarga de almacenar todas las variable que se escriban en el editor de ecuaciones.

DimensionList: Representa la clase que se encargara de almacenar la dimensión correspondiente a cada variable.

EquationController: Representa la clase que manejara todas las funcionalidades de la versión 1.2 del editor de ecuaciones del software alasBioSyS.

DimensionElement: Representa la clase que se encargara de crear objetos de tipo dimensión con el nombre de la dimensión y su fórmula correspondiente.

FontText: Representa una interfaz que le permitirá al investigador modificar la fuente del editor a su gusto.

Dimensión.txt: Representa un fichero que se crea para ayudar con el análisis dimensional el cual guarda después de haber aplicado un proceso la simplificación de una ecuación.

Verification.txt: Representa un fichero que se crea para ayudar con el análisis dimensional el cual guarda si dos expresiones son iguales.

4.2.2. Imágenes de la Aplicación

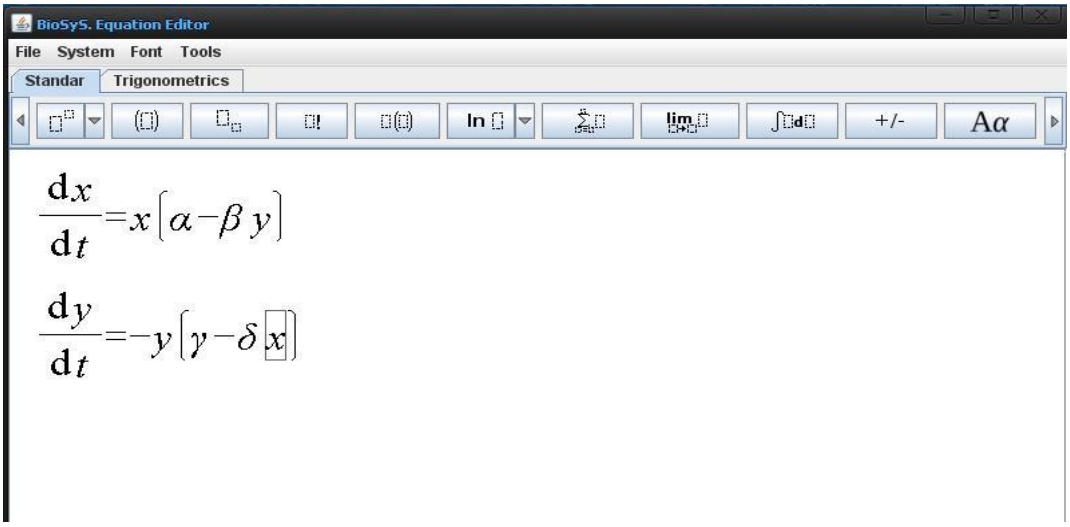

Figura 16.1: Editor de Ecuaciones

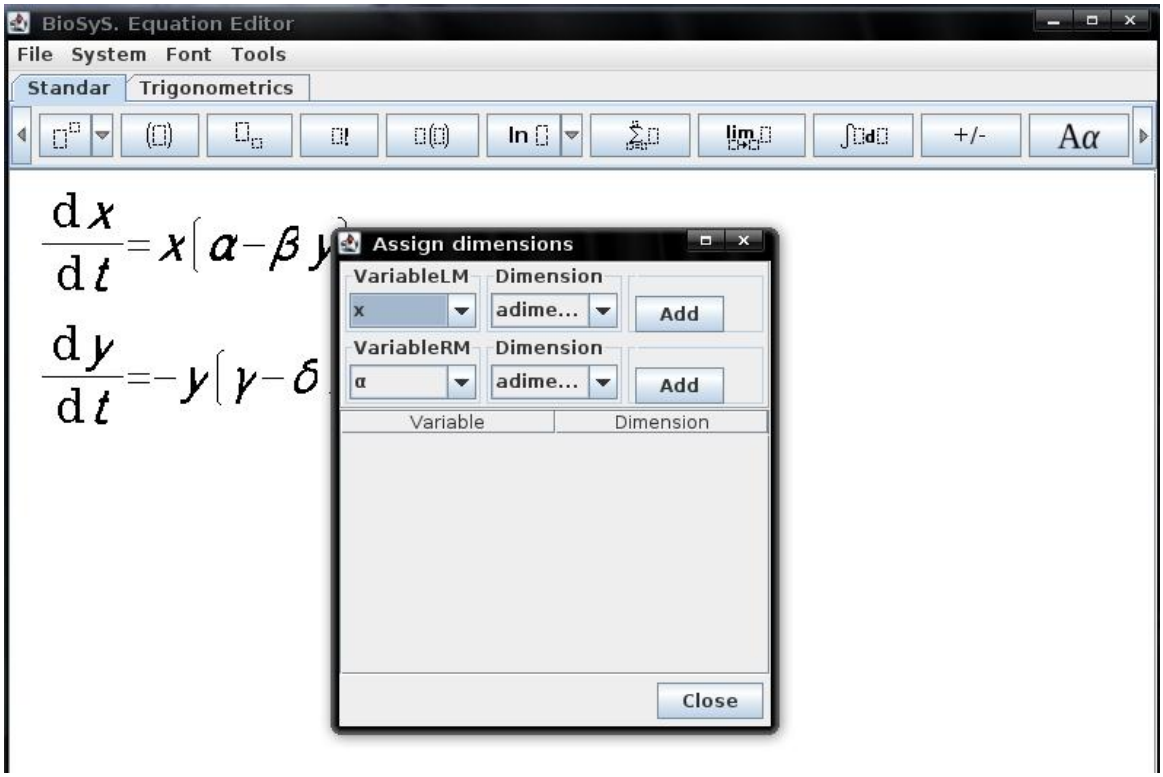

Figura 17.1: Visual para asignar las dimensiones.

