

**Universidad de las Ciencias Informáticas**

**Facultad 6**



**Título: Software para la Simulación de Sistemas  
Biológicos: Módulo de Simulación y Análisis**

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

**Autor(es):** Yanet Alonso Delgado

Yunet González Mulet

**Tutor(es):** Lic. Noel Moreno Lemus

Dr. Kalet León Monzón

**Co-tutor(es):** Jorge L. Vázquez

Julio, 2007

*“Confirmar es crear. Lo que hace crecer el mundo no es el descubrir cómo está hecho sino el esfuerzo que cada uno hace para descubrirlo.”*

*José Martí.*

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

## DATOS DE CONTACTO

### Tutores:

Dr. Kalet León Monzón.  
Centro de Inmunología Molecular, Habana, Cuba.  
Email: [kalet@ict.cim.sld.cu](mailto:kalet@ict.cim.sld.cu)

Lic. Noel Moreno Lemus.  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: [noel@uci.cu](mailto:noel@uci.cu)

### Co-Tutor:

Lic. Jorge Luis Vázquez González.  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: [jlvazquez@uci.cu](mailto:jlvazquez@uci.cu)

## AGRADECIMIENTOS

Agradecer a todos aquellos que tuvieron que ver de una forma u otra con nuestra formación, a nuestros padres por su apoyo, comprensión y soporte durante todos estos años, a nuestros amigos por habernos hecho disfrutar estos maravillosos cinco años al máximo y por ser partícipes de nuestros logros, desconciertos, malos y buenos momentos en general. A nuestros tutores, en especial a Noel, por su paciencia, esmero y ayuda. A Edel por su colaboración y apoyo. Un agradecimiento infinito a nuestro Comandante por habernos dado la oportunidad de formarnos como ingenieros en una Universidad de nuevo tipo, que ha marcado nuestro pensamiento revolucionario y profesional.

## DEDICATORIA

*A mis padres por su cariño y apoyo, por su paciencia y sobre todo por su dedicación.*

*A mis hermanos que sin duda son el motivo fundamental de mi superación y mi mejor regalo.*

*A mi abuela, mi tía y mis primos cuya ayuda fue fundamental.*

*A mi esposo por su dedicación y amor, por haber sido la guía que necesitaba en el momento preciso, por su eterna comprensión, por ser quien es, por alegrarme mis días y por estar ahí cuando más lo he necesitado.*

*A mis amigos que han sido siempre mi mejor compañía.*

*Yanet Alonso Delgado.*

*A Caridad y a Roberto ... donde estén.*

*A mis padres Elvia y Roberto por quererme siempre y por caminar junto a mi sin importarles el sacrificio.*

*A mi familia que siempre me ha apoyado.*

*A Alexis por ser más que un amigo, un hermano.*

*A mi gran amiga Reina porque juntas compartimos todos los momentos que pudimos.*

*A Yania, Annabell, Yadira y Yeni por escucharme y ser mis más sinceras amigas.*

*A mis demás amigos, a todos, pues con ellos compartí inolvidables momentos y estuvieron cuando necesité su presencia.*

*Yunet González Mulet.*

## RESUMEN

El presente trabajo tiene como objetivo fundamental implementar un software que facilite el estudio de sistemas biológicos que puedan ser descritos mediante sistemas de ecuaciones diferenciales. Dicha aplicación está concebida para que sea capaz de realizar simulaciones distribuidas usando tecnología Grid, minimizando así el tiempo de obtención de los resultados de las mismas; almacene dichos resultados en una base de datos diseñada al efecto y permita realizar estudios posteriores, incorporando varios tipos de análisis como son, el análisis a través de gráficas de dinámicas de población, algoritmos de clustering y reglas definidas por el usuario.

## PALABRAS CLAVE

- Análisis
- Biológicos
- Bases de Datos
- Modelos
- Sistemas
- Simulaciones
- Distribuidas
- Software

## TABLA DE CONTENIDOS

AGRADECIMIENTOS .....	I
DEDICATORIA .....	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	6
<b>1.1 Biología de Sistemas. Aplicaciones</b> .....	6
1.1.1 Biología de Sistemas (BS).....	7
1.1.2 Aplicaciones de la Biología de Sistemas.....	7
<b>1.2 Modelo matemático. Modelación de sistemas biológicos</b> .....	8
<b>1.3 Métodos de resolución de modelos matemáticos</b> .....	9
1.3.1 Métodos de Runge-Kutta.....	9
1.3.2 Métodos de Adams .....	9
<b>1.4 Simulación. Simulación biológica</b> .....	10
<b>1.5 Aplicaciones computacionales vinculadas al estudio de la Biología de Sistemas</b> .....	12
<b>1.6 Importancia del desarrollo de una nueva aplicación</b> .....	15
1.6.1 Ventajas de Computación Grid.....	16
1.6.2 Algunas aplicaciones de la tecnología Grid .....	16
<b>1.7 Tipos de análisis incluidos en la aplicación desarrollada</b> .....	17
1.7.1 Gráficas de dinámicas de población .....	17
1.7.2 Algoritmos de Clustering.....	17
1.7.3 Reglas definidas por el usuario.....	23
<b>1.8 Herramientas y metodologías utilizadas en la aplicación</b> .....	24
1.8.1 Metodologías de desarrollo de software .....	24
1.8.2 Herramientas CASE .....	26
1.8.3 Lenguaje de Modelado .....	27
1.8.4 Lenguajes de programación.....	27
1.8.5 Herramientas de desarrollo.....	28
1.8.6 Gestor de Base de datos.....	29
1.8.7 Software Weka .....	29
<b>1.9 Conclusiones</b> .....	30
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	31
<b>2.1 Modelo del negocio actual</b> .....	31
<b>2.2 Reglas del negocio</b> .....	31
<b>2.3 Actores del negocio</b> .....	32
<b>2.4 Diagrama de casos de uso del negocio</b> .....	33

<b>2.5 Trabajadores del negocio</b> .....	33
<b>2.6 Descripciones de Casos de uso del negocio</b> .....	34
<b>2.7 Diagramas de Actividades</b> .....	37
<b>2.8 Modelo de objetos</b> .....	39
<b>2.9 Actores del sistema a automatizar</b> .....	39
<b>2.10 Requisitos Funcionales</b> .....	40
<b>2.11 Diagrama de casos de uso del sistema a automatizar</b> .....	41
<b>2.12 Requisitos no funcionales</b> .....	42
<b>2.13 Casos de uso del sistema</b> .....	44
<b>2.14 Descripción de los casos de uso del sistema</b> .....	45
<b>2.15 Conclusiones</b> .....	47
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA</b> .....	48
<b>3.1 Diagrama de clases del Análisis</b> .....	48
<b>3.2 Estilo arquitectónico utilizado</b> .....	48
<b>3.3 Patrones de diseño utilizados</b> .....	49
3.3.1 ¿Qué son los Patrones de Diseño? .....	49
3.3.2 Principales Patrones de diseño utilizados .....	49
<b>3.4 Diagrama de clases del Diseño</b> .....	52
<b>3.5 Principios de diseño</b> .....	52
3.5.1 Interfaz de usuario .....	52
3.5.2 Tratamiento de errores .....	52
<b>3.6 Modelo de clases persistentes</b> .....	53
<b>3.7 Diseño de la base de datos</b> .....	54
<b>3.8 Modelo físico de datos</b> .....	55
<b>3.9 Diagrama de Despliegue</b> .....	55
<b>3.10 Conclusiones</b> .....	57
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA</b> .....	58
<b>4.1 Diagrama de Componentes</b> .....	58
<b>4.2 Casos de prueba de caja negra</b> .....	60
<b>CONCLUSIONES</b> .....	77
<b>RECOMENDACIONES</b> .....	78
<b>BIBLIOGRAFÍA</b> .....	82
<b>ANEXOS</b> .....	87
<b>GLOSARIO</b> .....	116

