

**Universidad de las Ciencias Informáticas**

**Facultad 6**



**Título: “Editor de Ecuaciones del Simulador de  
Sistemas Biológicos BioSyS, versión 2.0.”**

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autora:** Mirian Caridad Sáez Rodríguez

**Tutores:** Ing. Edel Moreno Lemus

Ing. Vilmavis La Rosa Sordo

**Consultantes:** Ing. Osvel Chávez Hernández

Ing. Andry Daniel Díaz León

**JUNIO 2013**

*“Mientras seamos persistentes en nuestra búsqueda de un destino más profundo, seguiremos creciendo. No podemos escoger el día o el momento en que floreceremos plenamente; eso ocurrirá en su propio tiempo.”*

Denis Waitley

## DECLARACIÓN DE AUTORÍA

Declaro ser la autora de la presente tesis y reconozco 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 \_\_\_\_\_.

**Mirian Caridad Sáez Rodríguez**

\_\_\_\_\_

Firma de la Autora

**Vilmavis La Rosa Sordo**

\_\_\_\_\_

Firma de la Tutora

**Edel Moreno Lemos**

\_\_\_\_\_

Firma del Tutor

## DATOS DE CONTACTO

### **Tutores:**

Ing. Vilma La Rosa Sordo

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: [vlarosa@uci.cu](mailto:vlarosa@uci.cu)

Ing. Edel Moreno Lemus

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: [emoreno@uci.cu](mailto:emoreno@uci.cu)

### **Consultantes:**

Ing. Osvel Chávez Hernández

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: [ochavez@uci.cu](mailto:ochavez@uci.cu)

Ing. Andry Daniel Díaz León

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: [addiaz@uci.cu](mailto:addiaz@uci.cu)

## **AGRADECIMIENTO**

Tengo que agradecer a tantas personas, que han hecho posible mis sueños, espero que no se me quede ninguna, en caso de que suceda, les pido que no se preocupen, porque las tengo escritas en el papel del alma.

Primeramente les agradezco a mi mamá María Luisa y a mi hermana Marisleidys, por todo el apoyo, sacrificio y dedicación que me han dado a lo largo de mi vida. A mi papá Sáez, por su apoyo y por creer siempre en mí. A Vicky, por enseñarme que siempre todos tenemos la oportunidad de lograr nuestros sueños. A mi primo Maikel, por estar siempre pendiente de mí, así como mis primos Lachy y Dunia. A mis tías Olgui, Belkis y en especial a Laude, que nunca han dejado de ayudarme, al igual que mi abuela Miriam. A toda mi familia en general.

De los amigos, que están lejos y que por muchos motivos no se encuentran cerca, hago una pausa especial para agradecer a mi amiga Meilyn que nunca ha dejado de apoyarme y todos los días me da fuerzas para seguir adelante. Gracias por creer en mí hasta los días de hoy, aun cuando muchos pensaban que no lo podía lograr.

A mis amigos de Sagua: Jaren, Rafael, Yasiel, José Oscar, Marbelys, Isidro y Jaile. A mis mejores amigos de la UCI: Keimer, Yonatan, Matyuri, Maikel, Elizabeth, Tony, Anabel, el Chino, Carmen, Abelito, Adrian, Dayana; gracias por tolerarme todos estos años, por siempre estar conmigo cuando más lo he necesitado y por demostrarme que no podemos dejar de batallar, que hay que echar para adelante con ganas. Quiero agradecer a Sergio por darme su apoyo incondicional desde el primer día que me conoció, hasta estos días, al igual que su mamá Tere.

A Osvel y Andrys por brindarme sus manos cada vez que lo necesitaba. A mi tutora Vilma a pesar de no haber estado conmigo desde el comienzo, le tengo que dar las gracias por todo su apoyo y dedicación. Quiero agradecer también, a la persona que más me ha ayudado en el desarrollo de la tesis, aun estando de misión en Venezuela, mi tutor Edel.

A mis amistades de convivencia Tony, Isenith, Darlys, Dalilys, Odalys, Jose, Yanet, Alejandro, Yudith, Dayanni, Leonel, Pablo, Yuliet. Y otros amigos que conocí durante la carrera: Alexo, Robert, Osniel, Erick, Fernando, Ernesto, Luis Miguel, Jony, Dayan, Javier, Yoko, Miguel Angel y todos los demás que no he mencionado. Gracias por hacerme sentir la vida más amena en la UCI.

A los que me hicieron sentir mal en algún momento, gracias por hacerme más fuerte.

## **DEDICATORIA**

Esta tesis va dedicada a una persona especial, a quien yo le tengo que agradecer quien soy y cuanto he hecho en mi vida, porque ha logrado que se hagan realidad todos mis sueños, porque siempre está conmigo, en cualquier tipo de situación y nunca ha dejado de extenderme su mano y decirme “yo estoy contigo hija”. Ella es mi mamá María Luisa; le dedico todo mi amor y cariño; mi vida, si algún día es preciso.

## RESUMEN

La Biología de Sistemas es una disciplina que surge a raíz de los modernos métodos analíticos de alto rendimiento, para analizar las funciones celulares a través de conjuntos complejos de componentes y funciones, viendo a la célula como un todo. Su creación introdujo la utilización de herramientas computacionales para modelar sistemas biológicos y realizar estudios sobre sus comportamientos. Una de estas herramientas es la Plataforma para la Simulación y Análisis de Sistemas Biológicos, BioSyS, la cual permite realizar un estudio y análisis de modelos de sistemas biológicos representados por sistemas de ecuaciones diferenciales. BioSyS cuenta con un Editor de Ecuaciones que posibilita la realización de cálculos numéricos y análisis matemático. Dicho Editor presenta problemas con el tiempo de respuesta cuando se editan modelos. En el presente trabajo se muestra un nuevo Editor de Ecuaciones que permite disminuir la demora en el tiempo de respuesta durante la edición de modelos, con el objetivo de proporcionar a los investigadores una herramienta computacional, con una interfaz más rápida y sencilla, que cumpla con las funcionalidades necesarias para brindar un mejor servicio al simular y analizar los modelos de sistemas biológicos.

## PALABRAS CLAVE

Biología de Sistemas, Software, Sistemas Biológicos, Ecuaciones Diferenciales Ordinarias, Editor de Ecuaciones

# ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 EDITORES DE ECUACIONES.....	5
1.2 BIOSYS.....	7
1.3 NECESIDAD DE LA CREACIÓN DEL EDITOR DE ECUACIONES DE BIOSYS, VERSIÓN 2.0.....	8
1.4 PLATAFORMA DE SERVICIOS BIOINFORMÁTICOS.....	8
1.5 APPLET JAVA .....	9
1.6 PORTLET .....	10
1.6.1 Liferay Portal 6.0.5 .....	11
1.6.2 Spring .....	11
1.7 METODOLOGÍA Y HERRAMIENTAS UTILIZADAS.....	12
1.7.1 Metodología de desarrollo de software .....	12
1.7.2 Lenguaje de programación.....	13
1.7.3 Herramientas de desarrollo de software.....	14
1.7.4 Herramienta CASE .....	15
1.7.5 Lenguaje de modelado .....	16
1.7.6 Lenguaje de marcado .....	16
1.8 CONCLUSIONES DEL CAPÍTULO .....	17
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>18</b>
2.1 BREVE DESCRIPCIÓN DEL SISTEMA .....	18
2.2 MODELO CONCEPTUAL.....	18
2.2.1 Representación del Modelo Conceptual.....	18
2.3 ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA.....	19
2.3.1 Requisitos Funcionales .....	19
2.3.2 Requisitos No Funcionales.....	20
2.4 DEFINICIÓN DE CASOS DE USO .....	22
2.4.1 Actores del Sistema .....	22
2.4.2 Diagrama de Casos de Uso del Sistema .....	23
2.4.3 Descripción de Casos de Uso del Sistema.....	23
2.5 CONCLUSIONES DEL CAPÍTULO .....	30
<b>CAPÍTULO 3. DISEÑO DE LA APLICACIÓN</b> .....	<b>31</b>
3.1 PATRÓN ARQUITECTÓNICO UTILIZADO .....	31



3.2	PATRONES DE DISEÑO UTILIZADOS .....	32
3.3	MODELO DE DISEÑO .....	36
3.3.1	<i>Diagrama de Clases del Diseño</i> .....	37
3.3.2	<i>Diagramas de Secuencia</i> .....	38
3.3.3	<i>Modelo de Despliegue</i> .....	40
3.4	CONCLUSIONES DEL CAPÍTULO .....	40
<b>CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA .....</b>		<b>41</b>
4.1	IMPLEMENTACIÓN .....	41
4.1.1	<i>Diagrama de Componentes</i> .....	41
4.1.2	<i>Imágenes del Editor de Ecuaciones, versión 2.0</i> .....	43
4.2	PRUEBAS DE SOFTWARE .....	45
4.2.1	<i>Plan de Prueba</i> .....	46
4.2.2	<i>Diseño de las Pruebas Caja Blanca</i> .....	48
4.2.3	<i>Diseño de las Pruebas Caja Negra</i> .....	51
4.2.4	<i>Método Experimental</i> .....	55
4.3	CONCLUSIONES DEL CAPÍTULO .....	56
<b>CONCLUSIONES GENERALES .....</b>		<b>57</b>
<b>RECOMENDACIONES .....</b>		<b>58</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>59</b>
<b>BIBLIOGRAFÍA .....</b>		<b>61</b>
<b>ANEXOS.....</b>		<b>63</b>
<b>GLOSARIO .....</b>		<b>68</b>

## ÍNDICE DE FIGURAS

<i>Figura 1 Editor de Ecuaciones de BioSys, versión 1.2.</i> .....	3
<i>Figura 2 Estructura de un portlet.</i> .....	10
<i>Figura 3 Modelo Conceptual.</i> .....	18
<i>Figura 4 Diagrama de Casos de Uso del Sistema.</i> .....	23
<i>Figura 5 Patrón MVC para la versión 2.0 del Editor de Ecuaciones.</i> .....	32
<i>Figura 6 Ejemplo de la aplicación del Patrón Experto.</i> .....	32
<i>Figura 7 Ejemplo de la aplicación del Patrón Creador.</i> .....	33
<i>Figura 8 Ejemplo de la aplicación del Patrón Controlador.</i> .....	34
<i>Figura 9 Ejemplo de la aplicación del Patrón Alta Cohesión.</i> .....	35
<i>Figura 10 Ejemplo de la aplicación del Patrón Bajo Acoplamiento.</i> .....	35
<i>Figura 11 Ejemplo de la aplicación del Patrón Fachada.</i> .....	36
<i>Figura 12 Ejemplo de la aplicación del Patrón Observador.</i> .....	36
<i>Figura 13 Diagrama de Clases del Diseño.</i> .....	37
<i>Figura 14 Diagrama de Secuencia del CU Gestionar SED (sección Adicionar nueva ecuación).</i> .....	38
<i>Figura 15 Diagrama de Secuencia del CU Gestionar SED (sección Insertar identificador o constante numérica).</i> .....	39
<i>Figura 16 Diagrama de Secuencia del CU Gestionar SED (sección Insertar operación matemática).</i> .....	39
<i>Figura 17 Diagrama de Secuencia del CU Gestionar SED (sección Eliminar elemento).</i> .....	39
<i>Figura 18 Modelo de Despliegue.</i> .....	40
<i>Figura 19 Diagrama de Componentes.</i> .....	41
<i>Figura 20 Editor de Ecuaciones, versión 2.0, en el Portal de Servicios Bioinformáticos.</i> .....	43
<i>Figura 21 Editor de Ecuaciones, versión 2.0.</i> .....	43
<i>Figura 22 Editor de Ecuaciones, versión 2.0, importando un fichero.</i> .....	44
<i>Figura 23 Editor de Ecuaciones, versión 2.0, insertando caracteres griegos.</i> .....	44
<i>Figura 24 Editor de Ecuaciones, versión 2.0, cambiando la fuente de las ecuaciones.</i> .....	45
<i>Figura 25 Código del método “selected”.</i> .....	48
<i>Figura 26 Gráfico de control del método “selected”.</i> .....	49
<i>Figura 27 Código del método “pintar”.</i> .....	50
<i>Figura 28 Gráfico de control del método “pintar”.</i> .....	50
<i>Figura 29 Diagrama de Secuencia del CU Exportar SED.</i> .....	63
<i>Figura 30 Diagrama de Secuencia del CU Importar ficheros.</i> .....	63
<i>Figura 31 Diagrama de Secuencia del CU Modificar fuente.</i> .....	63
<i>Figura 32 Clases de ecuaciones del método experimental.</i> .....	67

## ÍNDICE DE TABLAS

<i>Tabla 1 Actor del Sistema.....</i>	<i>22</i>
<i>Tabla 2 Descripción del CU Gestionar SED. ....</i>	<i>23</i>
<i>Tabla 3 Descripción del CU Exportar SED. ....</i>	<i>26</i>
<i>Tabla 4 Descripción del CU Importar ficheros. ....</i>	<i>27</i>
<i>Tabla 5 Descripción del CU Modificar fuente. ....</i>	<i>28</i>
<i>Tabla 6 Plan de Pruebas.....</i>	<i>47</i>
<i>Tabla 7 Variables utilizadas en el CU Gestionar SED.....</i>	<i>51</i>
<i>Tabla 8 Caso de Prueba 1 Adicionar nueva ecuación.....</i>	<i>52</i>
<i>Tabla 9 Caso de Prueba 2 Insertar identificador o constante numérica. ....</i>	<i>52</i>
<i>Tabla 10 Caso de Prueba 3 Insertar operación matemática. ....</i>	<i>53</i>
<i>Tabla 11 Caso de Prueba 4 Eliminar elemento.....</i>	<i>53</i>
<i>Tabla 12 No conformidades. ....</i>	<i>54</i>
<i>Tabla 13 Tiempo de respuesta.....</i>	<i>55</i>
<i>Tabla 14 Variable utilizada en el CU Importar ficheros.....</i>	<i>64</i>
<i>Tabla 15 Caso de Prueba 5 Importar ficheros. ....</i>	<i>64</i>
<i>Tabla 16 Variable utilizada en el CU Exportar SED.....</i>	<i>64</i>
<i>Tabla 17 Caso de Prueba 6 Exportar fichero. ....</i>	<i>64</i>
<i>Tabla 18 Caso de Prueba 7 Font.....</i>	<i>65</i>
<i>Tabla 19 Caso de Prueba 8 Size.....</i>	<i>65</i>
<i>Tabla 20 Caso de Prueba 9 Foreground. ....</i>	<i>66</i>
<i>Tabla 21 Caso de Prueba 10 Background.....</i>	<i>66</i>

### INTRODUCCIÓN

La regla para conocer el funcionamiento de una célula ha sido purificar, aislar y analizar los componentes y estructuras en condiciones experimentales simples y bien controladas. Sin embargo, a raíz de los modernos métodos analíticos de alto rendimiento, se ha generado un modo distinto de analizar las funciones celulares que no depende de aislar los componentes o las funciones sino por el contrario analizar conjuntos complejos de componentes y funciones. Esta modalidad para estudiar la célula como un todo, es decir tomando un punto de vista más holístico que el tradicional, es lo que hoy se denomina Biología de Sistemas (*Systems Biology*) y está en plena evolución. (1)

La Biología de Sistemas es denominada una disciplina, se encuentra vinculada a diferentes áreas como la Biología, la Física, la Informática y la Matemática. Su objetivo radica en integrar la información de estas áreas de forma que se consiga un mayor entendimiento de las interacciones entre los componentes de los sistemas vivos, y por consiguiente, de sus procesos biológicos. Actualmente es muy utilizada por numerosos expertos de todo el mundo. Su unidad fundamental es un modelo computacional (*o in-silico*) que es construido a partir de componentes celulares y sus mecanismos de comunicación. Brinda una oportunidad única de entrelazar moléculas y sus mensajes dentro de una herramienta computacional que puede ser probada contra un amplio conjunto de condiciones y combinaciones. (2) (3)

La Biología de Sistemas trata de entender los sistemas biológicos a diferentes niveles de abstracción, desde el nivel molecular hasta los ecosistemas y haciendo uso de diferentes tipos de modelos matemáticos y técnicas computacionales, que van desde los Sistemas de Ecuaciones Diferenciales (SED) hasta la Minería de Datos. (4)

El modelo de un sistema biológico puede ser representado mediante sistemas de ecuaciones. La solución de las ecuaciones describe cómo el sistema biológico se comporta tanto en el tiempo como en el equilibrio, ya sea por medios analíticos o numéricos. Existen disímiles tipos de ecuaciones y el tipo de comportamiento que puede ocurrir es dependiente de las condiciones iniciales y parámetros que estén representados en dicho modelo. A menudo el modelo hace suposiciones sobre los sistemas. (3)

El desarrollo de las tecnologías ha propiciado el estudio de modelos de sistemas biológicos más complejos, por lo que es mayor la complejidad de los sistemas de ecuaciones que son utilizados para representarlos y aumenta la cantidad de información que generan. Es ineludible

entonces, el empleo de herramientas computacionales que permitan analizar, modelar y simular los sistemas biológicos mediante sistemas de ecuaciones.

Software desarrollados por empresas como Entelos<sup>1</sup>, son ejemplos de aplicaciones que han surgido para facilitar el trabajo de los científicos en los centros de investigación; muchos de los existentes en el mundo no son distribuidos gratuitamente a la comunidad científica o no les son asequibles. Una dificultad encontrada en dichos software o herramientas, es que se crean en general, para darle solución a un problema biológico determinado. Otras aplicaciones de software desarrolladas de manera independiente que poseen la característica de ser distribuidos de forma gratuita, tienen como desventaja que solo almacenan la información relacionada con los modelos utilizados y descartan los resultados arrojados por estos. (3)

El departamento de Bioinformática de la Universidad de las Ciencias Informáticas (UCI) en conjunto con el Centro de Inmunología Molecular (CIM), se dispuso de la creación de la Plataforma para la Simulación y Análisis de Sistemas Biológicos (BioSyS), para dotar a los investigadores de estos centros científicos de una herramienta computacional capaz de realizar un estudio y análisis de los modelos de sistemas biológicos, a través de una interfaz amigable destinada a toda clase de usuario. Los modelos en BioSyS, son representados mediante sistemas de ecuaciones diferenciales. Se encuentra integrada actualmente a la Plataforma de Servicios Bioinformáticos, con el propósito de utilizar sus funcionalidades desde cualquier lugar a través de internet. A la Plataforma de Servicios Bioinformáticos se puede acceder por medio de un Portal Web implementado en forma de portlets, dentro del Portal de gestión de contenidos Liferay.

BioSyS, cuenta con un Editor de Ecuaciones creado por parte del equipo de desarrollo del departamento de Bioinformática, con el objetivo de facilitar al investigador, el proceso de edición de modelos matemáticos para su posterior simulación.

El Editor de BioSyS, en su versión 1.2, procesa las ecuaciones editadas o importadas en forma de paneles, lo que significa que cada elemento de la ecuación o sistema de ecuaciones, entiéndase como variables, números, operadores lógicos u operadores aritméticos, se identifica como un panel por debajo de la interfaz. Esto a simple vista el usuario que está interactuando con la aplicación no lo puede ver. El Editor tiene que reconocer la ecuación panel por panel (ver figura 1). De ahí es donde proviene la demora del Editor con el tiempo de respuesta cuando se

---

<sup>1</sup> Entelos es una empresa de modelado en silicio, con modelos propios mecanicistas matemáticos de la fisiología humana y de los animales que predicen la progresión de la enfermedad y los efectos de la intervención terapéutica.

editan modelos matemáticos. Se hace necesario potenciar el Editor, para que los usuarios que interactúen con la herramienta, editen los sistemas de ecuaciones que deseen estudiar, de una manera más rápida a través de una interfaz más sencilla.