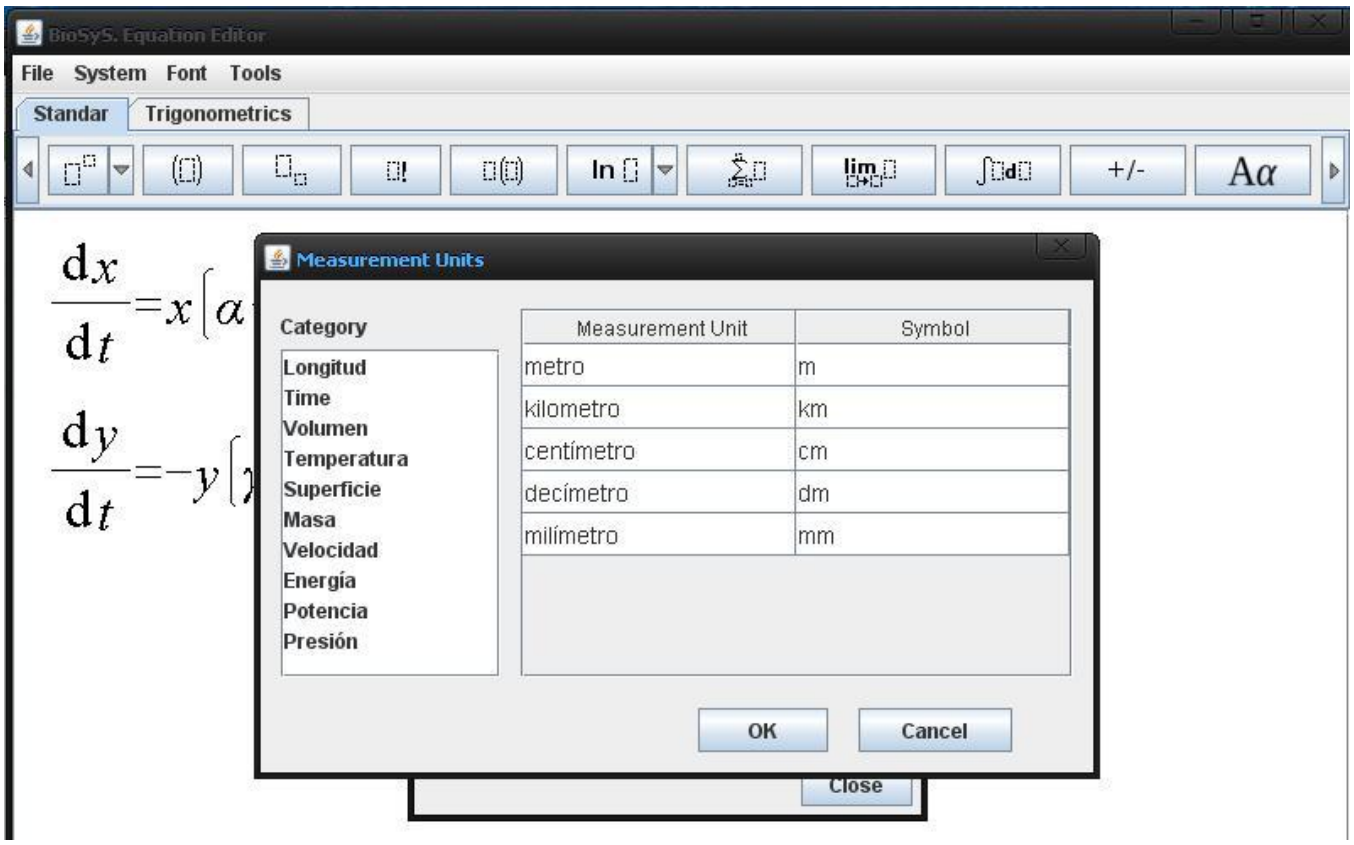

Figura 18.1: Visual para seleccionar las dimensiones que posterior mente se asignaran.