## INTRODUCCIÓN

En los últimos años los estudios correspondientes a la Biotecnología han alcanzado un avance significativo. Varias son las ramas que esta ciencia moderna emplea para su fundamento y dentro de ellas una de las más jóvenes y que comienza a ganar terreno dentro de los científicos que a nivel mundial se dedican al estudio de la misma, es la Biología de Sistemas.

Desde hace años el método utilizado por los científicos para estudiar un sistema biológico ha sido el método reduccionista, es decir, la descomposición del sistema en sus partes constituyentes, las cuales por separado, son mucho más sencillas de entender y modelar.

Esta estrategia se ha mostrado como una poderosa herramienta capaz de describir los sistemas físicos, porque proporciona un mapa completo de las piezas fundamentales, y de las funciones que las mismas van realizando, pero no permite la comprensión de las interacciones entre las partes, o sea conocer el funcionamiento del sistema como un todo. Este ha sido el marco propicio para el surgimiento de la Biología de Sistemas, que hoy en día ha resultado ser uno de los campos más activos, aprovechando no sólo los avances en el conocimiento de las partes fundamentales, sino también la posibilidad de realizar cálculos mucho más complejos y elaborar modelos informáticos gracias a los avances en las Ciencias de la Computación.

La Biología de Sistemas no sólo necesita de un mejor conocimiento de la Bioquímica, o de rápidas y potentes herramientas de cálculo masivo. La investigación implica una metodología científica susceptible, aplicada no sólo por personas, sino también por las máquinas utilizadas. De esta manera, el desarrollo en modelaciones, teoría de la decisión y formas de realizar análisis, son fundamentales a la hora de investigar.

Esta nueva ciencia permite entender la manera en que la transcripción de diferentes proteínas por los mismos genes está regulada por otros genes, y a la vez éstas regulan otros nuevos, estableciendo redes que, cuando se conozcan y almacenen en bases de datos, permitirán desarrollar sistemas que descubran nuevas redes, incluso de forma automatizada, además de comprender mejor los mecanismos de aparición de enfermedades, o el desarrollo de nuevos fármacos.

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, pasando por cadenas de Markov, Redes Booleanas y Redes Bayesianas.

La complejidad que entrañan las investigaciones en esta área ha hecho necesario el uso de computadoras y el desarrollo de múltiples aplicaciones. Hasta hace poco tiempo este desarrollo no llegaba a todos los investigadores, pues no existían software fáciles de usar, que permitieran modelar y analizar sistemas biológicos.

En las últimas décadas se han desarrollado un gran número de aplicaciones computacionales que pretenden facilitar la modelación y el análisis de sistemas biológicos. La mayor desventaja que existe en estas aplicaciones es que se han centrado mucho en la modelación y la simulación, pero han dejado en manos de los usuarios el análisis de los resultados, tarea esta, tan o más complicada que las anteriores.

Nuestro país ha estado siempre consciente de la necesidad de desarrollar la ciencia nacional y se ha enfrascado en propiciar el marco para fomentar el interés por la misma. En la década del 80 comienzan a crearse centros biotecnológicos donde se destaca el Centro de Investigaciones Biológicas, el Centro de Ingeniería Genética y Biotecnología (CIGB), Centro de Producción de Animales de Laboratorio (CENPALAB), Centro Nacional de Biopreparados (BIOCEN), Centro de Inmunoensayo y el Centro de Inmunología Molecular (CIM).

Como se pudo conocer en el mundo se han desarrollado aplicaciones que se encargan de realizar simulaciones distribuidas de los sistemas biológicos que son modelados mediante SED, almacenan dichas simulaciones en una base de datos y en algunos casos incluyen una o dos herramientas de análisis. Por lo que se puede plantear como **problema científico**: ¿Cómo informatizar el proceso de simulación y análisis de los sistemas biológicos?

En algunos de estos centros se estudia la simulación de los sistemas biológicos, pero no tienen acceso a aplicaciones que automaticen este proceso, ya sea por su alto costo en el mercado internacional, por su no comercialización o porque no satisfacen del todo, las necesidades de los investigadores de nuestro país. Este es el caso del Centro de Inmunología Molecular, donde este proceso no se realiza de manera automatizada.

Por lo que el **objeto de estudio** planteado es: El proceso de simulación y análisis de sistemas biológicos.

El **campo de acción**: Las simulaciones distribuidas y el análisis de sistemas biológicos que puedan ser modelados mediante SED.

La mayoría de las aplicaciones existentes presentan algunas desventajas para la realización de estudios de sistemas biológicos. Muchos de ellos fueron diseñados sólo para la modelación y simulación de estos sistemas y muy pocos incorporan técnicas de análisis con las cuales se pueda llegar a tener un criterio de decisión. En el caso de los que incluyen alguna técnica de análisis están dirigidos a un estudio en específico y no son capaces de generalizar el mismo.

De ahí que el **objetivo general** de esta investigación sea desarrollar una aplicación que permita realizar simulaciones distribuidas de sistemas biológicos que puedan ser modelados mediante Sistemas de Ecuaciones Diferenciales, almacenar los resultados y realizar meta-análisis sobre estos datos.

**Los objetivos específicos** que se derivan son los siguientes:

- Realizar análisis y diseño de una aplicación que simule distribuidamente y analice los resultados de la simulación de sistemas biológicos.
- Diseñar e implementar una Base de Datos para almacenar toda la información referente a un sistema biológico.
- Implementar una aplicación que permita simular sistemas biológicos distribuidamente y analizar los resultados de las simulaciones almacenadas en una BD.
- Realizar pruebas de caja negra que validen el correcto funcionamiento de la aplicación.

Para resolver el problema planteado y alcanzar el cumplimiento de los objetivos se trazaron las siguientes **tareas**:

- Revisión bibliográfica sobre la simulación de sistemas biológicos.
- Estudio de las aplicaciones computacionales existentes para la simulación distribuida y análisis de los resultados de la simulación de sistemas biológicos.
- Modelación del negocio.
- Análisis y diseño de una aplicación para la simulación distribuida y el análisis de sistemas biológicos.
- Diseño de una base de datos que almacene toda la información referente a un sistema biológico.

- Diseño de los prototipos de interfaz de una aplicación desktop que permita realizar simulaciones haciendo uso del cálculo distribuido y permita realizar análisis de los resultados.
- Programación de funciones en MatLab que permitan realizar cálculos complejos para la simulación del modelo matemático.
- Estudio del software Weka para utilizar los algoritmos de clustering que incluye.
- Implementación de una aplicación software para la simulación distribuida de sistemas biológicos y análisis de los resultados de las simulaciones.