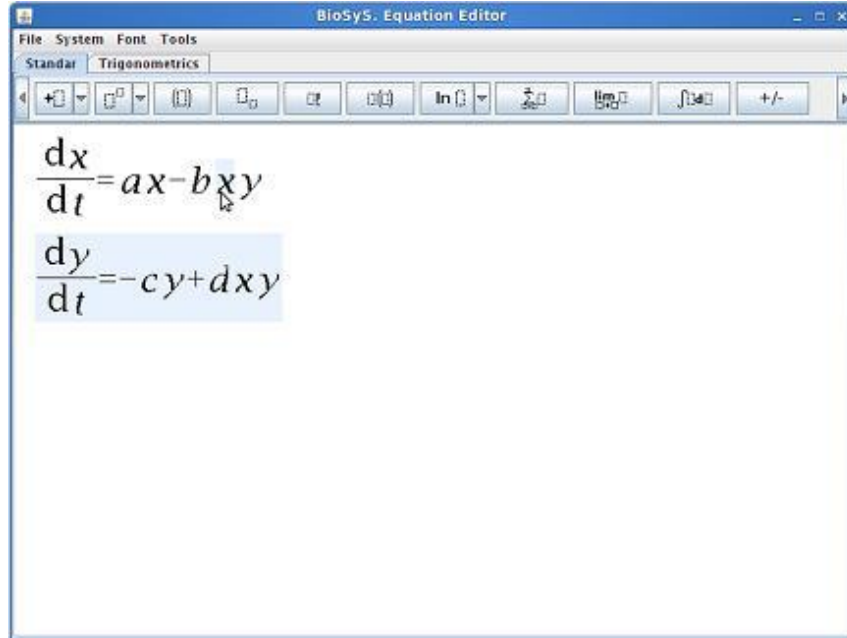


Figura 1 Editor de Ecuaciones de BioSyS, versión 1.2.

Debido a la importancia de esta necesidad surge como **problema a resolver**: ¿Cómo disminuir el tiempo de respuesta del Editor de Ecuaciones de BioSyS al editar modelos matemáticos?

Se plantea como **Objeto de Estudio**: Editores de Ecuaciones.

Enmarcado en el **Campo de acción**: Editor de Ecuaciones en el software BioSyS.

Se define como **Objetivo general**: Desarrollar la versión 2.0 del Editor de Ecuaciones de BioSyS que permita disminuir el tiempo de respuesta durante la edición de modelos matemáticos.

Desglosado en los **Objetivos específicos**:

- 1) Diseñar un nuevo Editor de Ecuaciones que permita editar modelos matemáticos de una forma más rápida y amigable.
- 2) Implementar el Editor de Ecuaciones versión 2.0 mediante el nuevo diseño propuesto.
- 3) Validar el Editor de Ecuaciones integrado en la Plataforma de Servicios Bioinformáticos.

Para guiar la investigación se establecen las siguientes **Tareas**:

- 1) Realización de una revisión bibliográfica sobre la edición de ecuaciones.
- 2) Estudio del diseño arquitectónico del Editor de Ecuaciones del software BioSyS.
- 3) Elaboración de un nuevo diseño arquitectónico del Editor de Ecuaciones.
- 4) Implementación del Editor de Ecuaciones con el nuevo diseño propuesto.
- 5) Integración del nuevo Editor de Ecuaciones al software BioSyS.
- 6) Implementación del portlet del módulo Editor de Ecuaciones del software BioSyS.
- 7) Realización de pruebas caja blanca y caja negra al Editor de Ecuaciones desarrollado.

La **estructura del documento** queda comprendida en: Resumen, Introducción, cuatro Capítulos, Conclusiones Generales, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos y Glosario.

**CAPITULO 1. Fundamentación Teórica.** Se exhibe el marco teórico estudiado y seleccionado para el presente trabajo, orientado al Editor de Ecuaciones del software BioSyS que se desea integrar a la Plataforma de Servicios Bioinformáticos. Se presenta un panorama general de los conceptos y herramientas imprescindibles para lograr los objetivos trazados.

**CAPITULO 2. Características del Sistema.** Se establece una breve descripción de la solución planteada, los requisitos funcionales y no funcionales, así como el modelo conceptual del sistema. Se describen los diagramas casos de uso del sistema y se definen los actores del sistema.

**CAPITULO 3. Diseño de la Aplicación.** Se describe la representación arquitectónica del sistema. Se detallan los principios de diseño que se tuvieron en cuenta durante el desarrollo de la aplicación. Se hace énfasis en el patrón de arquitectura utilizado, así como los patrones de diseño que fueron empleados. Se realiza el diagrama de clases del diseño dando respuesta a la solución que se propone.

**CAPITULO 4. Implementación y Prueba.** Se describe la implementación del sistema y se muestra el diagrama de componentes. Por lo demás, se muestran los casos de prueba de caja negra para cada caso de uso con el objetivo de validar el correcto funcionamiento del Editor de Ecuaciones integrado a la Plataforma de Servicios Bioinformáticos y se muestran los casos de prueba de caja blanca, con los resultados esperados.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### Introducción

El tiempo de respuesta de un software es cuánto tiempo tarda un sistema en empezar a devolver resultados al usuario. Este es el parámetro más importante que debe tener en cuenta el o los programadores. De él o ellos depende que se pueda ajustar el diseño para que los usuarios puedan empezar a ver visualizada la respuesta de su petición lo más rápido posible. En el vigente capítulo, se realiza una búsqueda y análisis de información para explicar las herramientas y metodologías utilizadas que dan respuesta al problema planteado anteriormente. Además se realiza una breve descripción de conceptos que serán aplicados por todo el ciclo de desarrollo de la aplicación.

#### 1.1 Editores de Ecuaciones

Los Editores de Ecuaciones son software empleados para elaborar fórmulas o trabajos matemáticos en formatos entendibles por las computadoras, que se realizan seleccionando símbolos de una barra de herramientas y escribiendo variables y números. Tienen la propiedad de ajustar automáticamente los tamaños de la fuente, el espacio y el formato, con el propósito de sustentar las convenciones matemáticas y tipografías para que las ecuaciones se perciban de una forma más correcta y elegante. Pueden definir los estilos automáticos y ajustar el formato mientras se va trabajando. Ciertos editores necesitan del aprendizaje de un lenguaje propio para introducir las fórmulas, entretanto otros son WYSIWYG<sup>2</sup>. Los editores WYSIWYG generalmente poseen una barra de herramientas con botones para que el usuario pueda añadir los símbolos a las fórmulas. (3)

Seguidamente se presentan software que permiten a los científicos editar ecuaciones:

#### Equation Editor

Equation Editor es un editor que fue desarrollado por Design Science y permite a los usuarios construir ecuaciones matemáticas mediante un ambiente WYSIWYG. Se encuentra incluido en la mayoría de los editores de texto (ej. MS Word), programas de presentación (ej. MS PowerPoint) y muchos otros productos. (3)

Es una versión simplificada de MathType, la misma se referencia a continuación.

---

<sup>2</sup> WYSIWYG, acrónimo de What You See Is What You Get. Aplicado a editores de texto con formato que permiten escribir un documento viendo directamente el resultado final.



# CAPÍTULO 1. Fundamentación Teórica.

---

## MathType

MathType es la versión profesional de Equation Editor, desarrollado también por Design Science y está acoplado a varias herramientas de edición de texto en las plataformas Windows y Macintosh. Posee una variedad de prestaciones adicionales, entre ellas tiene la opción "*Export to MathPage*" (Exportar a MathPage), que genera un documento XML en el cual las expresiones están incluidas en formato MathML. Posee características que lo diferencian de los demás editores, como:

- La posibilidad de inserción de referencias a secciones y ecuaciones.
- Presenta una interfaz gráfica sencilla de usar.
- Acceso a cientos de símbolos y plantillas.
- Ecuaciones reeditables tras su inserción.
- Exportación de expresiones a código MathML.
- Posibilidad de resaltar elementos de una misma ecuación en distintos colores. (3)

A pesar de tener las características mencionadas tiene como desventaja que no es multiplataforma, solo es funcional en Windows y Macintosh, es un software propietario y no permite la realización de modificaciones que posibilitan interacciones con herramientas de análisis matemático.

## OpenOffice Math

OpenOffice Math proporciona una interfaz simple para crear y editar fórmulas matemáticas. Contiene un panel de selección que contiene operadores, símbolos científicos, expresiones, etc., siendo suficiente seleccionar cualquiera de ellos mediante un clic de ratón para incorporarlo al documento. Las fórmulas creadas se pueden adicionar dentro de otros documentos. Soporta múltiples fuentes y exporta a diferentes formatos, por ejemplo MathML. Esta herramienta no permite su integración con herramientas de análisis matemático. No permite un trabajo amplio con las ecuaciones que se editan, manteniéndolas en un formato rígido. (3)

Los editores nombrados anteriormente no cuentan con mecanismos para comprobar si la estructura sintáctica de las ecuaciones está correcta. Otro problema que tienen, es que no permiten realizar ningún tipo de análisis sobre las ecuaciones creadas, ni tampoco logran integrarse a herramientas que sí lo hagan.

# CAPÍTULO 1. Fundamentación Teórica.

---

En la actualidad determinados editores posibilitan la realización de cálculos numéricos y análisis matemático, tal es el caso del Editor de Ecuaciones de BioSyS.

## 1.2 BioSyS

BioSyS es un potente software que tiene como objetivo principal realizar exploraciones intensivas sobre modelos de sistemas biológicos propiciando la aplicación de técnicas de meta análisis sobre los resultados obtenidos. Integra algoritmos y herramientas imprescindibles para servir de cimiento en las investigaciones a aquellos científicos que se dedican al estudio de la Biología de Sistemas. Se encuentra integrado en la Plataforma de Servicios Bioinformáticos en la actualidad, para que sea accesible al usuario a través de internet, ya que antes era una aplicación *desktop* y no era accesible para todo el personal destinado a la misma. Cuenta con los módulos que se describen a continuación:

### Base de Datos

El software BioSyS posee una base de datos que permite almacenar los modelos. También la información relacionada con los resultados alcanzados en las exploraciones intensivas sobre estos, así como los parámetros y condiciones iniciales de dichos modelos.

### Módulo de Simulación

Las funciones que proporciona este módulo se emplean en las exploraciones. Se ejecutan desde simulaciones sencillas hasta grupos de simulaciones que pueden realizarse de forma local o distribuida utilizando la Plataforma de Cálculo distribuidos T-Arenal, quedando todo almacenado en la base de datos.

### Módulo de Análisis

Este módulo permite realizar meta análisis sobre los resultados de las simulaciones. El mismo está constituido por los siguientes métodos de análisis:

- ✓ Dinámica de poblaciones.
- ✓ Análisis de clasificación.
- ✓ Análisis de clúster.
- ✓ Análisis de bifurcaciones.
- ✓ Análisis de bifurcaciones 1D/2D.
- ✓ Análisis de estabilidad.
- ✓ Análisis por reglas.

# CAPÍTULO 1. Fundamentación Teórica.

---

## Editor de Ecuaciones

Permite que los usuarios interactúen con la herramienta, se editan los sistemas de ecuaciones que describen los modelos que desean estudiar a través de una interfaz amigable. Garantiza además, que la estructura sintáctica de las ecuaciones sea correcta. El editor implementa una serie de funcionalidades adicionales que facilitan el trabajo con los modelos matemáticos, la reutilización de funciones y demás. (3)

### 1.3 Necesidad de la creación del Editor de Ecuaciones de BioSyS, versión 2.0

El Editor de BioSyS, actualmente presenta problemas en el tiempo de respuesta durante el proceso de edición de modelos de sistemas biológicos. Se hace necesario mejorar el tiempo de respuesta del Editor de Ecuaciones de BioSyS, así como la creación de una interfaz más amigable con el usuario, manteniendo su flexibilidad y que sea fácil de utilizar, con el propósito de poder integrar este nuevo software a la Plataforma de Servicios Bioinformáticos y a BioSyS.

Se utilizó el término amigable, refiriéndose a la estructuración de la interfaz, para describir un entorno visual sencillo que permita la comunicación fluida entre el usuario y la aplicación. Ahora los menús estarán más organizados en cuanto a funciones a realizar. El usuario, por ejemplo, podrá cambiar la fuente de las ecuaciones de forma ágil, sin tener que acceder a ningún menú. El funcionamiento del Editor será más rápido porque cambiará la forma de procesar las ecuaciones o sistemas de ecuaciones, editados o exportados. Como en versiones anteriores el usuario no se dará cuenta de este proceso, aunque si notará la mejora en el tiempo de respuesta.

### 1.4 Plataforma de Servicios Bioinformáticos

La Plataforma tiene como objetivo, posibilitar el acceso de características avanzadas, como son los flujos de trabajo y el uso de ontologías para la composición y ejecución de tareas, a especialistas de Biología, de manera que los mismos no tengan que manejar conceptos engorrosos. Integra un conjunto de herramientas de uso común en la Bioinformática, así como los productos desarrollados por el departamento con el propósito de brindar servicios que puedan ser consumidos por aplicaciones o usuarios a través de un Portal. Posee una arquitectura orientada a servicios y de cara al usuario con interfaces web, implementadas en forma de portlets, dentro del Portal de gestión de contenidos Liferay.

# CAPÍTULO 1. Fundamentación Teórica.

---

El Portal Web de Bioinformática es una aplicación que incorpora las últimas tendencias tecnológicas para el diseño y ejecución de experimentos *“in silicio”* mediante flujos de trabajo, ocultando a los usuarios los detalles técnicos del proceso.

A modo general, el trabajo de la plataforma se manifiesta cuando el usuario realiza una petición al servidor mediante el protocolo HTTP a través del navegador. Dicha petición es gestionada y enviada al contenedor de portlets, el cual en función del servicio solicitado, muestra los portlets correspondientes. Cuando el servicio a ejecutarse dentro del portlet requiere gran capacidad de cómputo se utiliza la plataforma de cálculos distribuidos T-Arenal para realizar los cálculos de manera eficiente. Al finalizar cada uno de los componentes se muestran al usuario cumpliendo las especificaciones estándares de portlets JSR 168 y 286; y la información referente a los perfiles de usuarios y configuración de los mismos es gestionada en una base de datos de PostgreSQL garantizando así la persistencia de los datos. (5)

## 1.5 Applet Java

Los applets constituyen la forma de incorporar programas complejos en el contorno de una página web. Se trata de pequeños programas, que se transfieren con las páginas web y ejecutan en el contexto de un navegador ya que son un elemento más de una página web, como una imagen o una porción de texto. Son programados en Java, por lo que se favorecen de la potencia de este lenguaje para la Red. (6)

El lenguaje Java logra producir dos tipos de programas: los applets y las aplicaciones. Al cargar el navegador de la página web, el applet incrustado en la misma se carga y ejecuta. Mientras que un applet puede transmitirse por Internet una aplicación reside en el disco duro local. La aplicación Java se instala en la máquina como cualquier otra. Una discrepancia es que applet no está autorizado a acceder a archivos o directorios del ordenador a menos que sea un applet totalmente fiable. (6)

Entre las características de los applets se vale destacar que son multiplataforma y son compatibles con la mayoría de los navegadores. Además pueden ejecutarse a velocidades semejantes a las de otros lenguajes compilados como por ejemplo C++. Se consiguen almacenar en la memoria caché de la totalidad de los navegadores web, de manera que se cargan rápido, al cargarse nuevamente la página web. Otra característica muy notable es que logran trasladar el trabajo del servidor al cliente, haciendo una solución web más escalable tomando en cuenta el número de los usuarios o clientes.

# CAPÍTULO 1. Fundamentación Teórica.

En el presente trabajo de diploma se incorpora el término de applet para poder llevar la versión *desktop* del Editor de Ecuaciones a la web y seguidamente se podrá realizar el portlet que se integrará a la plataforma. Es posible que en algunos casos se necesite que el applet importe o exporte en el disco local algún tipo archivo. Por las restricciones de seguridad de la Máquina Virtual de Java, esto no es posible, ya que para evitar ataques maliciosos un applet se ejecuta en un entorno controlado. Para que el applet que contiene el editor fuera confiable y pudiera acceder a las funciones de la aplicación totalmente, se firmó a través de un certificado (un fichero que contiene una firma digital). De esta forma el cliente puede estar tranquilo sabiendo que la ejecución del applet no va a tener ningún efecto malicioso en su equipo.

## 1.6 Portlet

Un portlet es una aplicación web que permite la creación de portales a través de una serie de convenciones. Es competente para gestionar su ciclo de vida y su integración con otros portlets. Es un componente modular de las interfaces de usuario gestionadas y visualizadas en un portal web (ver figura 2). Produce fragmentos de código de marcado que se agregan en una página de un portal.

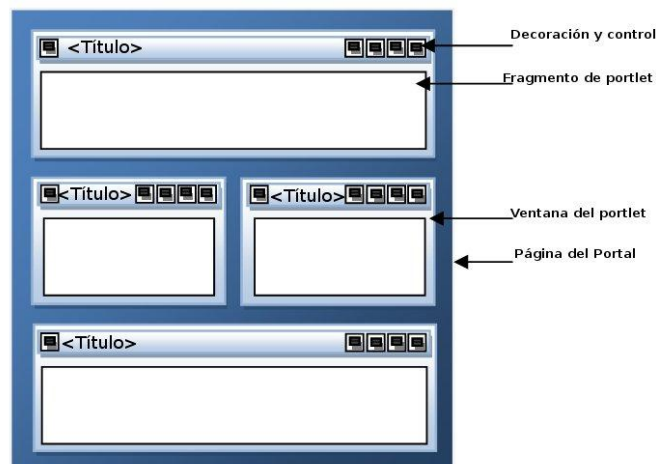


Figura 2 Estructura de un portlet.

Los portales Java facultan la escritura de nuestro propio código para extender el portal, a través de componentes estandarizados llamados Portlets. Los Portlets Java, son definidos en JSR 186 y JSR 286 que son las especificaciones de los mismos. Surgen por la necesidad de poder desplegar cualquier portlet desarrollado en cualquier sitio web. Además de especificar el ciclo de vida del portlet, las características, cómo deben lucir y la manera estándar en el que se deben desarrollar. (7) (8)

# CAPÍTULO 1. Fundamentación Teórica.

---

## 1.6.1 Liferay Portal 6.0.5

Es un portal de gestión de contenidos de código abierto escrito en Java. Se creó en el 2000. Actualmente es el portal *open source* líder, de la comunidad Java de orientación Web 2.0<sup>3</sup>, pensando en el usuario final para su desenvolvura a la hora de usarse. Liferay Portal constituye una plataforma web que asiste al desarrollo de soluciones empresariales con resultados inmediatos y valor a largo plazo.

La plataforma Liferay Portal, respecto a otras, se diferencia por su equilibrio inapreciable entre funcionalidad práctica, usabilidad e innovación técnica. El producto proporciona una gestión de contenidos, un entorno de colaboración y una plataforma para redes sociales fáciles de utilizar, gracias a su retribuida interfaz de usuario. Se puede reducir el coste de los portales basados en Liferay a través del acogimiento de estándares en las tecnologías utilizadas para el progreso de aplicaciones (portlets), almacenamiento y tratamiento de contenidos, servicios web y capa de presentación. (7)

Ayuda a desarrollar soluciones sólidas que brindan resultados apetecidos en varios ámbitos como portales de autoservicio, integración con aplicaciones corporativas, sitios Web 2.0 dinámicos y espacios de trabajo para la reciprocidad de conocimiento.