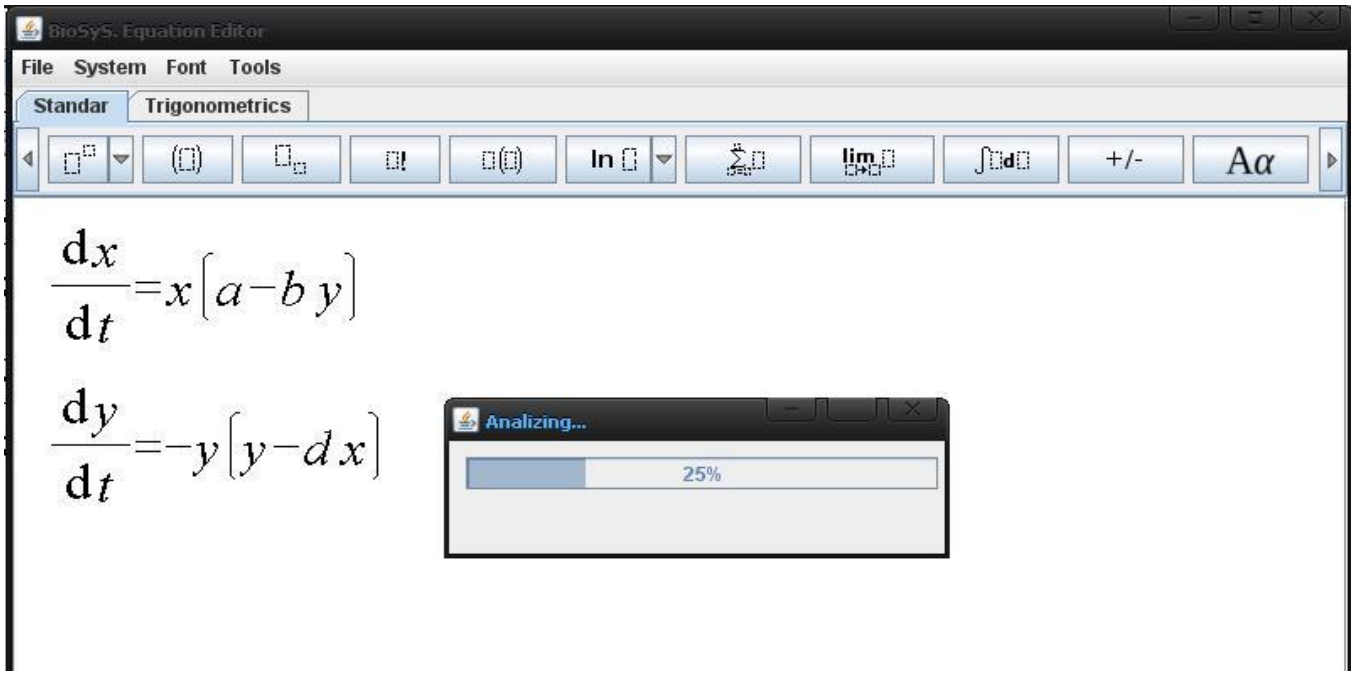

Figura 19.1: Visual mostrando el proceso del análisis de homogeneidad dimensional.