#### Aportes prácticos esperados del trabajo

Con el desarrollo de esta aplicación se le facilitará a los usuarios el proceso de realizar simulaciones de sistemas biológicos, permitiéndole realizar simulaciones distribuidas de los mismos. Brindará la posibilidad de manipular toda la información referente a los sistemas biológicos. Tendrá almacenada, en una base de datos, información correspondiente a los sistemas biológicos estudiados con anterioridad para poder modificarlos y compararlos con otros sistemas. Uno de los aportes más importantes es que permitirá realizar el análisis de las simulaciones obtenidas de forma automatizada.

#### Estructuración del contenido con una breve explicación de sus partes

**Capítulo 1: Fundamentación teórica.** En este capítulo se explicará en detalles las técnicas que se aplican en el uso de la Biología de Sistemas para la obtención de medicamentos, así como las distintas aplicaciones que existen en el mundo para facilitar el estudio de sistemas biológicos. Se tratarán las variadas tendencias y tecnologías que fueron utilizadas para el desarrollo de este software, además de los diferentes tipos de análisis que se utilizan para investigar el comportamiento de los distintos sistemas biológicos en estudio.

**Capítulo 2: Características del sistema.** En este capítulo se describen los casos de uso encontrados, se definen los actores y trabajadores del negocio y se brinda además una representación gráfica del mismo. Se conocerán las reglas del negocio que fueron definidas para el software. Se hará una descripción del sistema a automatizar y se mostrará un diagrama de casos de uso del sistema para un mejor entendimiento del mismo. Se definirán los actores del sistema y los requerimientos funcionales del sistema que se implementó. Se plantearán los requerimientos mínimos que debe tener el sistema de cómputo donde se vaya a instalar la aplicación.

**Capítulo 3: Análisis y diseño del sistema.** En este capítulo se realiza el diagrama de clases del diseño dando respuesta a la solución que se propone, este es el artefacto principal del Flujo de Trabajo Análisis y Diseño de RUP. Se describen los principios de diseño que se tuvieron en cuenta durante el desarrollo de la aplicación. Se describe además el estilo arquitectónico a utilizar y se define el diagrama de despliegue con el objetivo de mostrar la distribución física de los nodos de cómputo que necesita la aplicación.

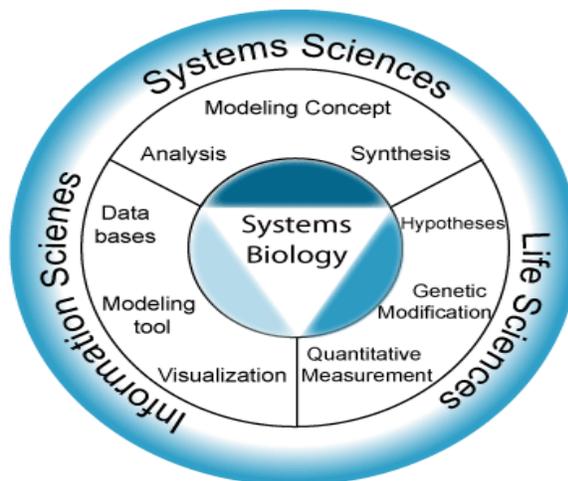
**Capítulo 4: Implementación y Prueba.** En este capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes, para esto se muestra el diagrama de componentes. Además se muestran los casos de prueba de caja negra para cada caso de uso con el objetivo de validar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

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se explicará en detalles las técnicas que se aplican en el uso de la Biología de Sistemas para la obtención de medicamentos, así como las distintas aplicaciones que existen en el mundo para facilitar el estudio de sistemas biológicos. Se tratarán las variadas tendencias y tecnologías que fueron utilizadas para el desarrollo de este software, además de los diferentes tipos de análisis que se utilizan para investigar el comportamiento de los distintos sistemas biológicos en estudio.

### 1.1 Biología de Sistemas. Aplicaciones

La Biología de Sistemas (BS) es una ciencia emergente que trata de entender el funcionamiento de los sistemas como un todo y no de sus componentes por separado. Para ellos hace uso de herramientas de otras ramas de la ciencia como son las ciencias de la computación, las ciencias de la vida y disciplinas ingenieriles. **Fig. 1.**



**Fig. 1.1 Biología de Sistemas.**

### **1.1.1 Biología de Sistemas (BS)**

A nivel mundial existen diferentes criterios acerca de la Biología de Sistemas. Se ha definido la Biología de Sistemas como:

*“... una disciplina que pretende integrar diferentes niveles de información con el fin de entender cómo funcionan los sistemas biológicos. Intenta crear modelos comprensibles de sistemas mediante el estudio de las relaciones y las interacciones entre las diferentes partes de un sistema biológico (por ejemplo, las redes génicas y las redes de interacción de proteínas implicadas en la señalización celular, las rutas metabólicas, los orgánulos, las células, los sistemas fisiológicos, los organismos, etc.).” [1]*

Una definición más acertada sobre la Biología de Sistemas es que *“Intenta describir los mecanismos que regulan la vida de los seres vivos de forma que, conocida la descripción, se pueda predecir qué va a ocurrir en una célula, tejido o ser vivo. Lo que se pretende es en el futuro poder describir a los seres vivos mediante modelos matemáticos que puedan predecir el mal funcionamiento de una célula o un ser vivo.” [2]*

### **1.1.2 Aplicaciones de la Biología de Sistemas**

La Biología de Sistemas es utilizada para predecir el comportamiento de muchas enfermedades y ayuda a la creación de fármacos para su cura y prevención. Tiene grandes aplicaciones como por ejemplo las cardiopatías. *“La Biología de Sistemas intenta recoger todo el conocimiento sobre esta patología que ha atesorado el bioquímico, el inmunólogo, el médico...y encontrar nuevas terapias para prevenir. En una enfermedad como la gota, que sólo dispone de un fármaco, mediante el conocimiento de la enfermedad, si se estudia bien la ruta metabólica, se pueden crear nuevos fármacos con menos efectos secundarios” [2]*

La comunidad de investigadores de la BS ha propuesto un modelo al que llaman “las 4 Ms” para el estudio de los sistemas biológicos, dividiéndolo esencialmente en dos áreas, una experimental y otra computacional. Dentro del área experimental encontramos la manipulación de datos y las mediciones, dentro del área computacional la minería de datos y la modelación. La aplicación que se desarrollará está concebida dentro del área computacional, específicamente en la Minería de Datos.

## 1.2 Modelo matemático. Modelación de sistemas biológicos

El estudio de sistemas biológicos se hace más factible mediante la creación de un modelo matemático que lo describa. Un modelo no es más que una invención que sirve para explicar una serie de datos que se quieran interpretar. Para que un modelo sea útil todos los elementos que lo componen deben ser planteados coherentemente, es decir, tiene que poder explicar lo que sucede de forma lógica. Los modelos no son más que esquemas o ecuaciones que explican aquello que se quiere estudiar, lo cual puede excluir características del ente en estudio que se consideren no sean importantes.