La versión utilizada de Liferay Portal es la 6.0.5, según está definida en el documento de Proyecto Técnico DATEC\_PSBio\_0208\_PryTec.doc perteneciente a la Plataforma de Servicios Bioinformáticos.

## 1.6.2 Spring

Spring es un *framework* de aplicación desarrollado por la compañía Interface 21, para aplicaciones escritas en el lenguaje de programación Java. Además se considera a Spring un *framework lightweight*, es decir liviano o ligero, ya que no es una aplicación que requiera de muchos recursos para su ejecución, además el *framework* completo puede ser distribuido en un archivo .jar de alrededor de 1 MB, lo cual representa muy poco espacio, y para la cantidad de servicios que ofrece es relativamente insignificante su tamaño.

Una de las características que ayuda a este éxito, es que es una aplicación *open source*, lo cual implica que no tiene ningún costo, ni se necesita una licencia para utilizarlo, por lo tanto da la libertad a muchas empresas y desarrolladores a incursionar en la utilización de esta aplicación.

---

<sup>3</sup> Web 2.0 comprende aquellos sitios web que proporcionan información compartida y servicios especiales, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web.

# CAPÍTULO 1. Fundamentación Teórica.

---

Además de que está disponible todo el código fuente de este *framework* en el paquete de instalación. (9)

Según está definido en el documento de Proyecto Técnico DATEC\_PSBio\_0208\_PryTec.doc perteneciente a la Plataforma de Servicios Bioinformáticos, se utilizará Spring como *framework* de desarrollo para la realización de portlet.

Teniendo en cuenta lo que se desea obtener y basándose en la arquitectura definida para BioSyS y la Plataforma de Servicios Bioinformáticos es que se hace un estudio de las herramientas y metodología a utilizar para la creación de la versión 2.0 del Editor de Ecuaciones.

## 1.7 Metodología y herramientas utilizadas

### 1.7.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software constituyen un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática específica que sirve como guía para afrontar y resolver nuevos problemas de índole similar, así como estructurar, planear y controlar el proceso de desarrollo en sistemas de información. (10)

#### OpenUp

Es un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

OpenUp solamente incluye el contenido necesario y fundamental, ya que es un proceso mínimo y suficiente. Por lo tanto, no provee lineamientos para todos los elementos que se manejan en un proyecto, pero tiene los componentes primordiales que pueden servir de base a procesos específicos. Para fomentar el intercambio de información entre los equipos de desarrollo y conservar un entendimiento compartido del proyecto, sus objetivos, alcance y avances es que se declaran la totalidad de sus elementos. (11)

#### Características

- Es conveniente para proyectos pequeños y de bajos recursos.
- Permite la disminución de probabilidades de fracaso e incrementa el éxito.
- Detecta errores tempranos mediante un ciclo iterativo.

# CAPÍTULO 1. Fundamentación Teórica.

---

- Impide la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- Tiene un enfoque al cliente y posee pocas iteraciones ya que es una metodología ágil.

Por lo escrito anteriormente, se determinó que la metodología que guiará el desarrollo de la aplicación es OpenUp debido a sus características, además de ser la metodología seleccionada y utilizada en el desarrollo de versiones anteriores del software de BioSyS.

## 1.7.2 Lenguaje de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Su composición está determinada por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. (12)

### Java

Java es un lenguaje compilado e interpretado, por lo que todo programa en Java se compila y el código que se concibe *bytecodes* es interpretado por una máquina virtual dependiente a la plataforma, logrando así, la independencia de la máquina. También es un lenguaje orientado a objetos de propósito general. (13)

Se encuentra diseñado con varios niveles de seguridad que datan desde el contorno del programador hasta el contorno de la ejecución de la máquina virtual. Consta de mecanismos que supervisan la seguridad durante la ejecución, cuidando que no se viole ninguna restricción de seguridad del sistema donde se ejecutará, antes de ejecutar el código. También goza de un gestor de seguridad para restringir el acceso a los recursos del sistema. No posee punteros, realiza el acceso a las instancias de las clases a través de referencias. Sin la necesidad de utilizar ninguna biblioteca puede acometer programación concurrente. (13)

Para favorecer al programador, Java ejecuta comprobación estricta de tipos durante la compilación, impidiendo problemas como desbordamiento de la pila. Conjuntamente, el programador se encuentra perennemente obligado a tratar las posibles excepciones que se produzcan en tiempo de ejecución. Define procedimientos para el tratamiento de errores.



# CAPÍTULO 1. Fundamentación Teórica.

---

Java ha sido el lenguaje en donde se han desarrollado las versiones anteriores del Editor de Ecuaciones. En esta versión 2.0 del Editor también se empleará, ya que sería muy engorrosa la realización de este módulo en otro lenguaje.

## 1.7.3 Herramientas de desarrollo de software

### NetBeans 7.2.1

NetBeans es un Entorno de Desarrollo Integrado (del inglés Integrated Development Environment, IDE). Se caracteriza por ser un proyecto de código abierto de gran éxito con una enorme base de usuarios. También es una plataforma para el desarrollo de aplicaciones con lenguaje de programación Java. Brinda servicios comunes a las aplicaciones de escritorios, con el objetivo de proporcionar al o los desarrolladores el enfoque a la lógica específica de su aplicación. Permite que las aplicaciones se desarrollen a partir de un conjunto de componentes de software llamados módulos. A modo general es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. (14)

Un módulo es un archivo Java, que posee clases de Java escritas para interactuar con las Interfaces de Programación de Aplicaciones (del inglés Application Programming Interface, APIs) del NetBeans. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (14)

Para la implementación de la aplicación fue necesario escoger NetBeans como herramienta de desarrollo a utilizar según el lenguaje de programación que se seleccionó. También ha sido utilizado en versiones anteriores del Editor.

### Eclipse 3.6

Eclipse es un IDE de código abierto multiplataforma y extensible para la integración de herramientas. Dispone de módulos (*plug-in*) para proveer toda su funcionalidad, al contrario de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Dicho mecanismo de módulo es una plataforma ligera para componentes de software. Además se logra que Eclipse se extienda usando otros lenguajes de programación como lo son C, C++ y Python. Permite el trabajo con lenguajes para procesado de texto, aplicaciones en red como Telnet y Sistema de Gestión de Base de Datos.

# CAPÍTULO 1. Fundamentación Teórica.

---

Eclipse provee al programador con *frameworks* muy ricos tanto para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, como aplicaciones web. Los clientes que lo utilizan para desarrollar *plug-ins* disponen de tecnología probada, fiable y escalable para diseñar, desarrollar e implementar productos comerciales. (15)

Se utilizará el Eclipse 3.6 para la creación del portlet, según está definida en el documento de Proyecto Técnico DATEC\_PSBio\_0208\_PryTec.doc perteneciente a la Plataforma de Servicios Bioinformáticos.

## 1.7.4 Herramienta CASE

Las Herramientas CASE (Computer Aided Software Engineering, en español Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones orientadas al crecimiento productivo en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. Están definidas también como el conjunto de métodos, utilidades y técnicas que proporcionan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. (16)

### Visual Paradigm para UML

Visual Paradigm para UML, es una herramienta CASE que brinda un conjunto de ayudas para el desarrollo de programas informáticos, partiendo de la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido diseñada para una extensa gama de usuarios interesados en la construcción de sistemas de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Soporta el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. (17)

### Características

- Es multiplataforma.
- Centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Capacidad de ingeniería directa e inversa.
- Licencia: gratuita y comercial.
- Genera código para Java y exporta como HTML.
- Compatible entre ediciones.

# CAPÍTULO 1. Fundamentación Teórica.

---

Se decidió utilizar el Visual Paradigm para UML 8.0 para visualizar y diseñar los elementos de software debido a que es multiplataforma y por las facilidades que brinda para el diseño de los diagramas necesarios y su documentación.

## 1.7.5 Lenguaje de modelado

La consistencia de un lenguaje de modelado no es más que la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica cuales son los pasos que se deben seguir para arribar a un diseño. Para la estandarización de un lenguaje de modelado se requiere de agentes involucrados en un proyecto informático porque es la parte principal del proceso de comunicación. (18)

## UML 2.0

UML (Unified Modeling Language, también conocido como LUM por sus siglas en español Lenguaje Unificado de Modelado), es el lenguaje de modelado de sistemas de software más conocida y utilizado en la actualidad. Se encuentra respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Cuenta con diversos tipos de diagramas que muestran aspectos diferentes de las entidades representadas. Ofrece un estándar para describir un "plano" del sistema (modelo). Incluye aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (18)

Por tanto, se define UML 2.0 para la realización del Editor porque es un lenguaje de modelado para especificar o describir métodos o procesos. Se emplea para la definición de sistemas, la caracterización de los artefactos de los sistemas, así como la documentación y construcción de los mismos. No especifica qué metodología o proceso usar, pero se aplica en el desarrollo del software diversidades de formas para dar soporte a la metodología de desarrollo de software. (19)

## 1.7.6 Lenguaje de marcado

Se entiende por lenguaje de marcado a la forma de codificar un documento que, en conjunto con el texto, integra etiquetas o marcas que contienen información adicional sobre la estructura del texto o su presentación. Es un mecanismo para identificar las estructuras en un documento. (20)

# CAPÍTULO 1. Fundamentación Teórica.

---

## XML

XML es un lenguaje de marcado para documentos que contienen información estructurada. Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Propone un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. (21)

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades enormes y básicas para la sociedad de la información. XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Además, permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello. (22)

Por todas las características reveladas anteriormente, XML se define como lenguaje de marcado, además permite una mejor organización en el trabajo con portlets.

### 1.8 Conclusiones del capítulo

En el presente capítulo se efectuó una definición de los conceptos más relevantes para la comprensión de la investigación con el fin de desarrollar el Editor de Ecuaciones de BioSyS, versión 2.0. Después de haberse realizado una conceptualización de las herramientas, software y tecnologías, queda determinado que la metodología de desarrollo con que se va a trabajar es OpenUp, como lenguaje de programación Java, lenguaje de marcado XML y lenguaje de modelado UML 2.0. Se utilizará el Visual Paradigm para UML 8.0 como herramienta CASE. Se implementará el nuevo diseño del Editor en la herramienta de desarrollo NetBeans 7.2.1. Se desarrollará el portlet dentro del Portal de gestión de contenidos Liferay 6.0.5, con la herramienta Eclipse 3.6 y Spring como *framework* de desarrollo, según queda definido en el Proyecto Técnico de la Plataforma de Servicios Bioinformáticos.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### Introducción

En el presente capítulo se ofrece una breve descripción de la solución propuesta en vista de solucionar el problema; así como el modelo conceptual y los requisitos funcionales y no funcionales. Se realiza una representación de los diagramas de casos de uso, su descripción y el actor que interviene en ellos.

### 2.1 Breve descripción del sistema

El Editor de Ecuaciones de BioSyS, en su versión 2.0 posibilitará la edición de modelos matemáticos de forma dinámica propiciando al usuario la gestión de sistemas de ecuaciones diferenciales. Brinda la opción de cambiar la fuente. Además facilitará importar los modelos matemáticos y exportar los sistemas de ecuaciones diferenciales a diferentes formatos.

### 2.2 Modelo Conceptual

A causa de la falta de estructuración de los procesos del negocio relacionados con el objeto de estudio y para una mayor comprensión del contexto en que se desarrolla el sistema, es que se describe el funcionamiento de la aplicación mediante una serie de conceptos, entidades y sus relaciones, agrupándolos en un modelo conceptual. Los modelos conceptuales son una representación del sistema por medio de definiciones organizadas en forma estructurada. Se utilizan para representar la realidad a un alto nivel de abstracción. Mediante los modelos conceptuales se puede construir una descripción de la realidad fácil de entender. (23)

#### 2.2.1 Representación del Modelo Conceptual

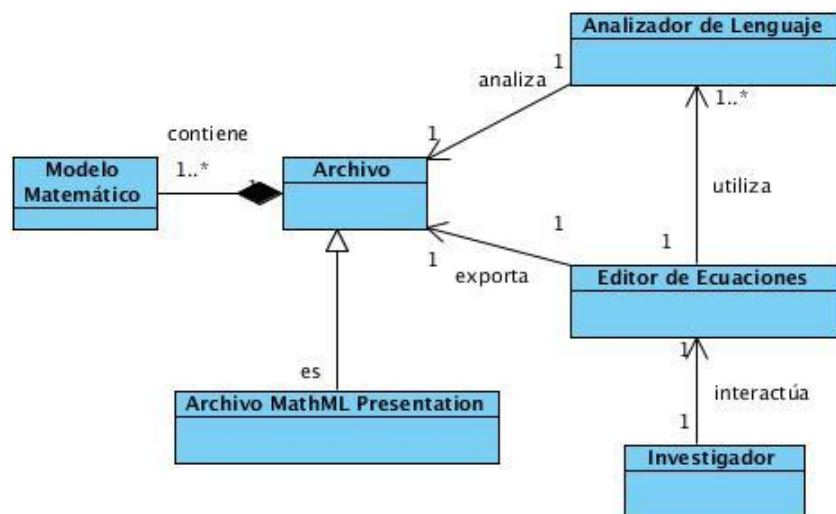


Figura 3 Modelo Conceptual.

## CAPÍTULO 2. Características del Sistema

---

### Descripción de las clases del Modelo Conceptual:

**Investigador:** Concepto que representa al papel de la persona encargada y capacitada para trabajar con la herramienta computacional.

**Modelo matemático:** Concepto que constituye un sistema de ecuaciones diferenciales ordinarias que describen la dinámica de desarrollo de los sistemas biológicos.

**Archivo:** Concepto que representa una unidad de datos o información almacenada en algún medio que es utilizada por el Editor de Ecuaciones para guardar la descripción de los modelos matemáticos en el formato MathML Presentation manejado por esta herramienta. Cada archivo se diferencia del resto debido a que tiene un nombre propio. En cuanto a su ubicación, se guarda en subdirectorios o directorios y para poder acceder a ellos, la ruta está conformada por los diversos subdirectorios que lo contienen, hasta llegar al directorio contenedor.

**Editor de Ecuaciones:** Concepto que representa a la aplicación integrada a BioSyS con el objetivo que los investigadores que interactúan con la herramienta, puedan editar los sistemas de ecuaciones ordinarias que describen los modelos matemáticos que desean estudiar, a través de una interfaz amigable.

**Analizador de Lenguaje:** Concepto que representa un procesador de lenguaje, comúnmente llamada traductor, para efectuar el proceso de análisis del contenido de un determinado tipo de archivo.

### 2.3 Especificación de los requisitos del sistema

Los requisitos son una descripción completa del comportamiento del sistema que se va a desarrollar, retribuyendo a un contrato, norma o determinada especificación. En otras palabras, son las condiciones o capacidades que necesita el usuario para resolver un problema u obtener una meta estipulada. Para la obtención de requisitos, tanto los funcionales como los no funcionales, se realizó un estudio de los requisitos definidos en versiones anteriores del Editor de Ecuaciones de BioSyS. Basándose en los documentos Especificación de Requisitos de Software de BioSyS y la tesis Software alasBioSyS v1.2: Módulo de Editor de Ecuaciones, quedaron definidos los que se muestran seguidamente.

#### 2.3.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de servicios que el sistema debe proporcionar, cómo debería reaccionar el sistema a determinadas entradas y cómo debería comportarse en

## CAPÍTULO 2. Características del Sistema

---

situaciones particulares. Definen el comportamiento interno del software y se complementan con los requisitos no funcionales. A continuación se muestran los requisitos funcionales del Editor de Ecuaciones, versión 2.0.

RF 1. Adicionar una nueva ecuación.

RF 2. Insertar literal o constante numérica.

RF 3. Insertar operación matemática.

RF 4. Eliminar elemento de la ecuación.

RF 5. Exportar a MathML File.

RF 6. Exportar a Java File.

RF 7. Exportar a Matlab File.

RF 8. Exportar a Mathematica File.

RF 9. Exportar como Imagen.

RF 10. Importar fichero en código MathML Presentation.

RF 11. Modificar la fuente del SED.

### 2.3.2 Requisitos No Funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones que brinda un sistema. Definen el comportamiento del sistema, no describen funciones a realizar o información a guardar. Contribuyen a la obtención de un producto usable, rápido, atractivo, confiable y son imprescindibles en su éxito.

#### ➤ **Apariencia**

- ✓ La aplicación deberá tener una interfaz externa amigable, sencilla y fácil de entender por el usuario para evitar que se pierda dentro de la aplicación.
- ✓ La distribución de la interfaz está en correspondencia con la actividad que el usuario vaya a realizar de manera coherente y lógica al realizar cada uno de los pasos dentro de la investigación que haga dentro de la aplicación.
- ✓ El cliente podrá ajustar la interfaz de trabajo a su gusto y facilidad.

#### ➤ **Usabilidad**

## CAPÍTULO 2. Características del Sistema

---

- ✓ El sistema podrá ser usado por aquellos usuarios que posean conocimientos básicos en el campo de la modelación de sistemas biológicos.
- ✓ Se debe lograr un diseño adaptable, con la capacidad de poder soportar funcionalidades adicionales o modificar las funcionalidades existentes sin impactar el resto de los requerimientos contemplados en el sistema.
- **Fiabilidad**
  - ✓ La reparación del sistema en caso de surgir fallas en el mismo debe realizarse en el menor tiempo posible, poniendo todos los esfuerzos en función de que no supere las 72 horas.
- **Software**
  - ✓ Multiplataforma.
  - ✓ Máquina virtual de Java 1.6 o versiones superiores.
- **Hardware**
  - ✓ Memoria RAM no menos de 512 MB.
  - ✓ Puertos USB para memoria externa.
  - ✓ Procesadores Pentium IV o superiores.
- **Soporte**
  - ✓ El sistema estará bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.
  - ✓ El sistema permite implementar cambios, ya sea cualquier corrección, mejora o adaptación del software.
  - ✓ La instalación del sistema debe caracterizarse por su facilidad, claridad y sencillez.
  - ✓ El sistema debe permitir la interacción con los demás módulos que componen la plataforma.
  - ✓ Se realizarán distintas pruebas al software una vez concluido para comprobar su funcionalidad.
  - ✓ Terminado el software se prestarán los servicios de instalación y configuración de la aplicación.
  - ✓ Se prestarán servicios de mantenimiento del software.
- **Requisitos de licencia**
  - ✓ El sistema se utilizará primeramente en el CIM y se piensa que posteriormente en los demás centros del Polo Científico, en el caso que luego se decida comercializar en el extranjero entonces deberá contar con una licencia.
- **Requisitos de diseño e implementación**



## CAPÍTULO 2. Características del Sistema

- ✓ Lenguaje de programación Java.
- ✓ NetBeans como IDE de desarrollo.
- ✓ Visual Paradigm para UML como herramienta CASE.
- **Requisitos legales, de derecho de autor y otros.**
  - ✓ Los derechos de autor serán registrados entre la UCI y el CIM.
- **Estándares aplicables**
  - ✓ Modelo ISO/IEC 9126 : Este es un modelo de calidad que centra atención en el modelo ISO/IEC 9126 adaptado para arquitecturas de software en el año 2003 por ser un modelo que se basa en los atributos de calidad que se relacionan directamente con la arquitectura: funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad.
  - ✓ MathML estándar de modelación. Es un estándar basado en XML, cuyo objetivo es expresar notación matemática de forma que distintas máquinas puedan entenderla. Aunque en sus inicios fue concebido para integrar las expresiones matemáticas en páginas web, ha evolucionado hasta convertirse en un formato para el intercambio y comunicación de las matemáticas en multitud de programas matemáticos y de otras ciencias.
  - ✓ Modelo 4+1 de Philippe Kruchten. Modelo para representación de la arquitectura del sistema a través de 4 vistas arquitectónicas.