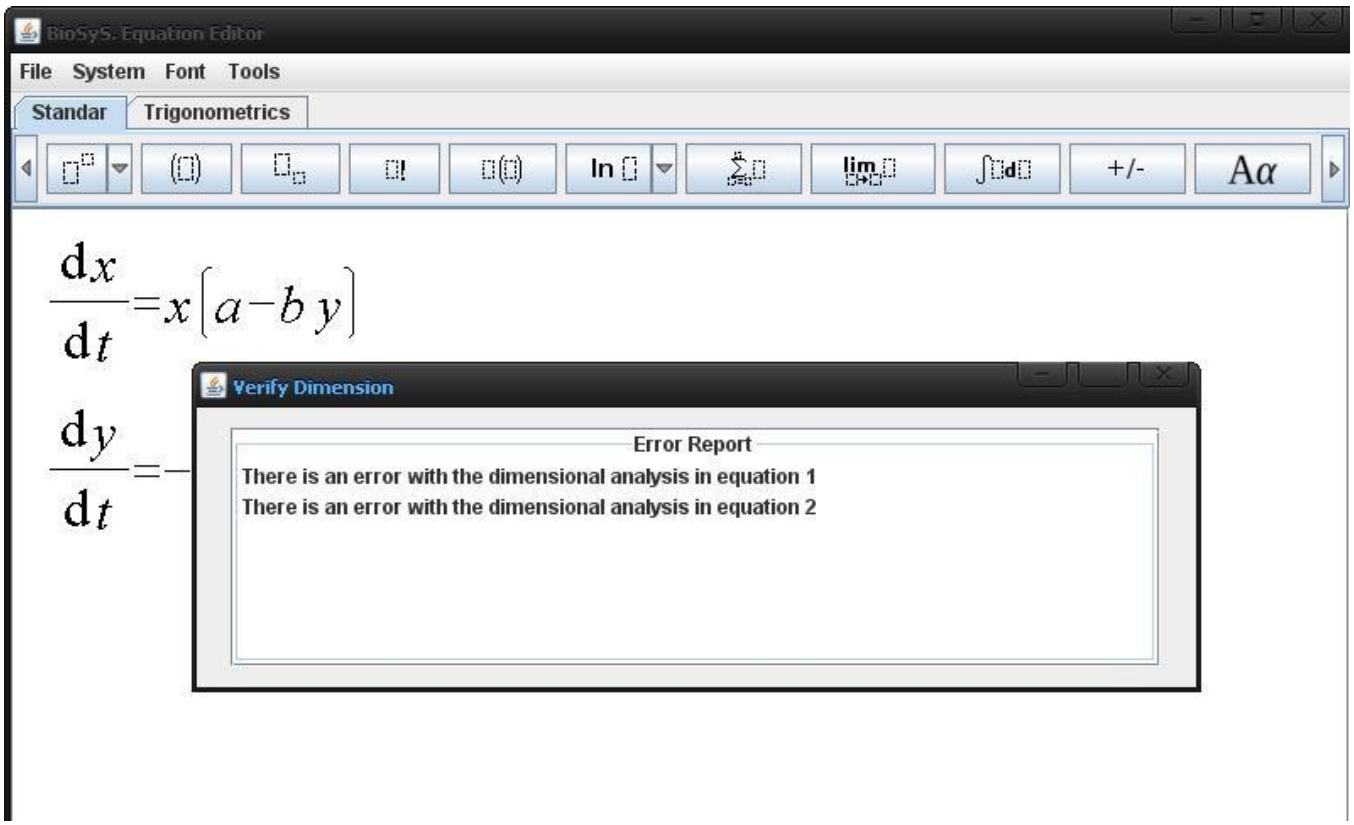

Figura 20.1: Reporte para visualizar donde existen errores en el sistema.