*“Un modelo matemático es una representación para un objeto o proceso, en el cual quedan plasmadas sus principales características, a partir de conjeturas o suposiciones iniciales, estableciendo un compromiso entre complejidad y exactitud; en esta idealización es posible utilizar entes, estructuras y leyes de la matemática; permitiendo interpretar los resultados del modelo en términos del objeto o proceso estudiado.”[3]*

La más acertada para la investigación que se llevó a cabo es que *“El modelamiento matemático es una de las herramientas que se utiliza hoy en día para el estudio de problemas en medicina, biología, fisiología, bioquímica y farmacocinética; sus objetivos primordiales son describir, explicar y predecir fenómenos y procesos en dichas áreas. El modelo matemático permite representar un problema médico o biológico de una manera objetiva en que se definen una serie de relaciones matemáticas entre las mediciones cuantitativas (del problema) y sus propiedades.”[4]*

Para la modelación de sistemas biológicos se puede hacer uso de distintos tipos de modelos. Los más utilizados son los autómatas celulares y la dinámica de población. Para el desarrollo de esta aplicación se usa el segundo tipo, porque permite realizar la modelación de un sistema biológico de forma sencilla. En este caso se definirá un modelo matemático basado en sistemas de ecuaciones diferenciales ordinarias.

### 1.3 Métodos de resolución de modelos matemáticos

Para comprender el comportamiento de los sistemas biológicos se deben analizar los resultados obtenidos del modelo matemático que lo describe. Existen distintos métodos para su resolución, algunos de estos métodos son:

#### 1.3.1 Métodos de Runge-Kutta

*“Basados en la Serie de Taylor, donde la cantidad de términos que se decidan tomar de la serie determinará la precisión del método. En estos métodos se le denomina paso (h) a  $X_{(n+1)} - X_n$  este paso puede ser fijo obteniéndose puntos equidistantes; y también se puede variar haciendo aumentar el paso al doble en caso de detectar errores pequeños o reducirlo a la mitad en presencia de errores grandes, esto complica el programa de cálculo, pero aumenta la eficiencia y en la mayoría de los casos son más rápidos.*

*El método de segundo orden requiere evaluar cada función 2 veces en cada paso de integración y el error local es del orden  $h^3$ , de la misma forma el método de cuarto orden necesita 4 evaluaciones de cada función y posee un error local del orden  $h^5$ ; esto posibilita que se pueda integrar con un paso mayor para lograr un error similar al que obtendríamos si utilizásemos el de orden dos, mejorando así el tiempo de ejecución del algoritmo.*

*Los métodos de Runge-Kutta son métodos de paso simple, que sólo requieren de los resultados del paso anterior, esto les brinda la posibilidad de auto iniciarse ya que parten de las condiciones iniciales. Al igual que todos los métodos de paso simple no presentan inestabilidad numérica para paso  $h$  suficientemente pequeño, esto implica que pequeños cambios en las condiciones iniciales del sistema sólo originan cambios acotados en la solución. Una desventaja de los métodos Runge-Kutta es que requieren múltiples evaluaciones de la funciones en cada paso... “[5]*

#### 1.3.2 Métodos de Adams

*“Utilizan procedimientos de integración numérica utilizando un polinomio de interpolación para aproximar la función  $f(x, y(x))$  de  $y' = f(x, y)$ , las diversas variantes de tomar este polinomio originan múltiples métodos de Adams.*

*Estos métodos son de paso múltiple ya que necesitan de los valores iniciales en varios puntos equidistantes para su iniciación, esta información se puede obtener aplicando primero algún método de*

*paso simple como los de Runge-Kutta, constituyendo una desventaja de los métodos de paso múltiple ante los de paso simple. Otra característica desventajosa de los métodos de Adams comparándolos con los de Runge-Kutta es que poseen un error local mayor en procedimientos del mismo orden, teniendo así menos precisión. Los métodos de Adams poseen estabilidad condicional.* [5]

Para el caso de esta aplicación se usarán los métodos ode45 y ode23, los cuáles son funciones de Matlab que implementan el método de Runge-Kutta para la resolución de modelos matemáticos de ecuaciones diferenciales ordinarias:

- ✓ ode23 - método Runge-Kutta de largo de paso variable que combina un método de orden dos con uno de orden tres.
- ✓ ode45 - método Runge-Kutta-Fehlberg de largo de paso variable que combina un método de orden cuatro con uno de orden cinco.

Al proceso de resolver un modelo matemático asociado a un sistema biológico mediante dichos métodos, es a lo que se le llama simulación de sistemas biológicos.

#### **1.4 Simulación. Simulación biológica**

*La simulación “es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso, y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar estrategias con las cuales se puede operar sobre él.”* [6]

La simulación computarizada surge desde la década del 40 como una forma de modelar situaciones de la vida real, es una técnica numérica para reproducir artificialmente un fenómeno o el funcionamiento de un sistema, utilizando modelos matemáticos y lógicos, específicos para cada proceso, que describen el comportamiento y la estructura de dicho fenómeno a través del tiempo. La simulación es una poderosa herramienta que es ampliamente utilizada por los científicos que estudian los sistemas complejos. Esta técnica incluye para estos estudios, tanto la construcción del modelo como su uso analítico. Ofrece la posibilidad de comprimir el tiempo, esfuerzo y cantidad de recursos necesarios para tomar decisiones.

La simulación biológica surge desde los años 60, se utiliza para simular cómo funciona un sistema biológico de manera integral. *“Permite a las compañías farmacéuticas reducir de manera significativa el número de experimentos de laboratorio requeridos para identificar posibles objetivos de los fármacos. El modelado “in silico” (en oposición al “in vitro” [en vidrio] o “in vivo” [en vivo]) también permite que los investigadores predigan los efectos de los fármacos en el cuerpo humano, incluyendo su eficacia y seguridad.”*[7]

#### **Desventajas de las simulaciones biológicas:**

- *“Un buen modelo de simulación puede resultar bastante costoso; a menudo el proceso de desarrollar un modelo es largo y complicado.*
- *La simulación no genera soluciones óptimas a problemas de análisis cuantitativos, en técnicas como cantidad económica de pedido, programación lineal. Por ensayo y error se producen diferentes resultados en repetidas corridas en el computador.*
- *Los directivos generan todas las condiciones y restricciones para analizar las soluciones. El modelo de simulación no produce respuestas por sí mismo.*
- *Cada modelo de simulación es único. Las soluciones e inferencias no son usualmente transferibles a otros problemas.*
- *Siempre quedarán variables por fuera y esas variables pueden cambiar completamente los resultados en la vida real que la simulación no previó.”*[8]

#### **Ventajas de las simulaciones biológicas:**

- *“Es un proceso relativamente eficiente y flexible.*
- *Puede ser usada para analizar y sintetizar una compleja y extensa situación real, pero no puede ser empleada para solucionar un modelo de análisis cuantitativo convencional.*
- *En algunos casos la simulación es el único método disponible.*
- *Los modelos de simulación se estructuran y resuelven en general problemas trascendentes.*
- *Los directivos requieren conocer cómo se avanza y qué opciones son atractivas; el directivo con la ayuda del computador puede obtener varias opciones de decisión.*
- *La simulación no interfiere en sistemas del mundo real.*
- *La simulación permite la inclusión de complicaciones del mundo real.*
- *La simulación permite estudiar los efectos interactivos de los componentes individuales o variables para determinar las más importantes.”*[8]

Las simulaciones biológicas han sido de interés para muchas compañías que a nivel mundial se dedican al estudio de las mismas. Estas compañías se han visto en la necesidad de desarrollar aplicaciones que permitan agilizar los cálculos que se deben realizar para la simulación. En el próximo epígrafe se abordarán algunas de las aplicaciones desarrolladas en el mundo, vinculadas a la simulación de sistemas biológicos.