### 2.4 Definición de Casos de Uso

Los casos de uso modelan comportamiento, interacción. Comprenden los pasos necesarios para alcanzar un objetivo de su actor principal. Deben proveer una especificación funcional completa, independiente de la tecnología. No describen el procesamiento interno del sistema, sólo la interacción y los resultados de valor para el usuario. (24)

#### 2.4.1 Actores del Sistema

Los actores del sistema son personajes o entidades que guardan relación con los casos de uso y que le demanda una funcionalidad.

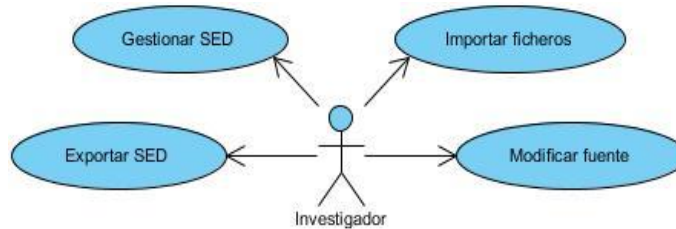
Tabla 1 Actor del Sistema.

Actor	Descripción
Investigador	Representa al usuario que va a hacer uso del sistema, con posibilidades de realizar interacción con todas sus funcionalidades.

## CAPÍTULO 2. Características del Sistema

### 2.4.2 Diagrama de Casos de Uso del Sistema

Los diagramas de casos de uso se utilizan para la ilustración de los requisitos del sistema al mostrar cómo reacciona a eventos que se producen en su contorno o en él mismo. Muestra la relación que existe entre los actores y los casos de uso en un sistema. Especifican la comunicación y el comportamiento de un sistema (ver figura 4).



**Figura 4 Diagrama de Casos de Uso del Sistema.**

### 2.4.3 Descripción de Casos de Uso del Sistema

**Tabla 2 Descripción del CU Gestionar SED.**

<b>Caso de Uso</b>	Gestionar SED.	
<b>Actor</b>	Investigador (inicia).	
<b>Propósito</b>	Crear y modificar el SED en estudio.	
<b>Resumen</b>	<p>El caso de uso se inicia cuando el investigador va a realizar alguna de las siguientes operaciones relacionadas con el trabajo con un SED:</p> <ul style="list-style-type: none"> <li>➤ Adicionar nueva ecuación.</li> <li>➤ Adicionar operación matemática.</li> <li>➤ Adicionar identificador o constante numérica.</li> <li>➤ Eliminar elemento de la ecuación.</li> </ul> <p>Por último, se finaliza el caso de uso cuando el sistema actualiza sus datos.</p>	
<b>Referencia</b>	RF1, RF2, RF3, RF4.	
<b>Precondición</b>	Debe estar insertada un área de trabajo.	
<b>Prioridad</b>	Crítico.	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Autor</b>	<b>Respuesta del Sistema</b>
	1. El caso de uso se inicia cuando el investigador decide realizar una de las	

## CAPÍTULO 2. Características del Sistema

<p>siguientes opciones:</p> <ul style="list-style-type: none"> <li>➤ Si desea Adicionar nueva ecuación, ir a sección “Adicionar nueva ecuación”.</li> <li>➤ Si desea Adicionar operación matemática ir a sección “Insertar operación matemática”.</li> <li>➤ Si desea Adicionar identificador o constante numérica ir a sección “Insertar identificador o Constante numérica”.</li> <li>➤ Si desea Eliminar un elemento ir a sección “Eliminar elemento”.</li> </ul>	
<b>Sección 1 “Adicionar nueva ecuación”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1. El investigador selecciona la opción de insertar nueva ecuación, para ello va al menú del editor Insert/Algebraic Equation.	1.2. El sistema adiciona en la región de trabajo una plantilla de ecuación. Esta está formada por una región editable seguida de un signo de igual y luego otra región editable. Finaliza el caso de uso.
<b>Flujos Alternos de la sección 1 “Adicionar nueva ecuación” del paso 1.1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1.1. El investigador selecciona la opción de insertar nueva ecuación diferencial, para ello va al menú del editor Insert/Differential Equation.	1.2.1. El sistema adiciona en la región de trabajo una plantilla de ecuación diferencial. Esta está formada:  En el miembro izquierdo una fracción que contiene tanto en el numerador como en el denominador una letra “d” seguida de una región editable.  En el miembro derecho una región editable.  Entre ambos miembros un signo de igual.  Finaliza el caso de uso.
<b>Sección 2 “Insertar identificador o constante numérica”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

## CAPÍTULO 2. Características del Sistema

2.1. El investigador selecciona el elemento donde desea insertar el texto.	2.2. El sistema pasa el foco de teclado al elemento seleccionado.
2.3. El investigador presiona una tecla asociada a un identificador o constante numérica.	2.4. El sistema adiciona en la región correspondiente al elemento seleccionado, el carácter presionado. Finalizando así el caso de uso.
<b>Flujos Alternos de la sección 2 “Insertar identificador o constante numérica” del paso 2.3</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2.3.1. El investigador inserta una coma o un punto.	2.4.1. El sistema no inserta en la región editable el carácter seleccionado. Finaliza el caso de uso.
<b>Sección 3 “Insertar operación matemática”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.1. El investigador selecciona el elemento donde desea insertar la operación matemática.	3.2. El sistema pasa el foco del teclado al elemento seleccionado.
3.3. El investigador presiona el botón asociado a una de las siguientes operaciones matemáticas: <ul style="list-style-type: none"> <li>➤ Suma</li> <li>➤ Resta</li> <li>➤ Multiplicación</li> <li>➤ División</li> <li>➤ Exponencial</li> <li>➤ Radical</li> <li>➤ Paréntesis</li> <li>➤ Subíndice</li> </ul>	3.4. El sistema inserta el elemento representativo de la expresión seleccionada. El elemento que seleccionó el investigador en el evento 3.2 pasa a ser parte de la nueva operación insertada como uno de los operandos. Finaliza el caso de uso.
<b>Flujos Alternos de la sección 3 “Insertar operación matemática” del paso 3.3</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.3.1. El investigador no presiona el botón asociado a alguna operación matemática.	3.3.1. El sistema no inserta el elemento representativo de la expresión seleccionada. Finaliza el caso de uso.
<b>Sección 4 “Eliminar elemento”</b>	

## CAPÍTULO 2. Características del Sistema

Acción del Actor	Respuesta del Sistema
4.1. El investigador selecciona el elemento a eliminar.	4.2. El sistema pasa el foco de teclado al elemento seleccionado.
4.3. El investigador presiona la tecla “delete”.	4.4. El sistema elimina el elemento. Finaliza el caso de uso.
Flujos Alternos de la sección 4 “Eliminar elemento” del paso 4.3	
Acción del Actor	Respuesta del Sistema
4.3.1. El investigador no selecciona el elemento que desea eliminar.	4.4.1. El sistema no elimina el elemento. Finaliza el caso de uso.
<b>Postcondiciones</b>	El sistema muestra un sistema de ecuaciones diferenciales.

**Tabla 3 Descripción del CU Exportar SED.**

<b>Caso de Uso</b>	Exportar SED.
<b>Actor</b>	Investigador (inicia).
<b>Propósito</b>	Exportar el sistema de ecuaciones diferenciales a diferentes formatos.
<b>Resumen</b>	El caso de uso se inicia cuando el investigador selecciona la opción de exportar. Selecciona uno de los diferentes formatos a exportar y la ruta en donde desea almacenar el fichero resultante, finaliza así el caso de uso.
<b>Referencia</b>	RF5, RF6, RF7, RF8, RF9.
<b>Precondición</b>	Debe existir un sistema de ecuaciones creado.
<b>Prioridad</b>	Crítico.
Flujo Normal de Eventos	
Acción del Autor	Respuesta del Sistema
1. El caso de uso se inicia cuando el investigador desea exportar un archivo y selecciona en el menú File/Export to/, uno de los siguientes formatos: <ul style="list-style-type: none"> <li>➤ Exportar a MathML File.</li> <li>➤ Exportar a Java File.</li> <li>➤ Exportar a Matlab File.</li> <li>➤ Exportar a Mathematica File.</li> <li>➤ Exportar como Imagen.</li> </ul>	2. El sistema muestra una ventana donde el investigador selecciona el lugar donde desea exportar el archivo.
3. El investigador introduce el nombre del	4. En dependencia del formato seleccionado, el

## CAPÍTULO 2. Características del Sistema

<p>archivo y escoge la ubicación del fichero, luego presiona el botón “Guardar”.</p>	<p>sistema realiza lo siguiente:</p> <ul style="list-style-type: none"> <li>➤ Si se seleccionó MathML File, el sistema guarda un archivo con extensión “.xml”.</li> <li>➤ Si se seleccionó Java File, el sistema guarda un archivo con extensión “.java”.</li> <li>➤ Si se seleccionó Matlab File, el sistema guarda un archivo con extensión “.m”.</li> <li>➤ Si se seleccionó Mathematica File, el sistema guarda un archivo con extensión “.m”.</li> <li>➤ Si se seleccionó Imagen, el sistema guarda un archivo con extensión “.png”.</li> </ul> <p>Se finaliza el caso de uso.</p>
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.1. El investigador selecciona el botón “Cancelar”.	4.1. El sistema cierra la ventana y finaliza el caso de uso.
<b>Postcondiciones</b>	Se crea un archivo con el sistema de ecuaciones en el formato especificado.

**Tabla 4 Descripción del CU Importar ficheros.**

<b>Caso de Uso</b>	Importar ficheros.
<b>Actor</b>	Investigador (inicia).
<b>Propósito</b>	Cargar un modelo matemático desde un fichero con formato MathML Presentation.
<b>Resumen</b>	El caso de uso se inicia cuando el investigador selecciona la opción importar archivo. Luego se especifica la localización física del modelo y finaliza cuando el sistema carga el modelo especificado.
<b>Referencia</b>	RF10.
<b>Precondición</b>	Debe existir un fichero .xml que contenga al modelo matemático.
<b>Prioridad</b>	Crítico.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Autor</b>	<b>Respuesta del Sistema</b>

## CAPÍTULO 2. Características del Sistema

1. El caso de uso se inicia cuando el investigador selecciona el menú File/ Import from.../ MathML Presentación.	2. El sistema muestra una ventana donde el investigador busca el archivo en el formato deseado.
3. El investigador busca el archivo deseado y presiona el botón "Aceptar".	4. El sistema carga el modelo matemático del archivo seleccionado y lo muestra en el editor de ecuaciones.
<b>Flujos Alternos del paso 3</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.1. El investigador busca el archivo deseado y presiona el botón "Cancelar".	4.1. El sistema cierra la ventana. Finaliza el caso de uso.
<b>Flujos Alternos del paso 4</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	El sistema no carga el modelo matemático del archivo seleccionado ya que no tiene extensión .xml. Se muestra el mensaje de error: " <i>Cont is not allowed in prolog</i> ". Finaliza el caso de uso.
<b>Postcondiciones</b>	Modelo matemático cargado en el sistema.

**Tabla 5 Descripción del CU Modificar fuente.**

<b>Caso de Uso</b>	Modificar fuente.
<b>Actor</b>	Investigador (inicia).
<b>Propósito</b>	Modificar la fuente del Editor de Ecuaciones brindándole una vista al usuario más amigable.
<b>Resumen</b>	El caso de uso se inicia cuando el investigador decide cambiar la fuente de las ecuaciones y selecciona las opciones Font, Size, Foreground y Background. Una vez especificada la fuente del editor, finaliza el caso de uso cuando el sistema cambia la fuente.
<b>Referencia</b>	RF11.
<b>Precondición</b>	Ninguna.
<b>Prioridad</b>	Secundario.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Autor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el	2. El sistema en dependencia de lo

## CAPÍTULO 2. Características del Sistema

<p>investigador selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>➤ Font.</li> <li>➤ Size.</li> <li>➤ Foreground.</li> <li>➤ Background.</li> </ul>	<p>seleccionado, realiza lo siguiente:</p> <ul style="list-style-type: none"> <li>➤ Si seleccionó Font, ir a la sección “Font”.</li> <li>➤ Si seleccionó Size, ir a la sección “Size”.</li> <li>➤ Si seleccionó Foreground, ir a la sección “Foreground”.</li> <li>➤ Si seleccionó Background, ir a la sección “Background”.</li> </ul>
<b>Sección 1 “Font”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1. El sistema muestra un menú desplegable con diferentes tipos de fuente.
1.2. El investigador selecciona el tipo de fuente.	1.3. El sistema cambia el tipo de fuente y finaliza el caso de uso.
<b>Flujos Alternos de la sección 1 “Font” del paso 1.2</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.2.1. El investigador no selecciona ningún tipo fuente.	1.3.1. Se finaliza el caso de uso.
<b>Sección 2 “Size”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.1. El sistema muestra un menú desplegable con los tamaños de la fuente.
2.2. El investigador selecciona el tamaño que desea de fuente.	2.3. El sistema cambia el tamaño de la fuente y finaliza el caso de uso.
<b>Flujos Alternos de la sección 2 “Size” del paso 2.2</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2.2.1. El investigador no selecciona ningún tamaño de fuente.	2.3.1. Se finaliza el caso de uso.
<b>Sección 3 “Foreground”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	3.1. El sistema muestra una ventana con diferentes opciones de colores de fuentes para las ecuaciones.
3.2. El investigador selecciona el color de	3.3. El sistema cambia el color de la fuente y



## CAPÍTULO 2. Características del Sistema

fuelle que desea y oprime el botón “Aceptar”.	finaliza el caso de uso.
<b>Flujos Alternos de la sección 3 “Foreground” del paso 3.2</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.2.1. El investigador selecciona la opción “Cancelar”.	3.3.1. Se finaliza el caso de uso.
<b>Sección 4 “Background”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1. El sistema muestra una ventana con diferentes opciones de colores de fondo para las ecuaciones.
4.2. El investigador selecciona el color de fondo que desea y oprime el botón “Aceptar”.	4.3. El sistema cambia el color fondo de las ecuaciones y finaliza el caso de uso.
<b>Flujos Alternos de la sección 3 “Foreground” del paso 4.2</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4.2.1. El investigador selecciona la opción “Cancelar”.	4.3.1. Se finaliza el caso de uso.
<b>Postcondiciones</b>	El sistema cambia la fuente del editor.

### 2.5 Conclusiones del capítulo

En este capítulo se propuso una explícita definición de los requisitos funcionales con los que debe contar el sistema para que sea exitoso, confiable, usable, con la ayuda de los requisitos no funcionales, así como los casos de uso del sistema y el actor. Se definieron 11 requisitos funcionales agrupados en 4 casos de uso. Los casos de uso sirven de cimiento para la continuación de las restantes fases del proceso de diseño e implementación. Se obtuvo una lucidez y precisión en cuanto al sistema a construir.

### CAPÍTULO 3. DISEÑO DE LA APLICACIÓN

#### Introducción

En el presente capítulo se muestra una panorámica del estilo arquitectónico empleado para el desarrollo del Editor. Se precisan los patrones de diseño necesarios para hacer un acercamiento a los resultados que se desean obtener y por último se muestra el diagrama de clases del diseño, así como los diagramas de secuencia y el modelo de despliegue.

#### 3.1 Patrón Arquitectónico utilizado

Los patrones arquitectónicos son los encargados de definir la estructura de un sistema. Especifican un conjunto predefinido de subsistemas con sus responsabilidades. Constan de una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño. En vistas de desarrollar la aplicación se hizo necesario la utilización del patrón arquitectónico Modelo-Vista-Controlador (MVC), el mismo es empleado en el software BioSyS y en la Plataforma de Servicios Bioinformáticos. La figura 5, muestra la aplicación del patrón MVC en el nuevo Editor de Ecuaciones. MVC, es un patrón de arquitectura empleado especialmente en aplicaciones que operan con complejas transacciones y gran cantidad de datos, donde es primordial la separación de conceptos para que el desarrollo sea mejor estructurado, proporcionando la programación en diferentes capas de manera paralela e independiente. Separa los componentes de aplicación en tres niveles, interfaz de usuario (vista), lógica de control (controlador) y lógica de negocio (modelo). (25)

- **Modelo:** La clase `Ecuacion.java` representa específicamente el dominio de la información sobre la cual funciona la aplicación. Está relacionada con la clase `Control.java`, quien invoca sus cambios y la actualiza. Una vez actualizado el modelo, la clase `Ecuacion.java` manda las acciones producidas en ella a la vista, sin hacer uso del controlador.
- **Vista:** La clase `PanelPrincipal.java` representa el modelo en un formato adecuado para que el usuario pueda interactuar con la aplicación a través de una interfaz. Muestra la información al usuario. Se encuentra asociada al componente controlador. Esta clase puede directamente cambiar alguna información en el modelo sin la intervención del controlador.
- **Controlador:** La clase `Control.java` responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

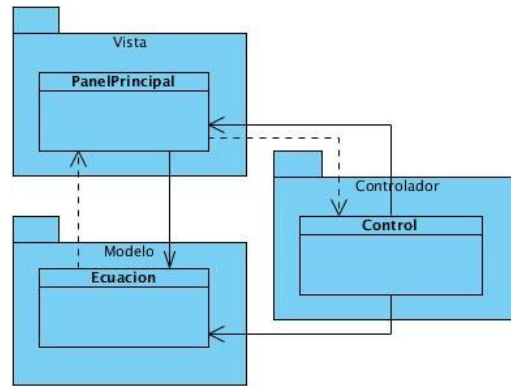


Figura 5 Patrón MVC para la versión 2.0 del Editor de Ecuaciones.

## 3.2 Patrones de Diseño utilizados

Los patrones de diseño expresan esquemas utilizados en la definición de estructuras de diseño, o bien sus relaciones, para la construcción de sistemas de software. (26) Seguidamente se muestran ejemplos de patrones utilizados en el diseño.

### Patrón Experto

El patrón experto está basado en asignar una responsabilidad al experto en la información, lo que significa que la responsabilidad de crear un objeto o implementar un método debe recaer sobre la clase que conoce la información necesaria para su creación. (27) La clase `CajaBasica.java` muestra la aplicación de este patrón de diseño (ver figura 6). Esta clase contiene la información necesaria para ser capaz de crear objetos de la misma y así conservar el encapsulamiento.

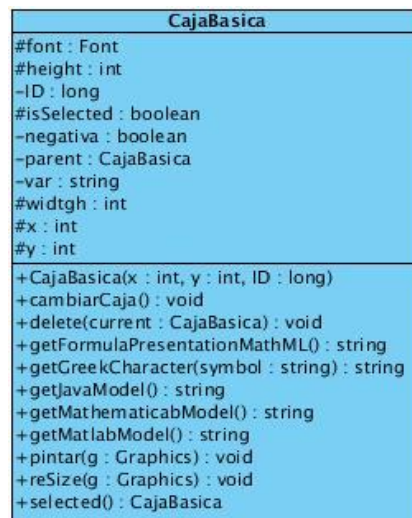


Figura 6 Ejemplo de la aplicación del Patrón Experto.

## CAPÍTULO 3. Diseño de la Aplicación.

### 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. (27)

- B agrega los objetos de A.
- B contiene los objetos de A.
- B registra las instancias de los objetos de A.
- B tiene los objetos de inicialización que serán enviados a A cuando este objeto sea creado.

La figura 7, es un ejemplo de cómo se aplica el patrón creador. La clase `PanelPrincipal.java`, tiene una instancia de la clase `Lienzo.java`, a la cual le delega la funcionalidad de mostrar los modelos que se están editando.