4.3. Pruebas del Sistema

Las pruebas de software son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. (23)

4.3.1. Plan de Prueba

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe dejar claro: qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad); cómo se mide el resultado; especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta) y definir cuál es el resultado que se espera. En lo adelante se abordara todo lo relacionado con las pruebas realizadas al sistema. (24)

4.3.1.1. Configuración del entorno de Prueba

La configuración del entorno donde se vayan a ejecutar las diferentes pruebas que se realizan a un software es un aspecto muy importante dentro del proceso de pruebas, pues si no se analizan bien los recursos de software y hardware que necesita el producto que se está construyendo, a la hora de probarlo se prescindirá de los elementos necesarios para la ejecución de un proceso de pruebas exitoso. Por ello se tuvo en cuenta algunos requerimientos de hardware y de software que hicieron posible un mejor desarrollo de las pruebas, en aras de lograr minimizar los errores de la aplicación en desarrollo. Los requerimientos que se consideraron necesarios para las pruebas se referencian a continuación: (24)

Requerimientos de software:

- PC con Windows XP o superior.
- PC con Linux.
- Máquina Virtual de Java 1.6.
- Asistente matemático: Maxima

Requerimientos de hardware:

- PC con Microprocesador Pentium IV o superior.
- 1 GB de memoria RAM o superior.

Fecha de Inicio	Fecha de fin	Actividades	Personal Implicado
03/05/2012	03/05/2012	Aceptación y firma del Plan de Prueba.	Ing. Carlos González Iglesias Ariel Pérez Rueda
03/05/2012	03/05/2012	Verificación de las condiciones previas para el inicio de las pruebas.	Ing. Carlos González Iglesias Ariel Pérez Rueda
Primera Iteración			

06/05/2012	06/05/2012	Ejecución de las pruebas de caja negra al caso de uso Asignar Dimensión.	Ariel Pérez Rueda
07/05/2012	07/05/2012	Ejecución de las pruebas de caja negra al caso de uso Verificar Homogeneidad Dimensional.	Ariel Pérez Rueda
08/05/2012	08/05/2012	Ejecución de las pruebas de caja negra al caso de uso Modificar Fuente.	Ariel Pérez Rueda
09/05/2012	03/06/2012	Corregir las no conformidades detectadas al aplicar las pruebas de cajas negras.	Ariel Pérez Rueda

4.3.2. Diseño de las Pruebas de Caja Negra

Las pruebas de caja negra son aquellas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene. (25)

4.3.2.1. Diseño de las pruebas.

Caso de Uso: Asignar Dimensión.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Asignar Dimensión.	**EC 1.1:** Asignar dimensión a los términos	El investigador selecciona la opción “Assign dimensions”. El sistema muestra una ventana que le permitirá al investigador asignarle las dimensiones a	Selecciona la opción “Assign dimensions”.

	del miembro derecho de las ecuaciones del modelo.	los términos del miembro derecho del modelo matemático, selecciona el término y la dimensión a asignar y presiona el botón Add. El sistema asigna la dimensión seleccionada al término correspondiente.	
SC 1: Asignar Dimensión.	**EC 1.2:** Asignar dimensión a los términos del miembro izquierdo de las ecuaciones del modelo	El investigador selecciona la opción “Assign dimensions”. El sistema muestra una ventana que le permitirá al investigador asignarle las dimensiones a los términos del miembro izquierdo del modelo matemático, selecciona el término y la dimensión a asignar y presiona el botón Add. El sistema asigna la dimensión seleccionada al término correspondiente.	Selecciona la opción “Assign dimensions”.

SC 1: Asignar Dimensión.