### 1.5 Aplicaciones computacionales vinculadas al estudio de la Biología de Sistemas

En la actualidad a nivel internacional se han desarrollado muchas aplicaciones destinadas a la simulación y estudio de sistemas biológicos, entre las cuales se destacan las siguientes:

1. **BioNetGen:** es una herramienta para generar modelos matemáticos automáticamente de sistemas biológicos con las reglas especificadas por el usuario, que definen las interacciones dentro del sistema. Tiene como ventaja, la habilidad de hacer predicciones acerca del comportamiento de un sistema mediante un método basado en las máquinas de soporte vectorial, donde a partir de la información de la secuencia de la proteína se puede predecir si un nuevo fenotipo, derivado de uno de los nucleótidos, se puede relacionar con una enfermedad genética en seres humanos.

Limitante: sólo modela los tipos de reacciones en red de genes y lo hace utilizando funciones booleanas.

2. **BioCharon:** esta herramienta permite a un usuario visualizar varias proyecciones diferentes y secciones transversales de una gráfica. Contiene un bloc de dibujo llamado Bio-Sketch Pad que es una herramienta interactiva para el modelado y diseño biomolecular y fácil de usar con poderosas herramientas para la teoría de control y sistemas híbridos. Brinda también la posibilidad de convertir modelos originales e híbridos SBML en el formato de entrada de la herramienta CHARON. Otra ventaja es que cuenta con una herramienta de análisis que permite al usuario realizar análisis de consistencia.

Limitante: realiza sólo análisis por medio de gráficas y respecto a la consistencia.

3. **ByoDyn:** permite simular un modelo, analiza la sensibilidad del sistema con relación a los parámetros y estima los parámetros cinéticos usando datos experimentales. Sus principales

ventajas son: utilizar formatos estándar que permiten que el programa pueda operar con modelos desarrollados por otros científicos y tener gran número de opciones para ejecutar que pueden ser aprobadas por el usuario, dando soporte a los cálculos complicados.

Limitante: está enfocado al estudio del funcionamiento de la regulación de la red de genes y sólo realiza análisis de sensibilidad.

4. **Jsim**: es un sistema de simulación implementado en el lenguaje de programación Java. Permite construir modelos numéricos cuantitativos y analizarlos con respecto a datos experimentales. Ha estado enfocado a la Fisiología y a la Bio-medicina aunque es un motor computacional que es bastante general y aplicable a un gran rango de dominios científicos. El tipo de análisis que realiza es el análisis de sensibilidad, el cual es el proceso de determinar cómo pequeñas perturbaciones en las entradas del modelo afectan las salidas de dicho modelo.

Limitante: supone que el usuario deba tener algún conocimiento de Java y no tiene una base de datos donde almacenar los resultados de hacer variar los parámetros de un modelo.

5. **SimBiology**: permite realizar análisis de sensibilidad de todos los sistemas para los parámetros especificados y las condiciones iniciales. Este análisis ayuda a mejorar el entendimiento del sistema y permite determinar los parámetros y las condiciones de mayor importancia. Para realizar la simulación se pueden especificar el tipo de solución que se desea, ya sea estocástica o determinista, el tiempo inicial y final de la simulación, como también qué análisis dimensional se realizará a cada simulación. Este análisis dimensional permite comparar combinaciones de cantidades físicas definidas en el modelo y verificar que las expresiones matemáticas son consistentes.

Limitante: sólo realiza análisis de sensibilidad. No incluye una base de datos donde almacenar los resultados de las simulaciones.

6. **Vcell**: permite crear modelos biológicos del cual se generará un código matemático con lo cual podrá ser representado gráficamente en 1D, 2D, 3D y que es necesario para correr las simulaciones que pueden ser complejas, variando uno o muchos parámetros según una lista especificada de valores posibles o de un rango definido. Las simulaciones son corridas por Internet en ochenta y cuatro servidores. También permite realizar nuevas representaciones

geométricas si el usuario lo desea. Los modelos pueden ser reutilizados, actualizados y publicados para el uso de otros usuarios, o compartidos privadamente. Los resultados de las simulaciones son guardados en una BD y se pueden exportar en varios formatos.

Limitante: no realiza análisis de los resultados.

7. **Cellware:** se ha diseñado para modelar y simular reacciones bioquímicas de la célula, utilizando computación Grid, lo que le permite simular grandes cantidades de modelos y reducir el tiempo de cálculo en gran medida. Cellware es capaz de importar y exportar modelos en formato SBML. Fue diseñado en lenguaje Java con la intención de ser multiplataforma, se ha probado en sistemas operativos como GNU/Linux, Windows y Mac OS. La interfaz de usuario es amigable para el biólogo que la utilice y tiene herramientas que permiten construir adecuadamente los modelos de células deseados. Además incluye herramientas de análisis como estimación de parámetros y algoritmos para el diseño de gráficas. Contiene un fichero donde se almacena toda la información correspondiente al modelo y a las simulaciones realizadas.

Limitante: no incluye una base de datos con la cual realizar consultas de los resultados obtenidos de las simulaciones. Tiene implementado sólo dos herramientas para el análisis: estimación de parámetros y diseño de gráficas.

8. **Biogrid:** es una plataforma que tiene el objetivo de ofrecer un acceso a las herramientas de visualización, a la simulación y a los datos en los centros científicos que se ocupan de las investigaciones sobre el genoma y así juntar los esfuerzos del trabajo de varios equipos de investigación alrededor del mundo. La integración de los recursos se realiza a través de la computación Grid. Biogrid permite que el usuario final utilice servicios a través de una interfaz sin importar si el algoritmo está en la plataforma local o en un ordenador remoto. El usuario seleccionará un objeto, por ejemplo una proteína y la plataforma ofrecerá automáticamente la lista de los servicios aplicables a este objeto particular que serán los que podrá usar.

Limitante: sólo se podrán realizar algunas funciones con elementos o proteínas que se encuentren en la plataforma. Imposibilidad de construir un modelo matemático para estudiar un problema.

9. **ImmunoGrid:** es una implementación virtual del sistema inmune humano usando tecnología Grid. Es capaz de simular procesos a escala natural y proporciona herramientas para el diseño de

vacunas. Lo novedoso está en lograr la conexión de las interacciones a nivel molecular con los modelos a nivel de sistema. Se hace uso de bases de datos para almacenar, manipular y modelar datos inmunológicos, lo que facilita la búsqueda de modelos matemáticos y predictivos. Estas bases de datos son especializadas y utilizan estándares, por lo que facilitarán obtener información más detallada, con un alto nivel de calidad y veracidad.