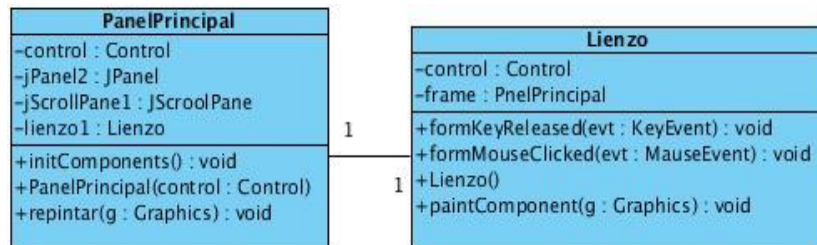


Figura 7 Ejemplo de la aplicación del Patrón Creador.

### Patrón Controlador

Es un objeto de interfaz que se encarga de manejar un evento del sistema. Define además el método de su operación. Garantiza que los procesos de dominio sean manejados por la capa de los objetos de dominio y no por la interfaz. (27)

Un ejemplo de la aplicación de este patrón es la clase `Control.java`, es la encargada de manejar los eventos ejecutados en las clases `CajaBasica.java` y `PanelPrincipal.java` (ver figura 8). Asigna la responsabilidad del manejo de mensajes de los eventos del sistema a la clase `CajaBasica.java`, por ejemplo, donde representa un manejador artificial de todos los eventos del sistema del caso de uso Gestionar SED.

## CAPÍTULO 3. Diseño de la Aplicación.

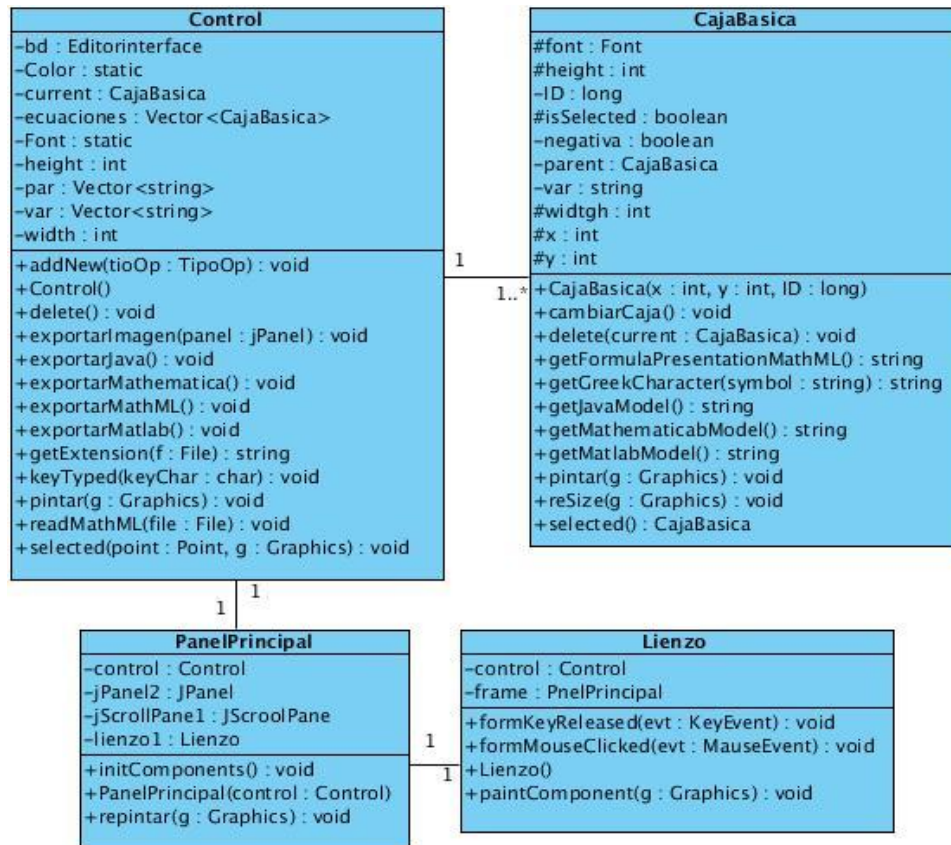


Figura 8 Ejemplo de la aplicación del Patrón Controlador.

### Patrón Alta Cohesión

La información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Una clase con alta cohesión tiene un número relativamente pequeño de métodos, con funcionalidades altamente relacionadas y no realiza mucho trabajo. El grano fino de funcionalidad altamente relacionada incrementa la reutilización porque una clase cohesiva se puede utilizar para un propósito muy específico. (27)

Las clases `Division.java`, `Suma.java`, `Resta.java`, `Potencia.java` son ejemplos de la aplicación del patrón alta cohesión. Ayudan a incrementar la claridad y facilita la comprensión del diseño, se simplifica el mantenimiento y las mejoras. En la figura 9, se puede observar la clase `Division.java`, la misma colabora con la clase `CajaBasica.java` para compartir el esfuerzo en caso de que el usuario que esté interactuando con el sistema desee realizar una operación de división en la ecuación en donde esté trabajando.

## CAPÍTULO 3. Diseño de la Aplicación.

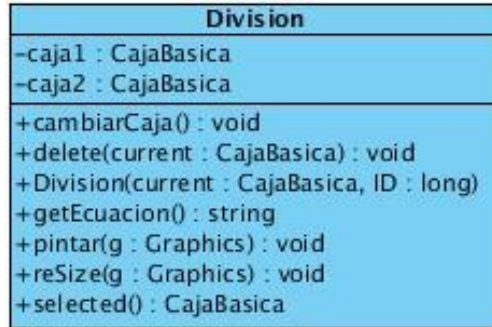


Figura 9 Ejemplo de la aplicación del Patrón Alta Cohesión.

### 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. (27) Un ejemplo de la aplicación del bajo acoplamiento son las clases GreckAlphabet.java, TableroGUI.java y CasillasGUI.java (ver figura 10). Es aplicado para apoyar al diseño de estas clases independientes, para reducir el impacto de los cambios. No afecta los cambios en otros componentes. Es fácil de entender de manera aislada y conveniente para reutilizar.

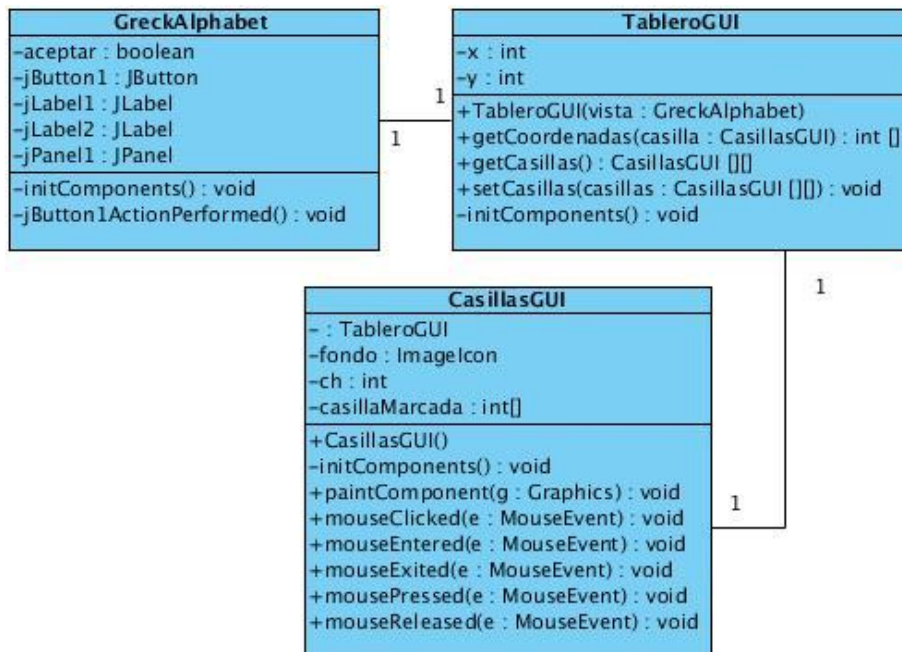


Figura 10 Ejemplo de la aplicación del Patrón Bajo Acoplamiento.

## CAPÍTULO 3. Diseño de la Aplicación.

### Patrón Fachada

La figura 11 es una muestra de la aplicación del patrón fachada. La clase `PanelPrincipal.java` provee de una interfaz unificada simple el acceso a una interfaz o grupo de interfaces de un subsistema. Es la interfaz de alto nivel que permite usar fácilmente cualquier subsistema.

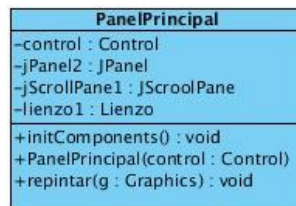


Figura 11 Ejemplo de la aplicación del Patrón Fachada.

### Patrón Observador

La figura 12, evidencia el uso del patrón observador en el sistema. La clase `Ecuacion.java` notifica a la clase `PanelPrincipal.java` de algún cambio que no requiera la utilización del controlador y viceversa. Por ejemplo cuando está seleccionada una región editable en la clase `PanelPrincipal.java`, esta clase notifica al modelo directamente. En general, un objeto de la vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista.

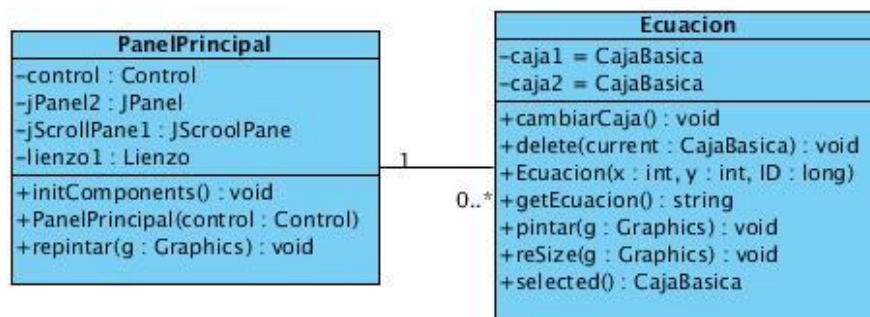


Figura 12 Ejemplo de la aplicación del Patrón Observador.

### 3.3 Modelo de Diseño

El modelo de diseño es la visión general del sistema que se refina sucesivamente, garantizando su calidad. Se le considera como la abstracción de modelo de implementación y por consiguiente su código fuente, ya que se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas con la implementación.

# CAPÍTULO 3. Diseño de la Aplicación.

Representa a los casos de uso en el dominio de la solución. Seguidamente se abordarán los temas relacionados con el diseño de la aplicación. (28)

## 3.3.1 Diagrama de Clases del Diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces (las de Java, por ejemplo) en una aplicación. Especifica la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Es utilizado durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. (27) A continuación se muestra en la figura 13 el diagrama de clases del Editor de Ecuaciones del software BioSys.

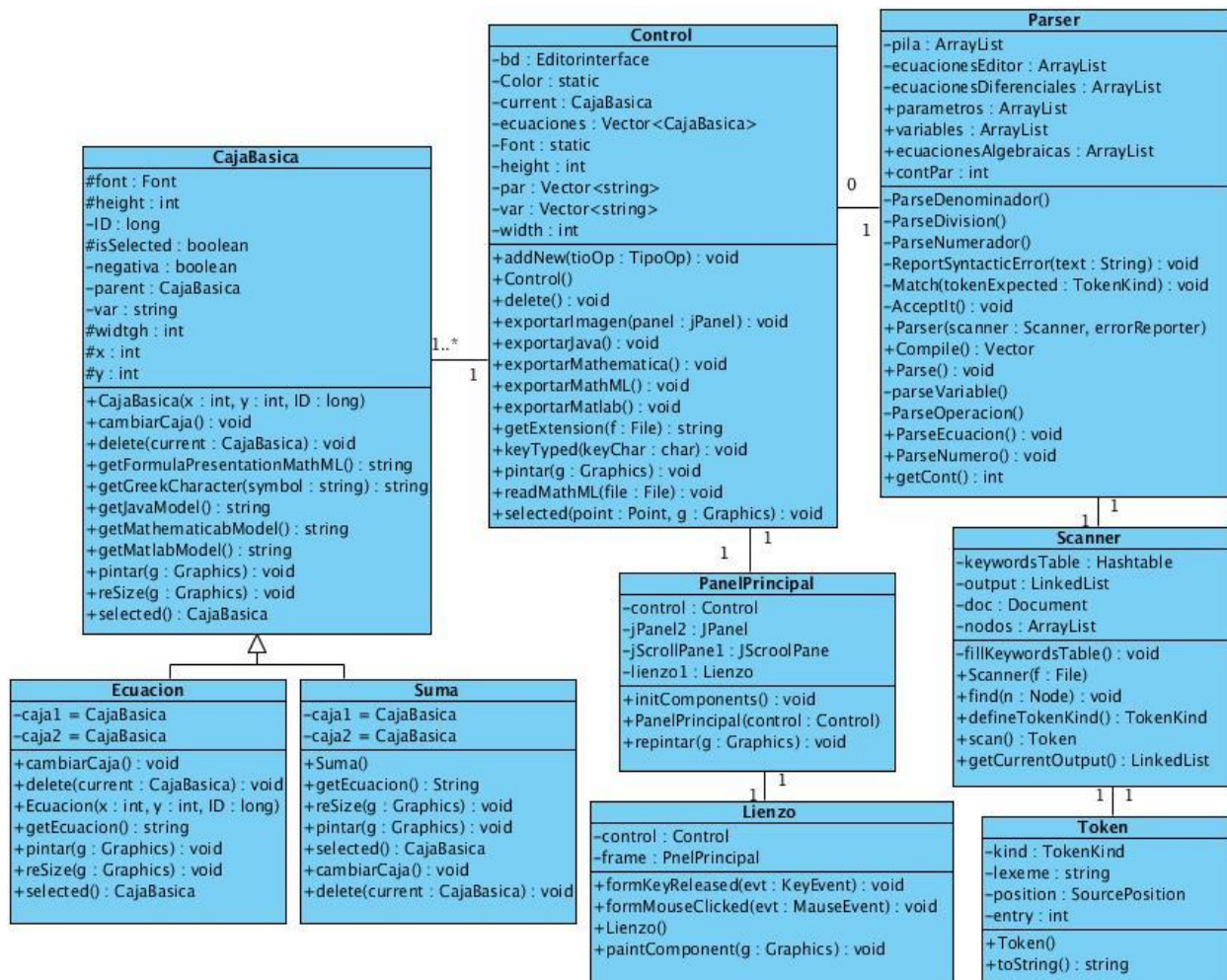


Figura 13 Diagrama de Clases del Diseño.



## CAPÍTULO 3. Diseño de la Aplicación.

Las clases `PanelPrincipal.java` y `Lienzo.java` son utilizadas para mostrar los modelos con los que se están trabajando en el Editor. Las clases `Control.java`, `CajaBasica.java`, `Ecuacion.java`, `Suma.java`, `Radical.java`, `Raiz.java`, `RaizCuadrada.java`, `Resta.java`, `Division.java`, `Potencia.java`, `Multiplificacion.java`, `Differential.java`, `DifferentialEcuacion.java`, `SubIndex.java` y `Parentesis.java` permiten realizar a través de sus funcionalidades las acciones que se llevan a cabo en el caso de uso Gestionar SED, así como el caso de uso Modificar fuente. En la clase `Control.java` se encuentran funcionalidades que permiten Exportar SED a diferentes formatos; entre tanto, la misma clase auxiliándose de las clases `Parser.java`, `Scanner.java` y `Token.java` permite efectuar el proceso de Importar ficheros en formato MathML Presentation.

### 3.3.2 Diagramas de Secuencia

Los diagramas de secuencia son usados para describir gráficamente un caso de uso o un escenario mediante líneas verticales y los mensajes conectados entre objetos, como flechas. A continuación se muestran los diagramas de secuencia por cada caso de uso.

Los diagramas de secuencias de las figuras 14, 15, 16 y 17, muestran la comunicación de los componentes del software en cuestión, durante la ejecución del caso de uso Gestionar SED. Se define una clase controladora llamada `Control.java` encargada de gestionar todas las peticiones del usuario que se producen mediante la clase interfaz `PanelPrincipal.java` e invoca los cambios solicitados por el usuario en las clases de entidad `CajaBasica.java` o `Ecuacion.java`.

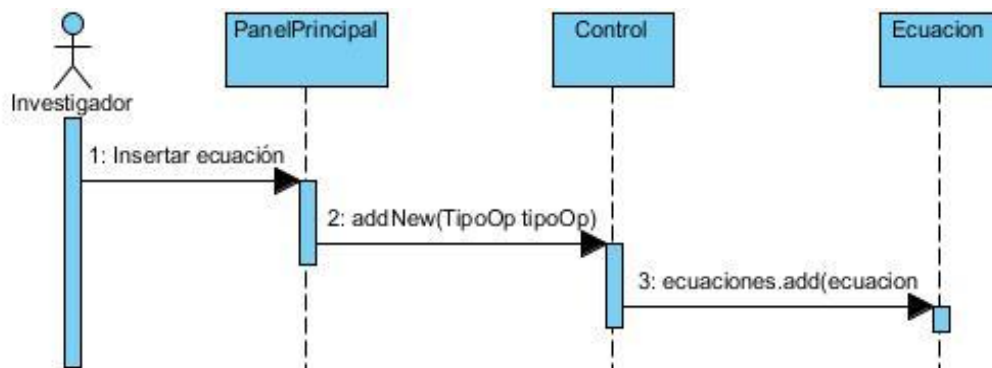


Figura 14 Diagrama de Secuencia del CU Gestionar SED (sección Adicionar nueva ecuación).

## CAPÍTULO 3. Diseño de la Aplicación.

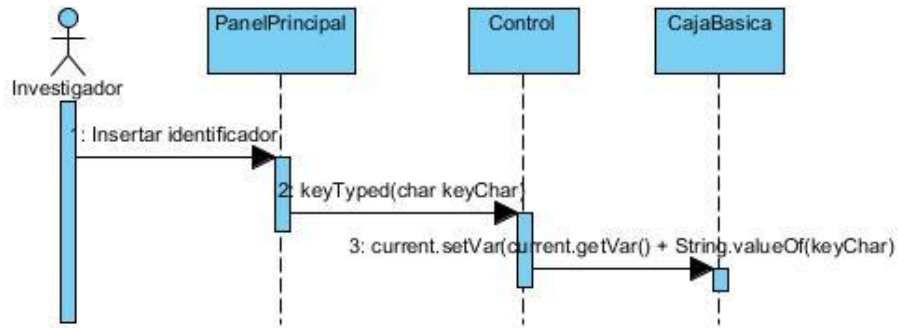


Figura 15 Diagrama de Secuencia del CU Gestionar SED (sección Insertar identificador o constante numérica).

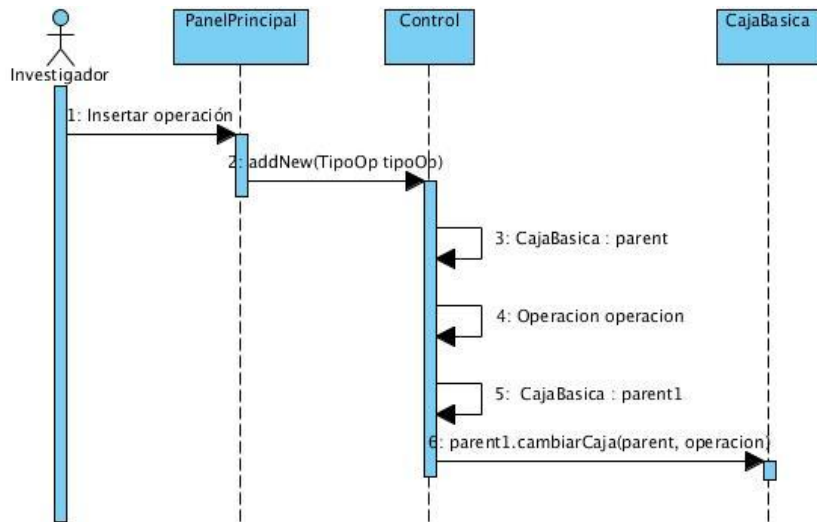


Figura 16 Diagrama de Secuencia del CU Gestionar SED (sección Insertar operación matemática).