Variables

1.- Ecuación en el editor: x=y+f

2.- Miembro de la ecuación: Derecho

3.- Términos del miembro de la ecuación: y, f

Id del escenario	Escenario	V1	V2	V3	Respuesta del sistema	Resultado de la prueba
EC 1.1	Asignar dimensión a los términos del miembro derecho de las ecuaciones del modelo.	V	V	V	El sistema muestra una ventana que le permitirá al investigador asignarle las dimensiones a los términos del miembro derecho del modelo matemático.	Se le asignaron las dimensiones a los términos del miembro derecho de la ecuación.

Variables

1.- Ecuación en el editor: x=y+f

2.- Miembro de la ecuación: Izquierdo

3.- Términos del miembro de la ecuación: x

Id del escenario	Escenario	V1	V2	V3	Respuesta del sistema	Resultado de la prueba
EC 1.2	Asignar dimensión a los términos del miembro izquierdo de las ecuaciones del modelo.	V	V	V	El sistema muestra una ventana que le permitirá al investigador asignarle las dimensiones a los términos del miembro izquierdo del modelo matemático.	Se le asignaron las dimensiones a los términos del miembro izquierdo de la ecuación.

Variables

1.- Ecuación en el editor: x=y+f

2.- Miembro de la ecuación: Derecho

3.- Términos del miembro de la ecuación: y, f

Id del escenario	Escenario	V1	V2	V3	Respuesta del sistema	Resultado de la prueba
EC 1.2	Asignar dimensión a los términos del miembro izquierdo de las ecuaciones del modelo.	V	V	F	El sistema muestra una ventana de error “Errors in the model, check them”.	Se muestra una ventana de error “Errors in the model, check them”.

La descripción de las variables definidas para el caso de uso Asignar dimensión se encuentra en el anexo 1.

Caso de Uso: Verificar Homogeneidad Dimensional.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Verificar Homogeneidad Dimensional	**EC 1.1:** Verificar Homogeneidad Dimensional	El investigador selecciona la opción “Check dimensions”. El sistema muestra una ventana de diálogo con un mensaje de que el modelo no tiene errores dimensionalmente o en caso de que exista alguno en que ecuación es.	Selecciona la opción “Check dimensions”

SC 1: Verificar Homogeneidad Dimensional

Variables

1.- Ecuación en el editor: x=y+f

2.- Dimensiones de los términos de la ecuación: m=m+s

3.- Asistente matemático: Maxima

Id del escenario	Escenario	V1	V2	V3	Respuesta del sistema	Resultado de la prueba
EC 1.1	Verificar Homogeneidad Dimensional.	V	V	V	El sistema muestra una ventana de diálogo con un mensaje de que el modelo no tiene errores dimensionalmente, ”There are no errors in the system of equations”.	Se muestra una ventana de diálogo con un mensaje de que el modelo no tiene errores dimensionalmente, ”There are no errors in the system of equations”.

Variables

1.- Ecuación en el editor: x=y+f

2.- Dimensiones de los términos de la ecuación: m=m+s

3.- Asistente matemático: Maxima

Id del escenario	Escenario	V1	V2	V3	Respuesta del sistema	Resultado de la prueba
EC 1.1	Verificar Homogeneidad Dimensional.	V	F	V	El sistema muestra una ventana de diálogo con un mensaje de que se le deben asignar todas las dimensiones a los términos del modelo, ”Errors in the model, make sure all terms are assigned their size”.	Se muestra una ventana de diálogo con un mensaje de que se le deben asignar todas las dimensiones a los términos del modelo, ”Errors in the model, make sure all terms are assigned their size”.

La descripción de las variables definidas para el caso de uso Verificar Homogeneidad Dimensional se encuentra en el anexo 2.

Caso de Uso: Modificar Fuente

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1: Modificar Fuente	**EC 1.1:** Modificar Fuente	El investigador selecciona la opción “Change font”. El sistema muestra una ventana donde le permite al investigador cambiar la fuente del editor. El investigador modifica la fuente a su criterio y presiona el botón	Selecciona la opción “Change font”.

		"Aceptar". El sistema cambia la fuente del editor.	

SC 1: Modificar Fuente

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EC 1.1	Modificar Fuente.	El sistema muestra una ventana donde le permite al investigador cambiar la fuente del editor.	El investigador modifica la fuente a su criterio y presiona el botón "Aceptar". El sistema cambia la fuente del editor.