Limitante: la desventaja de este software es que fue concebido para el estudio del sistema inmune humano, por lo que sólo puede ser aplicado en investigaciones relacionadas con dicho sistema.

## **1.6 Importancia del desarrollo de una nueva aplicación**

Después de haber realizado un estudio acerca de las aplicaciones desarrolladas en el mundo, dedicadas a la simulación de sistemas biológicos, se llegó a la conclusión de que no satisfacen de forma general las necesidades de los investigadores, pues en la mayoría de las aplicaciones estudiadas se hacen simulaciones, pero no se realiza un análisis posterior a las mismas. Entre estas aplicaciones se encontró que las que son capaces de realizarlo, sólo lo hacen a través de un único método, generalmente analizando la sensibilidad del sistema o por consistencia, pero no agrupan varios tipos de análisis.

Dentro de las aplicaciones estudiadas, la más completa en cuanto a la realización de simulaciones es Vcell, la cual realiza las simulaciones de forma distribuidas a través de ochenta y cuatro servidores en Internet, lo que supone una gran ventaja para realizar este tipo de estudio biológico. Esta aplicación además, almacena los resultados de las simulaciones en una BD, permite la variación de los parámetros en un rango de valores para realizar múltiples simulaciones y permiten reutilizar los modelos, actualizarlos y publicarlos, estas características satisfacen en parte las necesidades de los investigadores, pero la desventaja fundamental a pesar de ser lo suficientemente integral es que no realiza análisis a los resultados obtenidos de la simulación, necesidad fundamental para los que se dedican a este tipo de investigaciones, de lo contrario tendrían que analizar las simulaciones obtenidas una a una y de forma poco eficiente.

Cellware, Biogrid, Immunogrid fueron otras de las aplicaciones estudiadas. Estos software utilizan tecnología Grid para llevar a cabo sus estudios. Dicha tecnología permite hacer uso de todas las computadoras de una red para realizar muchos cálculos y minimizar el tiempo de obtención de los

resultados, más importante aún es que ofrece la oportunidad de utilizar las PCs cuando estén sin carga alguna de trabajo.

La importancia del uso de la tecnología Grid en la aplicación que se propone, es que permitirá hacer una gran cantidad de simulaciones en un tiempo razonablemente corto, debido al uso de varias PCs conectadas en red, todas resolviendo los cálculos a la misma vez y devolviendo sus respuestas. Esto facilita el estudio de sistemas biológicos y en general, esta tecnología es una gran herramienta para realizar investigaciones que requieran de numerosos y complejos cálculos.

La tecnología Grid es un tipo de sistema paralelo y distribuido que permite compartir, seleccionar y agregar dinámicamente recursos autónomos que están dispersos geográficamente, en función de su disponibilidad, capacidad, rendimiento, coste o de las necesidades de los usuarios. “ ... *en una Grid los recursos no están administrados de manera centralizada, sino que funcionan de manera análoga a la red eléctrica, integrando y conectando ordenadores de alto rendimiento con distintas arquitecturas a través de la red, y reasignando los servicios en función de los picos de actividad.*”[9]

### **1.6.1 Ventajas de Computación Grid**

1. *“Bajo coste entrada*
  - *Puede empezarse con inversiones mínimas*
  - *No hay que comprar un supercomputador*
  - *No hay gastos de soporte y mantenimiento*
2. *Flexibilidad en la potencia de uso*
  - *Pueden gestionarse picos de gran intensidad*
3. *No hay pérdida de valor de la infraestructura por obsolescencia”*[10]

### **1.6.2 Algunas aplicaciones de la tecnología Grid**

- Aplicaciones de cálculo numérico intenso.
- Tratamiento masivo de datos y requerimientos de procesos muy exigentes.

La importancia de la aplicación desarrollada viene dada precisamente, por la posibilidad que ofrece de simular de forma distribuida haciendo uso de tecnología Grid para así agilizar el proceso de cálculo, de

contar con una base de datos que permita almacenar los datos referente a un sistema biológico y lo más importante es que permite realizar distintos tipos de análisis: obteniendo gráficas de dinámica de la población, utilizando algoritmos de clustering y a través de reglas definidas por el usuario.

### 1.7 Tipos de análisis incluidos en la aplicación desarrollada

Para un mejor entendimiento de la aplicación desarrollada se deben comprender los tipos de análisis incluidos en la misma, con los cuales el usuario podrá obtener un criterio interesante y útil para el estudio que lleve a cabo, a continuación se explicará detalladamente en qué consiste cada uno:

#### 1.7.1 Gráficas de dinámicas de población

Estas gráficas mostrarán la dinámica de una población en relación a otra o al tiempo de simulación. Para esto se escogerá una simulación de la base de datos, en caso de no existir se realizará la misma y luego se graficará. Ejemplo de una gráfica de población Vs tiempo:

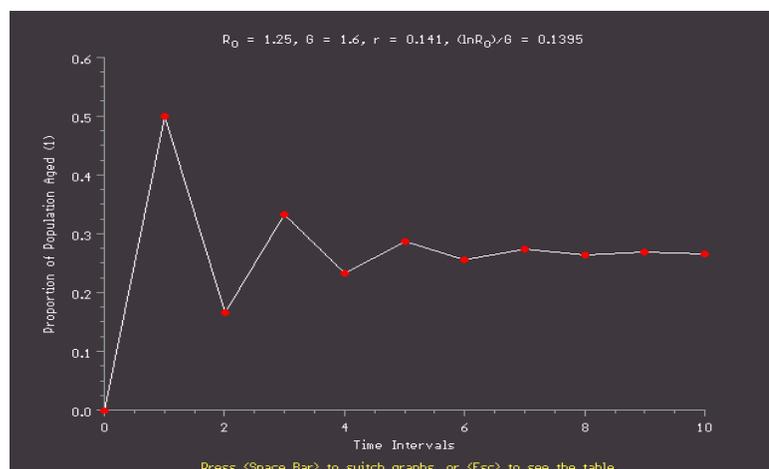


Fig. 1.2. Gráfica de dinámica de la población.

### 1.7.2 Algoritmos de Clustering

#### 1.7.2.1 Concepto de clustering

Según el Dr. Elmer A. Fernández, Universidad Católica de Córdoba, Fac. Ingeniería:

*Clustering*: Se basa en intentar responder cómo es que ciertos objetos (casos) pertenecen o “caen” naturalmente en cierto número de clases o grupos, de tal manera que estos objetos comparten ciertas características.

Esta definición plantea que los objetos que comparten características similares están agrupados razonablemente. El objetivo de los algoritmos de clustering es precisamente encontrar los objetos que comparten características comunes y agruparlos en un cluster.

- *Objetos (Datos) y Variables*

Un Objeto es un dato, el cual esta formado por un conjunto finito de variables. Las variables se pueden clasificar en:

- Nominales: son variables discretas, pero que no tienen un orden especificado (Ej: color de ojos, raza).
- Numéricas: son números reales en general (Ej: 1, 2, 3.5, 8.8).
- Ordinales: son variables discretas con una relación de orden (Ej: temperatura: Alta, Media, baja).
- Binarias: sólo pueden tomar dos estados posibles (dicotómicas) (Ej: sexo, estado civil).