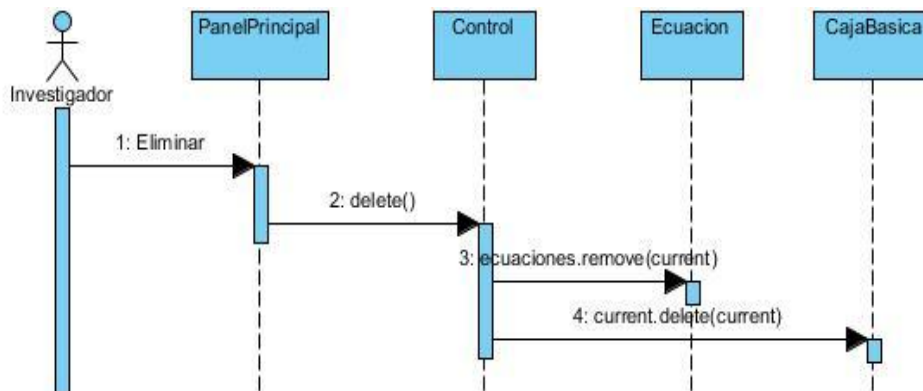


Figura 17 Diagrama de Secuencia del CU Gestionar SED (sección Eliminar elemento).

## CAPÍTULO 3. Diseño de la Aplicación.

---

Diagrama de Secuencia del Caso de Uso Exportar SED. [Ver Anexo 1].

Diagrama de Secuencia del Caso de Uso Importar ficheros. [Ver Anexo 2].

Diagrama de Secuencia del Caso de Uso Modificar fuente. [Ver Anexo 3].

### 3.3.3 Modelo de Despliegue

El modelo de despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto. (29)

En la figura 18, se presenta el modelo de despliegue del Editor de Ecuaciones del software BioSyS, en el cual una PC cliente por medio de un navegador, a través del protocolo de red HTTP realiza peticiones al Servidor de Aplicaciones, en donde se encapsula la lógica de aplicación.

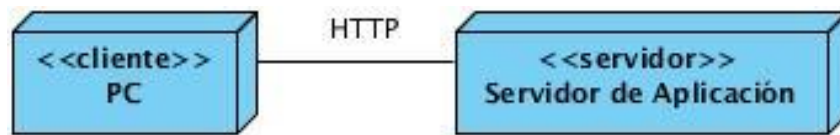


Figura 18 Modelo de Despliegue.

### 3.4 Conclusiones del capítulo

En este capítulo se realizó la selección del patrón arquitectónico Modelo-Vista-Controlador para la implementación de la aplicación. Con la aplicación de los patrones: Experto, Creador, Controlador, Alta Cohesión, Bajo Acoplamiento, Observador y Fachada, se logró un diseño robusto que garantiza en cierta medida la escalabilidad. Se mostró el modelo de diseño, donde se presenta el diagrama de clases, los diagramas de secuencia y el modelo de despliegue, para dar paso a la implementación.

## CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA

### Introducción

En el capítulo actual se realiza una descripción de la implementación del sistema, especificando el diagrama de componentes de las funcionalidades desarrolladas. También se presenta toda la información relacionada con las pruebas de caja negra y caja blanca realizadas al sistema.

#### 4.1 Implementación

La implementación es el proceso mediante el cual se distribuye una aplicación o componente finalizado para su instalación en otros equipos. Es un proceso de varias fases que empieza cuando se crea una aplicación en el equipo de un desarrollador y termina cuando está instalada y lista para ejecutarse en el equipo de un usuario. (30)

##### 4.1.1 Diagrama de Componentes

Los diagramas de componentes son empleados para modelar la vista estática y dinámica de un sistema y documentar cualquier arquitectura del mismo. Representan cómo un sistema de software es dividido en componentes, muestra la organización y las dependencias entre un conjunto de componentes. (30) Seguidamente se muestra una representación del diagrama de componentes del Editor de Ecuaciones, versión 2.0.

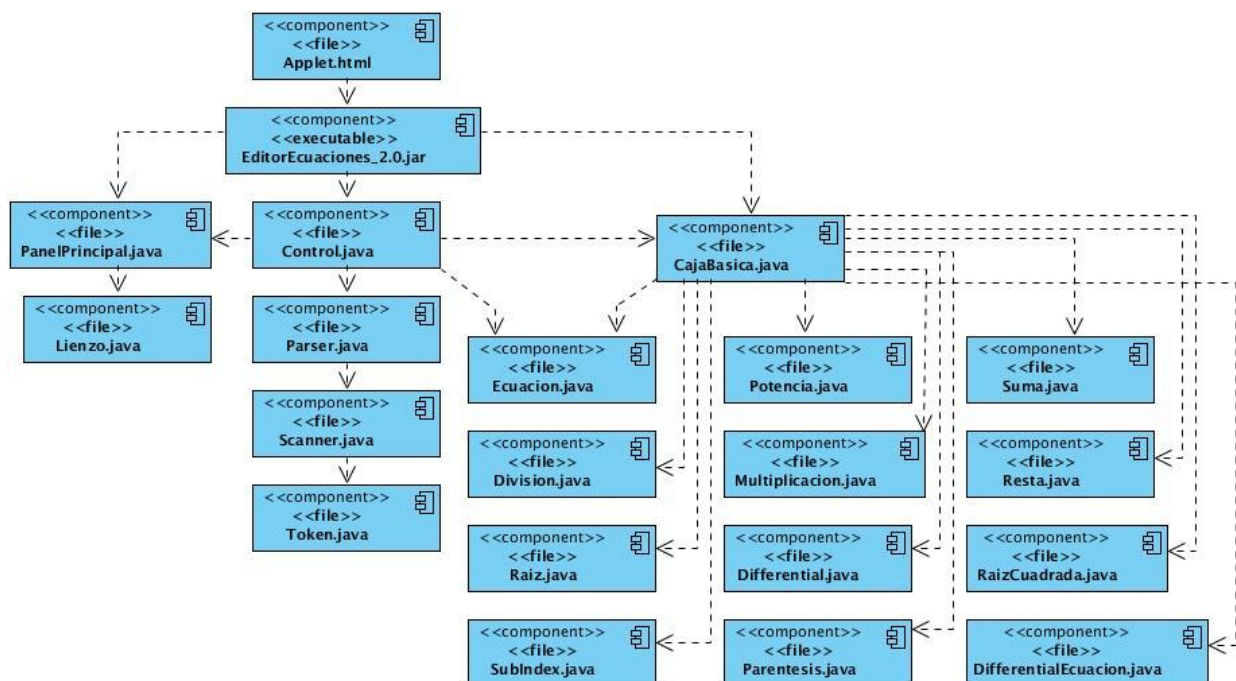


Figura 19 Diagrama de Componentes.

## CAPÍTULO 4. Implementación y Prueba

---

A continuación se presenta una breve descripción de los componentes representados en el diagrama de la figura 19.

**Applet.html:** Representa el archivo .html, que permite mostrar la aplicación *desktop* en la web.

**EditorEcuaciones\_2.0.jar:** Representa el ejecutable de la aplicación.

**PanelPrincipal.java:** Representa la clase interfaz que permite mostrar el Editor, para que el investigador pueda realizar la edición de ecuaciones.

**Lienzo.java:** Representa la clase interfaz que muestra el área de trabajo, donde se escriben las ecuaciones.

**Control.java:** Representa la clase que tiene el control de todas las acciones que son ejecutadas en el Editor.

Los componentes **Parser.java**, **Scanner.java** y **Token.java** son los representativos de las clases que permiten verificar la correcta estructura de un modelo matemático cuando se desea importar.

**CajaBasica.java:** Representa la clase que permite estructurar las ecuaciones, así como la posición que ocupan en el área de trabajo.

**Ecuacion.java:** Este componente representa la clase que contiene la estructura de una ecuación, es una especificación de la clase **CajaBasica.java**.

Los componentes **SubIndex.java**, **DifferentialEcuacion.java**, **Parentesis.java**, **Raiz.java**, **Multiplicacion.java**, **Differential.java**, **Potencia.java**, **Suma.java**, **Resta.java**, **RaizCuadrada.java** y **Division.java** representan a las clases que permiten insertar las operaciones en las ecuaciones.

# CAPÍTULO 4. Implementación y Prueba

## 4.1.2 Imágenes del Editor de Ecuaciones, versión 2.0

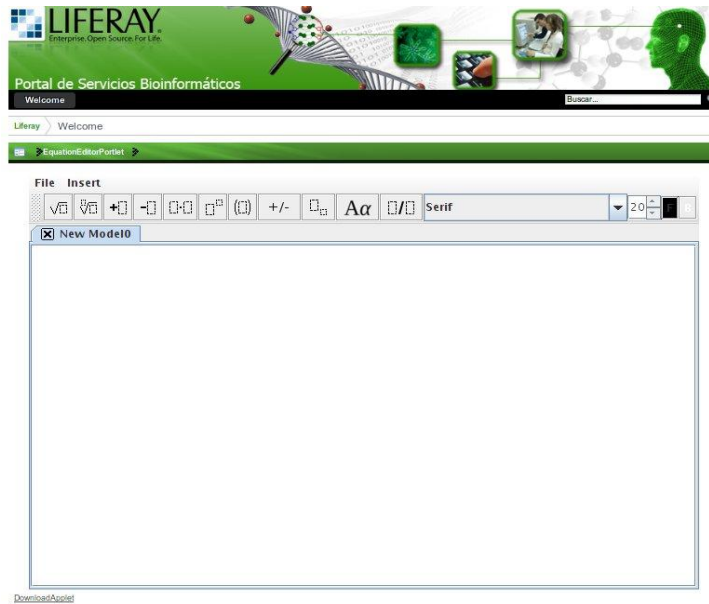


Figura 20 Editor de Ecuaciones, versión 2.0, en el Portal de Servicios Bioinformáticos.

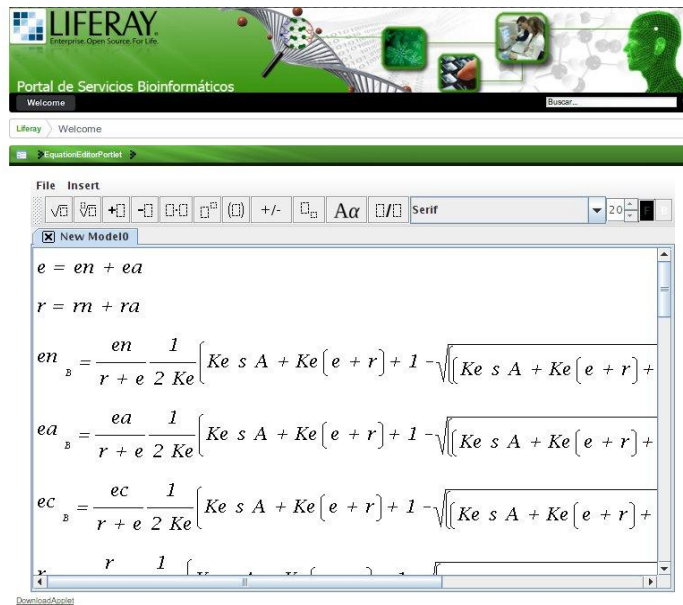


Figura 21 Editor de Ecuaciones, versión 2.0.

# CAPÍTULO 4. Implementación y Prueba

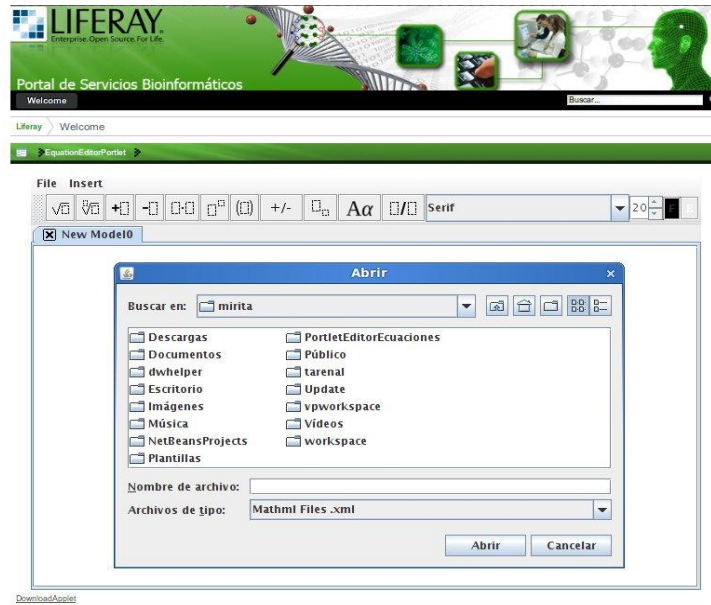


Figura 22 Editor de Ecuaciones, versión 2.0, importando un fichero.

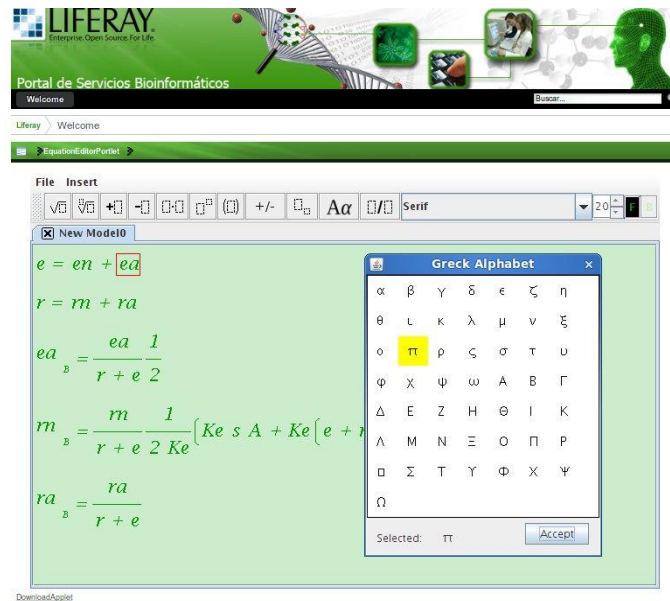


Figura 23 Editor de Ecuaciones, versión 2.0, insertando caracteres griegos.

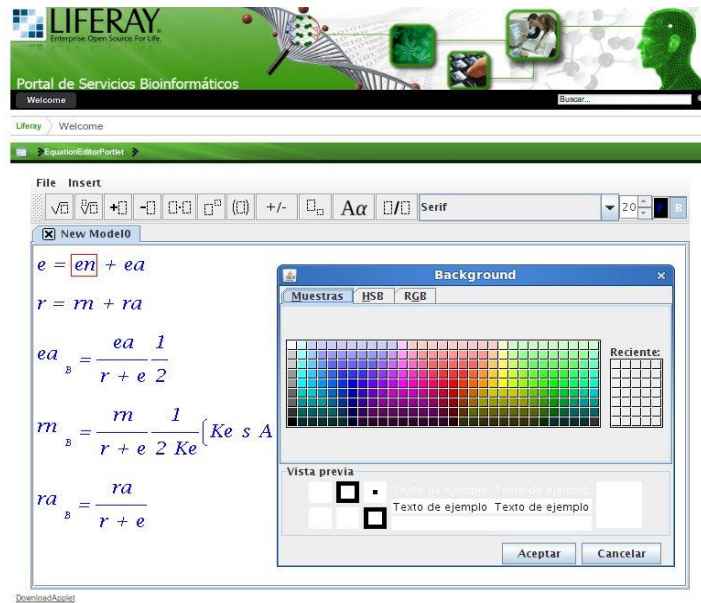


Figura 24 Editor de Ecuaciones, versión 2.0, cambiando la fuente de las ecuaciones.

## 4.2 Pruebas de software

Los niveles de pruebas son los escenarios de trabajo y los diferentes tipos de objetivos en donde se aplican las pruebas. Se conocen por el nombre de prueba de desarrollador, independiente, unidad, integración, sistema y aceptación. Para la realización de las pruebas al Editor de Ecuaciones se seleccionó las de sistema.

Se conoce con el nombre de pruebas de sistema a aquellas pruebas que toman el sistema de información al completo y lo prueban tanto en su conjunto como en sus relaciones con otros sistemas con los que se comuniquen. Son la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. Permiten verificar que tanto las especificaciones funcionales como las técnicas se cumplan para el entregable. Además, si el entorno en el que se realizan estas pruebas es equivalente al de producción, permiten obtener una visión sobre su comportamiento que podrá extrapolarse a dicho entorno. (31)

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Para la realización de las pruebas en la aplicación se utilizaron los métodos caja negra y caja blanca.

Las pruebas de caja negra son las pruebas que se llevan a cabo sobre la interfaz del software, se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales y el



## **CAPÍTULO 4. Implementación y Prueba**

---

probador se limita a suministrarle datos como entrada y estudiar la salida. Demuestran que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, manteniendo la integridad de la información externa por dentro. La técnica aplicada para la realización de las pruebas al Editor, es partición de equivalencia, la cual, divide el campo de entrada en clases de datos que tienden a ejecutar determinadas funciones del software. (32)

Las pruebas de caja blanca, por el contrario, realizan un examen escrupuloso de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado. Una de las técnicas más utilizadas en caja blanca es camino básico. En el editor también es aplicada, ya que esta prueba permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. (32)

Para comprobar si se cumplió el objetivo general de la creación del Editor de Ecuaciones 2.0, se aplicó el método experimental. Dicho método se utiliza para el estudio de un objeto en el cual el investigador crea las condiciones o adapta las existentes para el esclarecimiento de las propiedades, leyes y relaciones del objeto. El experimento es una actividad científica dirigida a comprobar la validez de una hipótesis. Es característico de las investigaciones cuantitativas. En las cualitativas no existe el experimento. (33)

### **4.2.1 Plan de Prueba**

El propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. A través del plan de pruebas se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del software.

### **Configuración del entorno de prueba**

Para la correcta realización de las pruebas al sistema, es preciso efectuar la configuración del entorno de prueba. Se considera que la ejecución de un proceso de pruebas es exitosa, cuando se toman en cuenta los recursos de software y hardware que necesita el producto que se está construyendo. En vistas de minimizar los errores de la aplicación en desarrollo, se hizo

## CAPÍTULO 4. Implementación y Prueba

necesario los siguientes requerimientos de hardware y de software que hicieron posible un mejor desarrollo de las pruebas.

Requerimientos de software:

- PC con Windows XP o superior.
- PC con Linux.
- Máquina Virtual de Java 1.6 o versiones superiores.

Requerimientos de hardware:

- Memoria RAM no menos de 512 MB.
- Puertos USB para memoria externa.
- Procesadores Pentium IV o superiores.