4.3.2.4. No Conformidades detectadas

Elemento	No	No Conformidad	Aspecto Correspondiente	Etapa de detección	Signif	No Signif
Implementación	1	Error al levantar la interfaz VisualAssignVariableDimension	Existía error en el modelo descrito.	Realización de las Pruebas de caja negra.	X	
Implementación	2	Error al verificar el análisis dimensional.	No se habían asignado todas las dimensiones a las variables del modelo.	Implementación y Prueba.	X	
Implementación	3	Error al	No se tenía el	Implementació	X	

		verificar el análisis dimensional.	Maxima como asistente matemático instalado en al PC.	n y Prueba.		
Interfaz	4	No se modificaba la fuente del editor de ecuaciones	No se estaban teniendo en cuenta la conformación de las ecuaciones en el editor	Implementació n y Prueba.		X
Interfaz	5	No se recogían las variables de las ecuaciones tipos función.	No se estaba teniendo en cuenta ese tipo de ecuación en el desarrollo	Implementació n y Prueba.	X	

4.4. Conclusiones

- Se logró implementar las funcionalidades necesarias para una versión 1.2 del software alasBioSyS con éxito.
- Se le realizaron pruebas de caja negra al sistema y se detectaron 5 no conformidades las cuales fueron registradas y solucionadas para un mejor funcionamiento del software.

CONCLUSIONES

- Se desarrolló la versión 1.2 del editor de ecuaciones del software alasBioSyS.
- Se diseñaron e implementaron las funcionalidades necesarias para brindar un mayor nivel de análisis de los sistemas que se estudien.
- Se validó la aplicación a través de pruebas de caja negra, detectándose solamente 5 no conformidades, lo cual demuestra el buen funcionamiento del software.

RECOMENDACIONES

- Agregar al editor de ecuaciones del software alasBioSyS la funcionalidad de exportar al formato LATEX.
- Realizar un estudio continuo sobre las magnitudes físicas existentes que no están en el editor de ecuaciones del software alasBioSyS para incorporarlas a este.
- Optimizar la funcionalidad de importar modelos del editor de ecuaciones del software alasBioSyS para disminuir el tiempo de espera.

BIBLIOGRAFÍA

1. Naranjo, L. G. (2008). *Editor de ecuaciones para la Plataforma.* Universidad de las Ciencias Informáticas.

2. *Fisica Facil*. (Enero de 2002). Recuperado el Septiembre de 2011, de http://fisicafacil.awardspace.com/tema00.html

3. Guevara Galdos, D. (Julio de 2004). *Física Conceptual.* Recuperado el Octubre de 2011, de www.antorai.com.pe/fisica/index.html

4. *Dirección Nacional de Servicios Académicos Virtuales*. (s.f.). Recuperado el Noviembre de 2011, de Universidad Nacional de Colombia: http://www.virtual.unal.edu.co/cursos/sedes/manizales/4090002/html/pages/cap2/c2_7.htm

5. Java Hispano. *Java2, tutorial de javahispano.* [Online] [Cited: 11 03, 2011.] ies- http://ies-jaumebalmes.xtec.net/montse/pav/Manual de Swing.pdf

6. **Gurley, Greg and Oster, Nate.** ATSC. *ATSC.* [Online] 06 05, 2008. [Cited: 11 03, 2011.] http://www.atsc.com/documents/briefings/20080605-atsc-agileopenup-presentation.pdf. 16.

7. **Gastón.** Club de Desarrolladores. *clubdesarrolladores.* [Online] 03 01, 2009. [Cited: 11 03, 2011.] http://www.clubdesarrolladores.com/articulos/mostrar/68-uml-definicion-historia-y-especificaciones.

8. **Sierra, María.** *Trabajando con Visual Paradigm for UML*. Univ. Cantabria – Fac. de Ciencias, España: s.n, http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf.

9. Lenguajes de Programación. *Lenguajes de Programación.* [Online] 2009. [Cited: 11 03, 2011.] http://www.lenguajes-de-programacion.com/programacion-java.shtml.

10. **Mendez, Justo.** monografias.com. *monografias.com.* [Online] Junio 2011. [Cited: 11 03, 2011.] http://www.monografias.com/trabajos/lengprog/lengprog.shtml.