- *Funciones de Distancias*

“Las técnicas de agrupamiento se basan fundamentalmente en el concepto de similitud o disimilitud entre ejemplos y agrupaciones. Muchas veces se utilizan métricas o distancias para medir la similitud entre ejemplos. Las métricas más utilizadas son:

$$✓ d_{ij} = \sum_{k=1}^p W_k |x_{ik} - x_{jk}| \text{ City-Block } ()$$

$$✓ d_{ij} = \sqrt{\sum_{k=1}^p W_k (x_{ik} - x_{jk})^2} \text{ Euclidean } ()$$

$$✓ d_{ij} = \sqrt[\lambda]{\sum_{k=1}^p W_k (x_{ik} - x_{jk})^\lambda} \lambda > 0 \text{ Minkowski } ()$$

$$✓ H(\bar{x}, \bar{y}) = (\{i \mid 1 \leq i \leq n \wedge x_i \neq y_i\}) \text{ Hamming (normalmente aplicada a vectores binarios, da el número de componentes con valores distintos)}$$

$$✓ \cos(\vec{x}, \vec{y}) = \sum_i \left( \frac{x_i * y_i}{\left( \sqrt{\sum x_i^2} * \sqrt{\sum y_i^2} \right)} \right) \text{Distancia del coseno (usada en recuperación de la información}$$

donde un documento se representa por un vector de pesos).[11]

Existen varios tipos de algoritmos de clustering. Los que se usan en esta aplicación son el K-Means, Cobweb y X-Means. Estos algoritmos están implementados en la aplicación Weka y se hace uso de los mismos. En esta aplicación el algoritmo SimpleKMeans implementa al algoritmo K-Means, Cobweb pertenece a la familia de algoritmos jerárquicos y X-Means es una variante del K-Means, a continuación se profundizará más en estos temas.

### 1.7.2.2 Algoritmo Kmeans

*“El algoritmo de las K-medias es un algoritmo de partición. Básicamente este algoritmo busca formar clusters (grupos), los cuales serán representados por K objetos. Cada uno de estos K objetos es el valor medio de los objetos que pertenecen a dicho grupo.”*[12]

Los pasos básicos para aplicar el algoritmo k-medias son muy simples. Primeramente se determinan la cantidad de clusters en los que se quiere agrupar la información. Luego se asume de forma aleatoria los centros por cada cluster. Una vez encontrados los primeros centroides el algoritmo hará los tres pasos siguientes:

1. Determinar las coordenadas del centroide.
2. Determinar la distancia de cada objeto a los centroides.
3. Agrupar los objetos basados en la menor distancia.

Estos pasos se harán hasta que se alcance la estabilidad de los objetos, esto se refiere a cuando los objetos no cambian más de grupo de una iteración a otra, lo cual quiere decir que cada objeto está en el cluster adecuado.

#### **Desventajas:**

- ✓ “Se necesita conocer k cantidad de clusters) de antemano.
- ✓ Sensible a ruido.
- ✓ Puede caer en mínimos locales.

- ✓ *No trata datos nominales.* [13]

**Ventajas:**

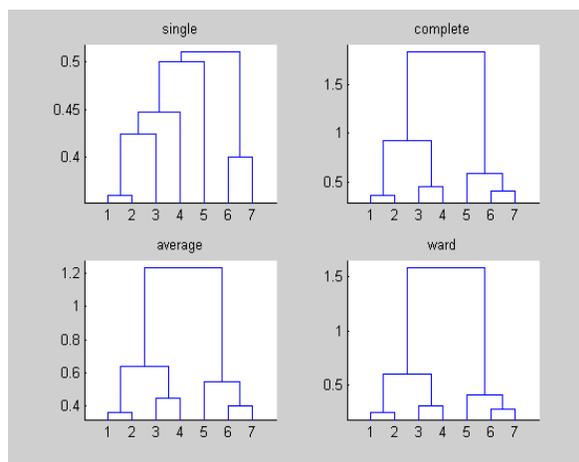
- ✓ *“Entre los algoritmos de particionamiento es eficiente.*
- ✓ *Implementación sencilla.* [13]

**1.7.2.3 Algoritmo Hierarchical (Jerárquico):**

Algoritmos jerárquicos: crean una jerarquía que descompone el conjunto de datos usando algún criterio. Construye un árbol binario o dendograma a partir de un conjunto de ejemplos:

- ✓ Aglomerativo (bottom-up) métodos que comienzan con cada ejemplo en un cluster diferente y combinan iterativamente los clusters para formar clusters mayores. (Ej. AGNES, Agglomerative Nesting ).
- ✓ Divisivo (top-down) métodos que comienzan con todos los ejemplos en un mismo cluster y los separan en clusters más chicos (Ej. DIANA, Divisive Analysis).

Un dendograma muestra cómo se mezclan los cluster, de manera que cortando el dendograma en diferentes niveles se consiguen diferentes clusters. **Fig.1.3**



**Fig. 1.3. Dendogramas.**

Pasos que rigen el algoritmo Hierarchical:

- ✓ El primer paso es calcular las distancias entre todos los pares de objetos. Esto es lo mismo que asumir que cada objeto constituye un cluster  $\{C_1, \dots, C_N\}$ .
- ✓ Se buscan los dos clusters más cercanos  $(C_i, C_j)$ , estos se juntan y constituyen uno solo  $C_{ij}$ .
- ✓ Se repite el paso 2 hasta que no quedan pares de comparación.

La cuestión crítica de este método es la forma de “juntar” los clusters entre sí, se utilizan básicamente tres formas:

- Enlace simple (usa la menor distancia entre objetos de dos grupos)

$$d(r, s) = \min(\text{dist}(x_{ri}, x_{sj})), i \in (1, \dots, n_r), j \in (1, \dots, n_s)$$

- Enlace promediado (usa el promedio de las distancias entre todos los pares de objetos en el cluster “r” y cluster “s”)

$$d(r, s) = \frac{1}{(n_r n_s)} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} \text{dist}(x_{ri}, x_{sj})$$

- Enlace completo (usa la mayor distancia entre objetos de dos grupos)

$$d(r, s) = \max(\text{dist}(x_{ri}, x_{sj})), i \in (1, \dots, n_r), j \in (1, \dots, n_s)$$

Uno de los algoritmos que pertenecen a la familia de algoritmos jerárquicos es el Cobweb. El Cobweb se caracteriza por la utilización de aprendizaje incremental, quiere decir, realiza las agrupaciones instancia a instancia. Durante la ejecución del algoritmo se forma un árbol (árbol de clasificación) donde las hojas representan los segmentos y el nodo raíz engloba por completo el conjunto de datos. Al principio, el árbol consiste en un único nodo raíz. Las instancias se van añadiendo una a una y el árbol se va actualizando en cada paso. La clave para saber cómo y dónde se debe actualizar el árbol la proporciona una medida denominada *utilidad de categoría* que mide la calidad general de una partición de instancias en un segmento. Este algoritmo es además, muy sensible a otros dos parámetros: Acuity y Cutoff.

**Acuity:** Este parámetro es muy necesario, ya que la utilidad de categoría se basa en una estimación de la media y la desviación estándar del valor de los atributos, pero cuando se estima la desviación estándar del valor de un atributo para un nodo en particular, el resultado es cero si dicho nodo sólo contiene una instancia. Así pues, el parámetro Acuity representa la medida de error de un nodo con una sola instancia, es decir, establece la varianza mínima de un atributo.