**Tabla 6 Plan de Pruebas.**

Fecha de inicio	Fecha de finalizado	Actividades	Personal implicado
2/4/2013	2/4/2013	Aceptación y firma del Plan de Prueba.	Mirian Sáez Rodríguez, Vilmavis La Rosa Sordo, Edel Moreno Lemus
9/4/2013	9/4/2013	Verificación de las condiciones previas para el inicio de las pruebas.	Mirian Sáez Rodríguez, Vilmavis La Rosa Sordo
14/4/2013	15/4/2013	Ejecución de las pruebas caja blanca.	Mirian Sáez Rodríguez, Edel Moreno Lemus
16/4/2013	6/5/2013	Ejecución de las pruebas caja negra.	Mirian Sáez Rodríguez
7/5/2013	7/5/2013	Ejecución del método experimental.	Mirian Sáez Rodríguez
<b>Primera Iteración del Editor de Ecuaciones, versión 2.0</b>			
16/4/2013	16/4/2013	Ejecución de las pruebas caja negra al caso de uso Gestionar SED.	Mirian Sáez Rodríguez
17/4/2013	17/4/2013	Ejecución de las pruebas caja negra al caso de uso Exportar SED.	Mirian Sáez Rodríguez
18/4/2013	18/4/2013	Ejecución de las pruebas caja negra al caso de uso Importar ficheros.	Mirian Sáez Rodríguez
19/4/2013	19/4/2013	Ejecución de las pruebas caja negra al caso de uso Modificar fuente.	Mirian Sáez Rodríguez

## CAPÍTULO 4. Implementación y Prueba

Segunda Iteración del Editor de Ecuaciones, versión 2.0			
3/5/2013	3/5/2013	Ejecución de las pruebas caja negra al caso de uso Gestionar SED.	Mirian Sáez Rodríguez
4/5/2013	4/5/2013	Ejecución de las pruebas caja negra al caso de uso Exportar SED.	Mirian Sáez Rodríguez
5/5/2013	5/5/2013	Ejecución de las pruebas caja negra al caso de uso Importar ficheros.	Mirian Sáez Rodríguez
6/5/2013	6/5/2013	Ejecución de las pruebas caja negra al caso de uso Modificar fuente.	Mirian Sáez Rodríguez

### 4.2.2 Diseño de las Pruebas Caja Blanca

Se decidió utilizar la técnica camino básico a los métodos “selected” de la clase `Ecuacion.java` y “pintar” de la clase `Control.java`, ya que los mismos tienen gran impacto sobre el funcionamiento del Editor.

#### Técnica Camino Básico en el método “selected”

El método “selected” (ver figura 25) permite seleccionar una región editable, que es utilizada por el investigador para insertar identificadores, constantes numéricas, operadores matemáticos, caracteres griegos o puede ser utilizado también para eliminar cualquier elemento de la ecuación.

```
public CajaBasica selected(double x1, double y1, Graphics g) {
    if (x1 >= caja1.getX() && x1 <= caja2.getX() + caja2.getWidth() &&
1   y1 >= Math.min(caja2.getY(), caja1.getY()) &&
   y1 <= Math.max(caja2.getY(), caja1.getY()) + Math.max(caja2.getHeight(),
   caja1.getHeight())) {
2   CajaBasica selected = caja1.selected(x1, y1, g);
   CajaBasica selected1 = caja2.selected(x1, y1, g);
3   if (selected != null) {
4   return selected;
   }
5   if (selected1 != null) {
6   return selected1;
   }
7   return this;
   }
8   return null;
}
```

Figura 25 Código del método “selected”.

#### Grafo de control

Los nodos en azul más fuerte, representan las condicionales y los predicados, en el gráfico de la figura 26.

## CAPÍTULO 4. Implementación y Prueba

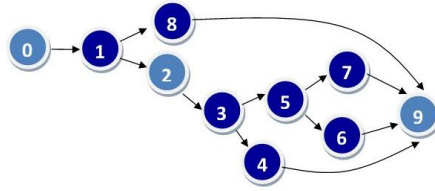


Figura 26 Gráfico de control del método “selected”.

### Complejidad Ciclomática

$$V(G) = 12 \text{ aristas} - 10 \text{ nodos} + 2$$

$$V(G) = 2 + 2$$

$$V(G) = 4$$

### Caminos

1. 0-1-8-9
2. 0-1-2-3-5-7-9
3. 0-1-2-3-5-6-9
4. 0-1-2-3-4-9

### Caso de prueba para cada camino

- Camino 1: 0-1-8-9. Se cumple que las posiciones de las cajas son incorrectas. Se retorna nulo.
- Camino 2: 0-1-2-3-5-7-9. Se cumple que las posiciones de las cajas son correctas, se le asigna a la variable `selected` el valor de la selección de la caja1 y a `selected1` el valor de la selección de la caja2. En caso de que `selected` y `selected1` tengan valor nulo se retorna *this*.
- Camino 3: 0-1-2-3-5-6-9. Se cumple que las posiciones de las cajas son correctas, se le asigna a la variable `selected` el valor de la selección de la caja1 y a `selected1` el valor de la selección de la caja2. En caso de que `selected` tenga valor nulo y `selected1` tenga valor distinto de nulo, se retorna `selected1`.
- Camino 4: 0-1-2-3-4-9. Se cumple que las posiciones de las cajas son correctas, se le asigna a la variable `selected` el valor de la selección de la caja1 y a `selected1` el valor de la selección de la caja2. Se retorna `selected` por tener valor distinto de nulo.

### Técnica Camino Básico en el método “pintar”

## CAPÍTULO 4. Implementación y Prueba

El método “pintar” (ver figura 27), consiste en dibujar la ecuación en el área de trabajo, es el encargado de que las ecuaciones estén bien alineadas; basándose en las posiciones X y Y del eje de coordenadas cartesianas y del ancho y alto de cada ecuación.

```
public void pintar(Graphics g) {
    width = 0;
    height = 10;
    for (int i = 0; i < ecuaciones.size(); i++) {
        ecuaciones.get(i).setY(height);
        ecuaciones.get(i).reSize(g);
        if (width < ecuaciones.get(i).getWidth()) {
            width = ecuaciones.get(i).getWidth();
        }
        height += ecuaciones.get(i).getHeight() + 20;
    }
    for (int i = 0; i < ecuaciones.size(); i++) {
        ecuaciones.get(i).pintar(g);
    }
}
```

Figura 27 Código del método “pintar”.

### Grafo de control

Los nodos en azul más fuerte, representan las condicionales y los predicados, en el gráfico de la figura 28.

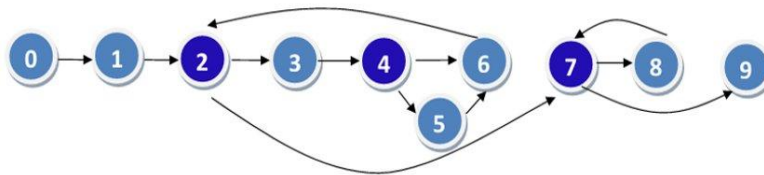


Figura 28 Gráfico de control del método “pintar”.

### Complejidad Ciclomática

$$V(G) = 12 \text{ aristas} - 10 \text{ nodos} + 2$$

$$V(G) = 2 + 2$$

$$V(G) = 4$$

### Caminos

1. 0-1-2-7-9
2. 0-1-2-3-4-6-2-7-9
3. 0-1-2-3-4-5-6-2-7-9

## CAPÍTULO 4. Implementación y Prueba

4. 0-1-2-3-1-5-6-2-7-8-9

### Caso de prueba para cada camino

- Camino 1: 0-1-2-7-9. width = 0 y height = 10. No se recorre la lista de las ecuaciones para actualizarle el valor de height y redimensionar el size; ni se pintan las ecuaciones.
- Camino 2: 0-1-2-3-4-6-2-7-9. width = 0 y height = 10. Se va recorriendo la lista de las ecuaciones para actualizar el valor de height y se va redimensionando el size. Luego se modifica el valor de height con el de la ecuación que se está analizando y se le suma 20.
- Camino 3: 0-1-2-3-4-5-6-2-7-9. width = 0 y height = 10. Se va recorriendo la lista de las ecuaciones para actualizar el valor de height y se va redimensionando el size. El width es menor que el que tiene la ecuación, por lo tanto, se va actualizando. Luego se modifica el valor de height con el de la ecuación que se está analizando y se le suma 20.
- Camino 4: 0-1-2-3-4-5-6-2-7-8-9. width = 0 y height = 10. Se va recorriendo la lista de las ecuaciones para actualizar el valor de height y se va redimensionando el size. El width es menor que el que tiene la ecuación, por lo tanto, se va actualizando. Luego se modifica el valor de height con el de la ecuación que se está analizando y se le suma 20. Por último se manda a pintar ecuación por ecuación para que se actualicen todos los datos de las mismas.

Después de haber ejecutado los casos de prueba de los métodos “pintar” y “selected” y de comparar los resultados, se determinó que todas las sentencias se han ejecutado por lo menos una vez. Se ejercitaron las estructuras internas de datos para asegurar su validez. El producto satisface los requisitos del usuario, tal y como se describe en las especificaciones. También el producto se comporta como se desea.

### 4.2.3 Diseño de las Pruebas Caja Negra

#### Caso de Uso Gestionar SED

Tabla 7 Variables utilizadas en el CU Gestionar SED.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	ecuación	ecuación algebraica o diferencial	no	Igualdad asociada a un miembro derecho y uno izquierdo.
2	operador	operador aritmético	no	Los operadores son+, -, *, /. Además de los paréntesis y raíz.

## CAPÍTULO 4. Implementación y Prueba

3	región	región editable	no	Región editable seleccionada en cualquier ecuación.
4	identificador	identificador	no	Números o constantes numéricas.
5	caracter	caracter	no	Son los puntos o comas.

**Tabla 8 Caso de Prueba 1 Adicionar nueva ecuación.**

Escenario	Descripción	V1	V2	V3	V4	V5	Respuesta del sistema	Flujo Central
EC 1.1: Se selecciona insertar ecuación algebraica.	El investigador va al menú Insert/Algebraic Equation y entra los datos de la nueva ecuación.	V	N/A	N/A	N/A	N/A	El sistema adiciona la nueva ecuación algebraica en el área de trabajo.	Seleccionar la opción nueva ecuación algebraica e introducir los datos necesarios correctamente.

**Tabla 9 Caso de Prueba 2 Insertar identificador o constante numérica.**

Escenario	Descripción	V1	V2	V3	V4	V5	Respuesta del sistema	Flujo Central
EC 2.1: Insertar identificador o constante numérica	Se inserta un identificador o constante numérica en la región editable seleccionada en el modelo matemático.	N/A	N/A	N/A	V	F	El sistema inserta el identificador o constante numérica.	Insertar un identificador o constante numérica en el modelo matemático.
EC 2.2: Insertar punto o coma	Se inserta un punto o coma en la región editable	N/A	N/A	N/A	V	V	El sistema mantiene la región	Insertar un punto o coma en el modelo

## CAPÍTULO 4. Implementación y Prueba

	seleccionada en el modelo matemático.	N/A	N/A	N/A	F	V	editable vacía.	matemático.
--	---------------------------------------	-----	-----	-----	---	---	-----------------	-------------

**Tabla 10 Caso de Prueba 3 Insertar operación matemática.**

Escenario	Descripción	V1	V2	V3	V4	V5	Respuesta del sistema	Flujo Central
EC 3.1: Insertar operaciones matemáticas.	Se selecciona alguna operación matemática y se inserta en el modelo seleccionado.	N/A	V	V	N/A	N/A	El sistema adiciona en el modelo la operación matemática seleccionada.	Seleccionar la operación que se desea incorporar al modelo e insertarla en el área de trabajo donde se encuentra el modelo.
EC 3.1: Cancelar la inserción de operaciones matemáticas.	Se selecciona alguna operación matemática y no se inserta en el modelo seleccionado.	N/A	V	F	N/A	N/A	El sistema no adiciona en el modelo ninguna operación matemática.	Seleccionar la operación que se desea incorporar al modelo y no insertarla en el área de trabajo donde se encuentra el modelo.

**Tabla 11 Caso de Prueba 4 Eliminar elemento.**

Escenario	Descripción	V1	V2	V3	V4	V5	Respuesta del sistema	Flujo Central
EC 4.1: Eliminar elemento.	Se elimina algún elemento del modelo	N/A	N/A	V	N/A	N/A	El sistema elimina el elemento seleccionado	Se selecciona algún elemento del modelo matemático que



## CAPÍTULO 4. Implementación y Prueba

	matemático.						del modelo.	se desee eliminar y se oprime la tecla "delete".
EC 4.2: Eliminar elemento.	No se elimina ningún elemento del modelo matemático.	N/A	N/A	F	N/A	N/A	El sistema no elimina el elemento seleccionado del modelo matemático.	Se selecciona algún elemento del modelo matemático que se desee eliminar y no se oprime la tecla "delete".

Caso de Prueba del Caso de Uso Importar ficheros. [Ver Anexo 7].

Caso de Prueba del Caso de Uso Exportar SED. [Ver Anexo 8].

Casos de Pruebas del Caso de Uso Modificar Fuente. [Ver Anexo 9].

### No conformidades detectadas

En la primera iteración fueron encontradas las no conformidades descritas a continuación:

**Tabla 12 No conformidades.**

Elemento	No	No Conformidad	Aspecto correspondiente	Etapas de detección	S.	N.S.
Implementación	1	Error al importar un modelo matemático.	El modelo que se deseaba importar, tenía ecuaciones inconclusas.	Implementación y prueba.	x	
Implementación	2	Error al importar modelo matemático.	El editor no reconoció un modelo matemático con una sola ecuación.	Implementación y prueba	x	
Interfaz	3	Error de internacionalización.	El menú "Insert" estaba escrito en	Implementación y Prueba		x

## CAPÍTULO 4. Implementación y Prueba

			español y el Editor se encuentra en inglés, al igual que BioSyS.			
--	--	--	--	--	--	--

Una vez solucionadas las no conformidades encontradas, se realizó una segunda iteración, en la cual, no se encontraron no conformidades, por lo que no se hizo necesario una tercera iteración de las pruebas de caja negra.

### 4.2.4 Método Experimental

El objetivo del método experimental aplicado, es importar varias clases de modelos matemáticos en el Editor 1.2 y en el Editor 2.0, para comprobar el tiempo que demora en responder cada software. Una vez que se tengan esos tiempos se efectuará una comparación para ver cual es más eficaz y así se realizará una verificación a ver si los resultados de la prueba son los esperados.

Para la realización del método experimental se precisó de 3 clases de modelos matemáticos, con las características siguientes: [Ver Anexo 7]

- clase 1. Modelo compuesto por 5 ecuaciones.
- clase 2. Modelo compuesto por 10 ecuaciones.
- clase 3. Modelo compuesto por 20 ecuaciones.

Una vez realizado el proceso de importación de estos modelos matemáticos se obtuvieron los siguientes tiempos de respuesta del sistema. La unidad de medida de tiempo está en segundos. En la tabla 13, se observa cómo a medida que va aumentando la cantidad de ecuaciones en los modelos matemáticos más aumenta la demora en el tiempo de respuesta.

**Tabla 13 Tiempo de respuesta**

Clases	Editor 1.2	Editor 2.0	Tiempo de diferencia
1	9 s	0.5 s	8.5 s
2	18 s	0.8 s	17.2 s
3	35 s	1 s	34 s

Al realizar un análisis de los resultados arrojados por el experimento se determinó que el Editor de Ecuaciones 2.0 cumplió con las expectativas esperadas, se disminuyó el tiempo de respuesta durante el proceso de edición de modelos matemáticos.

## **CAPÍTULO 4. Implementación y Prueba**

---

### **4.3 Conclusiones del capítulo**

El Editor de Ecuaciones se logró implementar exitosamente. Se integró a BioSyS y a la Plataforma de Servicios Bioinformáticos satisfactoriamente. Una vez realizadas las pruebas de caja negra al sistema, se detectaron 3 no conformidades, las mismas quedaron registradas y solucionadas para mejorar el funcionamiento del software. En las pruebas de caja blanca después de confeccionar los casos de prueba y ejecutarlos, se verificó que se ejecutaran las sentencias al menos una vez. Una vez aplicado el método experimental se pudo confirmar que el Editor de Ecuaciones, versión 2.0, disminuye el tiempo de respuesta al editar modelos matemáticos.

### CONCLUSIONES GENERALES

Se desarrolló la versión 2.0 del Editor de Ecuaciones del software BioSyS y se integró al Portal de Servicios Bioinformáticos.

Se logró disminuir el tiempo de respuesta del Editor durante la edición de modelos matemáticos.

Se validó la aplicación a través de pruebas de caja negra y caja blanca, detectándose solamente 3 no conformidades, las mismas fueron solucionadas, lo cual demuestra el correcto funcionamiento del software.

Se diseñaron e implementaron las funcionalidades necesarias en el Editor de Ecuaciones, para proporcionar un mejor servicio al simular y analizar los modelos matemáticos.

### RECOMENDACIONES

- Continuar incrementando la cantidad de operaciones matemáticas del editor, con el fin de ampliar las funcionalidades del software.
- Incorporar en el Editor la funcionalidad de asignar dimensión así como verificar la homogeneidad dimensional.
- Agregar al Editor las funcionalidades de importar ficheros en MathML Content y Código Interno.

### REFERENCIAS BIBLIOGRÁFICAS

1. *Biología de Sistemas y su aplicación en Biología Celular*. **Maryorga, Luis S.** Cuyo, Argentina : s.n., 2008, Vols. Vol 4 - N<sup>o</sup> 1-2008.
2. **López, Marta, Ruiz Romero, Gema y Vega, Miguel.** *Biología de Sistemas*. Madrid, España : Spainfo, S.A., 2007. 84-609-9762-6.
3. **Arias Naranjo, Gilberto.** *Editor de Ecuaciones para la Plataforma de Simulación de Sistemas Biológicos*. La Habana, Cuba : s.n., 2008.
4. **González Iglesias, Carlos y Moreno Lemus, Edel.** *Software para el Estudio de Sistemas Biológicos versión 2.0*. La Habana, Cuba : s.n., 2010.
5. **Reinoso Cabrera, Carlos Alberto y Scantlebury Sorí, Yanier.** *Integración de los módulos del software alasBioSyS a la Plataforma de Servicios Bioinformáticos*. La Habana, Cuba : s.n., 2012.
6. **Campus Universitario del País Vasco.** El primer applet. [En línea] 12 de 7 de 2010. [Citado el: 9 de 12 de 2012.] <http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/intro/primero.htm>.
7. **Shum, Joseph, y otros.** *Portal 4 - Portlet development guide*. 2007.
8. *Spring Portlet MVC*. **Lewis, Jhonm A.** Atlanta : s.n., 2006.
9. **Minter, Dave.** *Beginning Spring 2. From Novice to Professional*. United States of America : s.n., 2008. 1-4302-0493-1.
10. Metodologías de desarrollo de software. [En línea] Instituto de Ingeniería del Software de la Universidad Carnegie Mellon de Pittsburgh, 30 de 12 de 2006. [Citado el: 16 de 01 de 2013.] <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
11. **Gurley, Greg y Oster, Nate.** ATSC. *Partners Delivering Results*. [En línea] 2008. [Citado el: 13 de 12 de 2012.] <http://www.atsc.com>.
12. **Communitic International** . Kioskea. [En línea] 2009. [Citado el: 13 de 12 de 2012.] <http://es.kioskea.net/contents/langages/langages.php3>.
13. **Belmonte Fernández, Oscar.** *Introducción al lenguaje de programación Java*. 2004.
14. Enterprise Open Source For Life. [En línea] [Citado el: 9 de 12 de 2012.] <http://www.liferay.com>.
15. Featured Eclipse Project. [En línea] by IBM, 11 de 2004. [Citado el: 14 de 12 de 2012.] <http://www.eclipse.org/>.
16. Agile Modeling. [En línea] Ambyssoft Inc., 2001. [Citado el: 14 de 12 de 2012.] <http://www.agilemodeling.com/>.
17. UML,BPMN and database Tool for Software Development. [En línea] 2004. [Citado el: 14 de 12 de 2012.] <http://www.visual-paradigm.com>.
18. Document Information Retrieval Systems. *Docirs*. [En línea] [Citado el: 14 de 12 de 2012.] <http://www.docirs.cl>.
19. **OMG.** [En línea] 2005. [Citado el: 14 de 12 de 2012.] [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
20. Estudios Lingüística del Español. *ELiEs*. [En línea] Fernández-Valmayor, Navarro, Fernández-Manjón y J. Sierra, 2006. [Citado el: 14 de 12 de 2012.] <http://elies.rediris.es/elies24/fernandezvalmayor.htm>.