11. **Corporation, Oracle.** NetBeans. *NetBeans.* [Online] 2011. [Cited: 11 03, 2011.] http://netbeans.org/community/releases/69/.

12. **Jaque, Miguel.** Modelo de Dominio. [Online] 04 27, 2008. [Cited: 01 15, 2012.] http://migueljaque.com/index.php/tecnicas/tecnicasmodnegocio/37-modelado_negocio/46-modelo-de-dominio?tmpl=component&print=1&page=.

13. **RAMIREZ, ALEJANDRO FARIAS and GONZALEZ, EDUARDO LUNA.** Scribd. *Scribd.* [Online] [Cited: 01 15, 2012.] http://es.scribd.com/doc/58135266/18/Definicion-de-Requisitos-Funcionales.

14. The War of Software. *The War of Software.* [Online] 11 16, 2006. [Cited: 01 15, 2012.] http://zoftwar.blogspot.com/2006/11/requisitos-no-funcionales-parte-1-con.html.

15. Beshenov, L. (s.f.). *Maxima, un sistema de álgebra computacional*. Recuperado el 11 de 2011, de Maxima, un sistema de álgebra computacional: http://maxima.sourceforge.net/es/

16. **Rojas, Sara Ramírez.** Casos de Uso. [Online] 03 01, 2009. [Cited: 01 15, 2012.] http://exposicioncasosdeusouml.blogspot.com/2009/03/definicion.html.

17. Wesley Longman, A. (1992). *Ingeniería de Software Orientada a Objetos - Un acercamiento a través de los casos de uso.*

18. **Tello, Jesús Cáceres.** Diagramas de Casos de Uso. [Online] [Cited: 01 15, 2012.] http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf.

19. Eckstein, R., Loy, M., & Wood, D. (1998). Java Swing. En O. Reilly, *Java Swing* (pág. 22). United States of America: Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

20. Rios, S. S. (04 de 2011). *Principales patrones - GRASP*. Recuperado el 04 de 2012, de Principales patrones - GRASP: http://www.slideshare.net/SergioRios/unidad-9-patrones-de-diseo

21. *Sistemas de Programas*. (1999). Recuperado el 03 de 2012, de Patrones de Diseño: http://ldc.usb.ve/~teruel/ci3711/patron3a/index.html#experto

22. *MSDN*. (s.f.). Recuperado el 10 de 05 de 2012, de Microsoft Developer Network Platforms: http://msdn.microsoft.com/es-es/library/dd409390.aspx

23. León Carrillo, L. V. (MAY-JUN de 2005). *www.softwareguru.com.mx.* Recuperado el 05 de 2012, de PRUEBA DE SOFTWARE: http://www.e-quallity.net/articulos/SG-200503-Luis03.pdf

24. *MSDN*. (s.f.). Recuperado el 10 de 05 de 2012, de Microsoft Developer Network Platforms: http://msdn.microsoft.com/es-es/library/ dd286583.aspx

25. Mañas, J. A. (Marzo de 1994). *Prueba de Programas*. Recuperado el Mayo de 2012, de http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm

ANEXOS

Anexo 1: Descripción de las variables del caso de uso Asignar Dimensión.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Ecuación en el editor	Campo de texto	No	x=y+f: Variable que describe una ecuación en el editor.
2	Miembro de la ecuación	Campo de texto	No	Derecho y Izquierdo: Variable que describe la conformación de una ecuación.
3	Términos del miembro de la ecuación	Campo de texto	No	x, y, f: Describe las variable que conforman una ecuación.

Anexo 2: Descripción de las variables del caso de uso Verificar Homogeneidad Dimensional.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Ecuación en el editor	Campo de texto	No	x=y+f: Variable que describe una ecuación en el editor.
2	Dimensiones de los términos de la ecuación	Magnitud física	No	m, s, kg: Variable que describe las dimensiones que se le puede asignar a las variables de una ecuación.
3	Asistente matemático	Asistente matemático	No	Maxima: Variable que describe si se encuentra el Maxima instalado en la PC.

1