## Referencias Bibliográficas.

---

21. Extensible Markup Language (XML). W3C. [En línea] 15 de 4 de 2004. [Citado el: 14 de 12 de 2012.] <http://www.w3.org/TR/2004/REC-xml11-20040204>.
22. **Guiarte Multimedia S.L.** Desarrollo Web. [En línea] 2010. [Citado el: 10 de 12 de 2012.] <http://www.desarrolloweb.com/articulos/449.php>.
23. **Martínez, Carolina.** *MODELO CONCEPTUAL/MODELO DE DOMINIO*.
24. *Sistema para auditar el cumplimiento de cmmi -sw nivel 2.* **Vargas, César y Biagioli, German.**
25. **Márquez Gómez, José Jorge.** *ARQUITECTURA MVC, Visión General*.
26. msdn. [En línea] [Citado el: 06 de 05 de 2013.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
27. **Bustos Thames, Juan Pablo.** SlideShare. [En línea] 2003. [Citado el: 22 de 03 de 2013.] <http://www.slideshare.net/jpbthames/patrones-para-asignar-responsabilidades-grasp>.
28. **Universidad Rey Juan Carlos.** *Conceptos y principios del diseño*.
29. **Marca Huallpara, Hugo Michael y Quisbert Limachi, Nancy Susana.** Universidad Salesiana. [En línea] 2013. [Citado el: 1 de 04 de 2013.] [virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc](http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc).
30. SlideShare. [En línea] [Citado el: 07 de 05 de 2013.] <http://www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue>.
31. Scrum Manager. [En línea] 23 de 01 de 2013. [Citado el: 12 de 4 de 2013.] [http://www.scrummanager.net/bok/index.php?title=Pruebas\\_de\\_sistema](http://www.scrummanager.net/bok/index.php?title=Pruebas_de_sistema).
32. **Mañas, José A.** Prueba de Programas. [En línea] 16 de 03 de 1994. [Citado el: 17 de 04 de 2013.] <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
33. **Hernández León, Rolando Alfredo y Coello González, Sayda.** *El proceso de investigación científica*. Ciudad de la Habana : Editorial Universitaria, 2011. 978-959-16-1307-3.

### BIBLIOGRAFÍA

1. *Biología de Sistemas y su aplicación en Biología Celular*. **Maryorga, Luis S.** Cuyo, Argentina : s.n., 2008, Vols. Vol 4 - N<sup>a</sup> 1-2008.
2. **López, Marta, Ruiz Romero, Gema y Vega, Miguel.** *Biología de Sistemas*. Madrid, España : Spainfo, S.A., 2007. 84-609-9762-6.
3. **Arias Naranjo, Gilberto.** *Editor de Ecuaciones para la Plataforma de Simulación de Sistemas Biológicos*. La Habana, Cuba : s.n., 2008.
4. **González Iglesias, Carlos y Moreno Lemus, Edel.** *Software para el Estudio de Sistemas Biológicos versión 2.0*. La Habana, Cuba : s.n., 2010.
5. **Reinoso Cabrera, Carlos Alberto y Scantlebury Sorí, Yanier.** *Integración de los módulos del software alasBioSyS a la Plataforma de Servicios Bioinformáticos*. La Habana, Cuba : s.n., 2012.
6. **Campus Universitario del País Vasco.** El primer applet. [En línea] 12 de 7 de 2010. [Citado el: 9 de 12 de 2012.] <http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/intro/primero.htm>.
7. **Shum, Joseph, y otros.** *Portal 4 - Portlet development guide*. 2007.
8. *Spring Portlet MVC*. **Lewis, Jhonm A.** Atlanta : s.n., 2006.
9. **Minter, Dave.** *Beginning Spring 2. From Novice to Professional*. United States of America : s.n., 2008. 1-4302-0493-1.
10. Metodologías de desarrollo de software. [En línea] Instituto de Ingeniería del Software de la Universidad Carnegie Mellon de Pittsburgh, 30 de 12 de 2006. [Citado el: 16 de 01 de 2013.] <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
11. **Gurley, Greg y Oster, Nate.** ATSC. *Partners Delivering Results*. [En línea] 2008. [Citado el: 13 de 12 de 2012.] <http://www.atsc.com>.
12. **Communitic International** . Kioskea. [En línea] 2009. [Citado el: 13 de 12 de 2012.] <http://es.kioskea.net/contents/langages/langages.php3>.
13. **Belmonte Fernández, Oscar.** *Introducción al lenguaje de programación Java*. 2004.
14. Enterprise Open Source For Life. [En línea] [Citado el: 9 de 12 de 2012.] <http://www.liferay.com>.
15. Featured Eclipse Project. [En línea] by IBM, 11 de 2004. [Citado el: 14 de 12 de 2012.] <http://www.eclipse.org/>.
16. Agile Modeling. [En línea] Ambyssoft Inc., 2001. [Citado el: 14 de 12 de 2012.] <http://www.agilemodeling.com/>.
17. UML,BPMN and database Tool for Software Development. [En línea] 2004. [Citado el: 14 de 12 de 2012.] <http://www.visual-paradigm.com>.
18. Document Information Retrieval Systems. *Docirs*. [En línea] [Citado el: 14 de 12 de 2012.] <http://www.docirs.cl>.
19. OMG. [En línea] 2005. [Citado el: 14 de 12 de 2012.] [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
20. Estudios Lingüística del Español. *ELíEs*. [En línea] Fernández-Valmayor, Navarro, Fernández-Manjón y J. Sierra, 2006. [Citado el: 14 de 12 de 2012.] <http://elies.rediris.es/elies24/fernandezvalmayor.htm>.



21. Extensible Markup Language (XML). W3C. [En línea] 15 de 4 de 2004. [Citado el: 14 de 12 de 2012.] <http://www.w3.org/TR/2004/REC-xml11-20040204>.
22. **Guiarte Multimedia S.L.** Desarrollo Web. [En línea] 2010. [Citado el: 10 de 12 de 2012.] <http://www.desarrolloweb.com/articulos/449.php>.
23. **Martinez, Carolina.** *MODELO CONCEPTUAL/MODELO DE DOMINIO*.
24. *Sistema para auditar el cumplimiento de cmmi -sw nivel 2.* **Vargas, César y Biagioli, German.**
25. **Márquez Gómez, José Jorge.** *ARQUITECTURA MVC, Visión General*.
26. msdn. [En línea] [Citado el: 06 de 05 de 2013.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
27. **Bustos Thames, Juan Pablo.** SlideShare. [En línea] 2003. [Citado el: 22 de 03 de 2013.] <http://www.slideshare.net/jpbthames/patrones-para-asignar-responsabilidades-grasp>.
28. **Universidad Rey Juan Carlos.** *Conceptos y principios del diseño*.
29. **Marca Huallpara, Hugo Michael y Quisbert Limachi, Nancy Susana.** Universidad Salesiana. [En línea] 2013. [Citado el: 1 de 04 de 2013.] [virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc](http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc).
30. SlideShare. [En línea] [Citado el: 07 de 05 de 2013.] <http://www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue>.
31. Scrum Manager. [En línea] 23 de 01 de 2013. [Citado el: 12 de 4 de 2013.] [http://www.scrummanager.net/bok/index.php?title=Pruebas\\_de\\_sistema](http://www.scrummanager.net/bok/index.php?title=Pruebas_de_sistema).
32. **Mañas, José A.** Prueba de Programas. [En línea] 16 de 03 de 1994. [Citado el: 17 de 04 de 2013.] <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
33. **Hernández León, Rolando Alfredo y Coello González, Sayda.** *El proceso de investigación científica*. Ciudad de la Habana : Editorial Universitaria, 2011. 978-959-16-1307-3.
34. **Molpeceres, Alberto.** Procesos de desarrollo: RUP, XP y FDD. *Licencia de Documentación de javaHispano*. [En línea] 15 de 12 de 2002. [Citado el: 10 de 12 de 2012.] <http://www.javahispano.org/licencias>.
35. *Programacion eXtrema y Software Libre.* **Robles, Gregorio y Ferrer, Jorge.** s.l. : spalinux, 2002.
36. **Casas, Eduardo.** Introducción a MathML. [En línea] 11 de 01 de 2013. [Citado el: 11 de 04 de 2013.] <http://www.eduardocasas.com/blog/11-01-2013/introduccion-a-mathml>.
37. The WebEQ Equation Editor. WebEQ 2.5. [En línea] Design Science, Inc. [Citado el: 24 de 11 de 2012.] <http://web.mit.edu/webeq/currenthome/docs/editor/welcome.html>.

## ANEXOS

### Anexo 1. Diagrama de Secuencia del Caso de Uso Exportar SED.

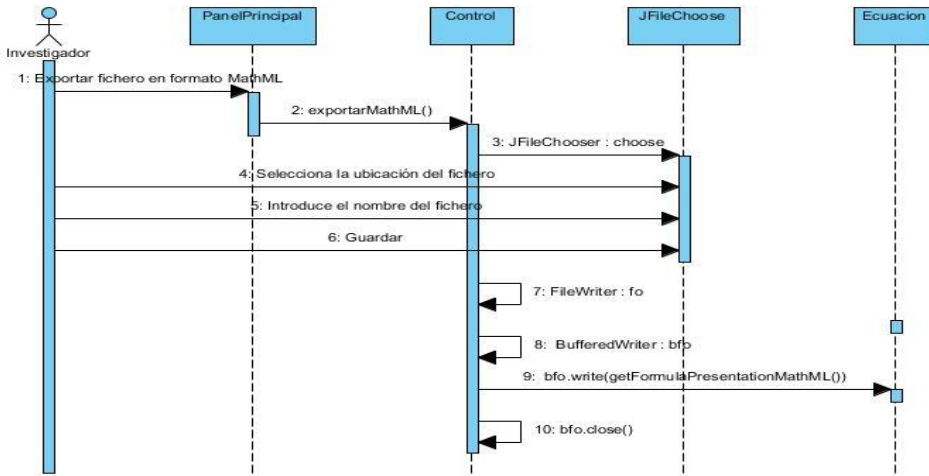


Figura 29 Diagrama de Secuencia del CU Exportar SED.

### Anexo 2. Diagrama de Secuencia del Caso de Uso Importar ficheros.

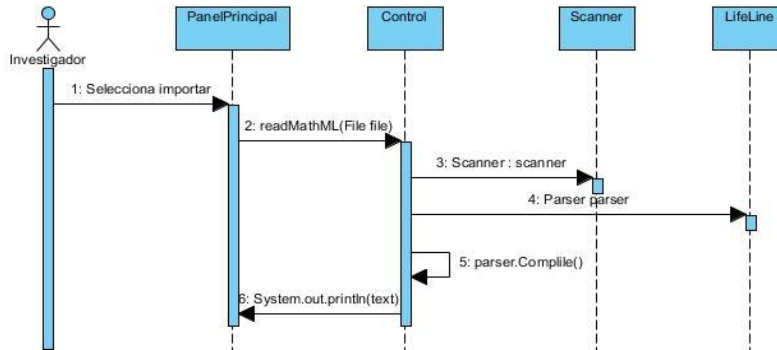


Figura 30 Diagrama de Secuencia del CU Importar ficheros.

### Anexo 3. Diagrama de Secuencia del Caso de Uso Modificar fuente.

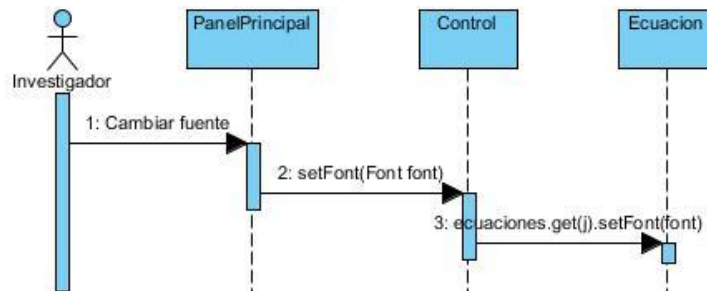


Figura 31 Diagrama de Secuencia del CU Modificar fuente.

**Anexo 4. Caso de Prueba del Caso de Uso Importar ficheros**

**Tabla 14 Variable utilizada en el CU Importar ficheros.**

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	archivo	archivo	si	Archivos con extensión: .xml.

**Tabla 15 Caso de Prueba 5 Importar ficheros.**

Escenario	Descripción	V1	Respuesta del sistema	Flujo Central
EC 1.1: Se selecciona el fichero correctamente.	El sistema muestra una ventana donde el investigador busca el archivo deseado con extensión .xml.	V	El sistema carga el modelo matemático del archivo seleccionado y lo muestra en el Editor de Ecuaciones.	Selecciona una opción para importar en el formato establecido.
EC 1.2: Se selecciona el fichero incorrectamente.	El sistema muestra una ventana donde el investigador busca el archivo deseado.	F	El sistema muestra un mensaje de error con el siguiente texto <i>"Cont is not allowed in prolog."</i>	Se selecciona el archivo con la extensión incorrecta y el sistema muestra un mensaje de error.

**Anexo 5. Caso de Prueba del Caso de Uso Exportar SED**

**Tabla 16 Variable utilizada en el CU Exportar SED.**

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	archivo	archivo	si	Archivos con extensión: .xml, .java, .m, .png.

**Tabla 17 Caso de Prueba 6 Exportar fichero.**

Escenario	Descripción	V1	Respuesta del sistema	Flujo Central
EC 1.1: Se exporta el fichero.	El sistema muestra una ventana donde el investigador busca donde se va a guardar el archivo en el formato deseado y selecciona la	V	El sistema crea el archivo que contiene el sistema de ecuaciones con el formato deseado en la ruta y con el nombre especificados, finalizando así el caso de	Selecciona una opción para exportar en uno de los formatos establecidos.

	opción "Aceptar".		uso.
--	-------------------	--	------

### Anexo 6. Casos de Pruebas del Caso de Uso Modificar fuente.

**Tabla 18 Caso de Prueba 7 Font.**

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1.1: Modificar el tipo de fuente.	Se selecciona el tipo de fuente que se desea en el menú desplegable.	El sistema cambia el tipo de fuente de todas las ecuaciones en el área de trabajo.	Se da clic en el tipo de fuente y rápidamente se desplaza un menú, con las opciones y se selecciona un tipo en particular.
EC 1.2: No modificar el tipo de fuente.	No se selecciona el tipo de fuente que se desea en el menú desplegable.	El sistema no cambia el tipo de fuente de todas las ecuaciones en el área de trabajo.	Se da clic en el tipo de fuente y rápidamente se desplaza un menú, con las opciones de tipo fuentes y no se selecciona ningún tipo en particular.

**Tabla 19 Caso de Prueba 8 Size.**

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1.1: Modificar el tamaño de fuente.	Se selecciona el tamaño de fuente que se desea en el menú desplegable.	El sistema cambia el tamaño de fuente de todas las ecuaciones en el área de trabajo.	Se da clic en el tamaño de fuente y rápidamente se desplaza un menú, con las opciones de fuentes y se selecciona un tipo en particular.
EC 1.2: No modificar el tamaño de fuente.	No se selecciona el tamaño de fuente que se desea en el menú desplegable.	El sistema no cambia el tamaño de fuente de todas las ecuaciones en el área de trabajo.	Se da clic en el tamaño de fuente y rápidamente se desplaza un menú, con las opciones tamaño de fuentes y no se selecciona ningún tipo en particular.

Tabla 20 Caso de Prueba 9 Foreground.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1.1: Modificar el color de fuente.	El sistema muestra una ventana donde el investigador selecciona el color que desea para la fuente de las ecuaciones y selecciona la opción "Aceptar".	El sistema cambia el color de todas las ecuaciones en el área de trabajo.	Se da clic en la F de la barra de herramientas, rápidamente se muestra una ventana, con las opciones de colores y se selecciona un tipo en particular. Luego se oprime "Aceptar".
EC 1.2: No modificar el color de fuente.	El sistema muestra una ventana donde el investigador selecciona el color que desea para la fuente de las ecuaciones y selecciona la opción "Cancelar".	El sistema no cambia el color de todas las ecuaciones en el área de trabajo.	Se da clic en la F de la barra de herramientas, rápidamente se muestra una ventana, con las opciones de colores y se selecciona un tipo en particular. Luego se oprime "Cancelar".

Tabla 21 Caso de Prueba 10 Background.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1.1: Modificar el color de fondo de la fuente.	El sistema muestra una ventana donde el investigador selecciona el color que desea para el fondo de la fuente de las ecuaciones y selecciona la opción "Aceptar".	El sistema cambia el color de fondo en el área de trabajo.	Se da clic en la B de la barra de herramientas, rápidamente se muestra una ventana, con las opciones de colores y se selecciona un tipo en particular. Luego se oprime "Aceptar".
EC 1.2: No modificar el color de fondo de la fuente.	El sistema muestra una ventana donde el investigador selecciona el color que desea para el fondo de la fuente de las ecuaciones y selecciona la	El sistema no cambia el color de fondo en el área de trabajo.	Se da clic en la B de la barra de herramientas, rápidamente se muestra una ventana, con las opciones de colores y se selecciona un tipo en particular. Luego se oprime

	opción "Cancelar".	"Cancelar".
--	--------------------	-------------

**Anexo 7. Clases de ecuaciones.**

**Clase 1**

$$e = en + ea$$

$$ea_s = \frac{ea}{r + e} \frac{1}{2}$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

**Clase 2**

$$e = en + ea$$

$$ea_s = \frac{ea}{r + e} \frac{1}{2}$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec + rir m + rira ra + rirc rc$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec + rir m + rira ra + rirc rc$$

**Clase 3**

$$e = en + ea$$

$$ea_s = \frac{ea}{r + e} \frac{1}{2}$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec + rir m + rira ra + rirc rc$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec + rir m + rira ra + rirc rc$$

$$e = en + ea$$

$$ea_s = \frac{ea}{r + e} \frac{1}{2}$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec + rir m + rira ra + rirc rc$$

$$m_s = \frac{m}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec$$

$$en = \frac{en}{r + e} \frac{1}{2 Ke} [Ke s A + Ke(e + r)]$$

$$rit = rien en + riew ea + riec ec + rir m + rira ra + rirc rc$$

**Figura 32 Clases de ecuaciones del método experimental.**

### GLOSARIO

**API:** Interfaces de Programación de Aplicaciones.

**Bifurcación:** División en dos ramales.

**Bioinformática:** Es la aplicación de los ordenadores y los métodos informáticos en el análisis de datos experimentales y simulación de los sistemas biológicos.

**Bytecodes:** Es el código independiente del hardware generado por el compilador Java™ y ejecutado por su intérprete.

**Clúster:** Son todos los entes que trabajan en conjunto para lograr un fin en diferentes ámbitos.

**CIM:** Centro de Inmunología Molecular.

**Framework:** Marco de aplicación o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones.

**OMG:** Grupo de Gestión de Objetos.

**Open source:** Calificación que se le atribuye a los programas informáticos que permiten el acceso a su código de programación.

**Plug-in:** Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

**Telnet:** Nombre de un protocolo de red que se utiliza para acceder a una computadora y manejarla de forma remota.

**UCI:** Universidad de las Ciencias Informáticas.