**Cutoff:** Este valor se utiliza para evitar el crecimiento desmesurado del número de segmentos. Indica el grado de mejoría que se debe producir en la utilidad de categoría para que la instancia sea tenida en cuenta de manera individual. En otras palabras: cuando no es suficiente el incremento de la utilidad de categoría en el momento en el que se añade un nuevo nodo, ese nodo se corta, conteniendo la instancia otro nodo ya existente.

Al algoritmo Cobweb no hay que proporcionarle el número exacto de clusters que se desea, sino que en base a los parámetros anteriormente mencionados, encuentra el número óptimo.

### **Ventajas y Desventajas de los algoritmos jerárquicos.**

#### **Desventajas:**

*“No puede recuperarse de decisiones incorrectas.*

*Computacionalmente costoso”[13]*

#### **Ventajas:**

*“No se necesita conocer el número de clusters  $k$ .” [13]*

#### **1.7.2.4 Algoritmo X-Means**

X-Means es una variante mejorada del algoritmo K-Means. Su ventaja fundamental está en haber solucionado una de las mayores deficiencias presentadas en K-Means, el hecho de tener que seleccionar a priori el número de clusters que se deseen obtener, a X-Means se le define un límite inferior  $K$  (número mínimo de clusters) y un límite superior  $K_{max}$  y este algoritmo es capaz de obtener en ese rango el número óptimo de clusters. Además se ha comprobado que sus resultados son más fiables que los obtenidos con el K-Means debido a que presentan un valor de distorsión menor, son mucho mejor para realizar clusters de un conjunto grande de datos y es incluso una variante mucho más rápida.

En esencia el algoritmo comienza con  $K$  igual al límite inferior del rango dado y continúa agregando centroides donde son necesarios hasta que se alcance el límite superior. Durante este proceso, el

conjunto de centroides que alcanza el mejor valor son almacenados y estos serían la salida final. El algoritmo consiste de las siguientes operaciones básicas, repetidas hasta su terminación.

- ✓ Mejorar-Parámetros (Improve-Params)
- ✓ Mejorar-Estructura (Improve-Structure)
- ✓ Si  $K > K_{max}$  parar y reportar el mejor modelo de valores encontrados durante la búsqueda, sino ir al paso 1.

**Improve-Params:** esta operación es muy sencilla, consiste en correr el k-Means convencional hasta su convergencia.

**Improve-Structure:** esta operación consiste en encontrar si deben aparecer nuevos centroides y dónde deben aparecer. Esto se alcanza dejando que algunos centroides se dividan en dos.

### 1.7.3 Reglas definidas por el usuario

Para hacer una simulación biológica primeramente se pasan como datos los parámetros y condiciones iniciales, para poder resolver el modelo matemático. Este tipo de análisis le brinda la posibilidad al usuario de realizar un estudio con las simulaciones que cumplan con una regla definida por él.

Ejemplo: Suponiendo que se realice una simulación que contenga tres poblaciones (A,B,C), dos parámetros  $(P_1, P_2)$ , cuyas condiciones iniciales sean 1 y 3 respectivamente, el usuario puede establecer una regla como la siguiente:

$$A > B < C$$

Dadas estas reglas definidas por el usuario, se hará una búsqueda en la base de datos de las simulaciones que cumplan con dichas reglas y tengan los parámetros establecidos para mostrar una gráfica de fronteras mostrando en qué rango de los valores de los parámetros están las simulaciones que cumplen con las reglas.

Una vez comprendidas las funcionalidades que debe tener el software y la importancia de su implementación se prosiguió a realizar un estudio de las tecnologías más adecuadas para llevar a cabo la realización del mismo.

## 1.8 Herramientas y metodologías utilizadas en la aplicación

Para la realización de una aplicación siempre se deben definir los tipos de tecnologías a utilizar, así como las herramientas y metodologías que serán de mayor utilidad para su implementación. A continuación se explica en detalle cada una de las tecnologías que fueron seleccionadas para llevar a cabo la implementación y documentación de la aplicación desarrollada.

### 1.8.1 Metodologías de desarrollo de software

La dificultad que presentan hoy en día los desarrolladores a la hora de realizar una aplicación es la necesidad de saber cómo organizar las actividades para cada desarrollador por separado y para el equipo, definir qué artefactos deben ser creados y contar con una serie de criterios que permitan controlar y medir los productos que se obtienen, por lo tanto se necesita de una metodología capaz de dirigir estas actividades y así convertir los requisitos de los usuarios en un producto software. En el mundo existen distintas metodologías para dirigir las actividades vinculadas al proceso de desarrollo de software, entre estas metodologías se estudiaron las siguientes:

#### **XP (Extreme Programming)**

XP es una metodología ágil, más bien utilizada en proyectos de corto plazo, desventaja fundamental para la aplicación de esta metodología al software en cuestión, debido a que el mismo es un proyecto grande y que requiere un plazo de tiempo más bien largo para su realización. Esta metodología se utiliza para proyectos que tienen una urgencia en la fecha de entrega, por lo que hace uso de una programación rápida o extrema.

La metodología *XP* se basa en:

- *“Pruebas Unitarias: basadas en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.*
- *Re-fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.*
- *Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de*

*trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.* [14]

## **RUP**

El Proceso Unificado del Rational (RUP) goza de mucho prestigio debido a que ya cuenta con más de 30 años de experiencia y fue desarrollada teniendo en cuenta y tomando lo mejor de otras metodologías orientadas a objetos.

*“El Proceso Unificado del Rational es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.”* [15]

Pero los aspectos más meritorios del proceso son: estar dirigido por casos de uso, iterativo, incremental y centrado en la arquitectura. Estas características van a permitir una planificación ajustada al avance que vaya presentando el producto, además, se irá verificando que las funcionalidades que van siendo implementadas en el software sean exactamente las que desea el usuario, evitando así que en la etapa final, es decir, en la entrega del producto, sea cuando el usuario le comunique a los desarrolladores si está de acuerdo o no con el producto obtenido y de no estarlo se tenga que comenzar de nuevo todo el proceso, por lo que se irán eliminando los errores durante el proceso, logrando obtener un producto con calidad.

El Proceso Unificado del Rational aporta además un enfoque disciplinado a la asignación de tareas y responsabilidades en proyectos de desarrollo de software. También permite dirigir las tareas de desarrolladores individuales y equipos de trabajo como una sola, incluso ofrece criterios para monitorear y medir los productos y actividades del proyecto. Una característica importante es que permite corregir errores en cada iteración y es flexible a cambios en los requerimientos. Por estas características se eligió como la metodología a emplearse durante el proceso de desarrollo de este software.

## 1.8.2 Herramientas CASE

### Rational Rose

Rational Rose es una herramienta software para el Modelado Visual mediante UML de sistemas software. Permite especificar, analizar y diseñar el sistema antes de codificarlo. Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Características Rational Rose:

- *“Mantiene la consistencia de los modelos del sistema software.*
- *Chequeo de la sintaxis UML.*
- *Generación documentación automáticamente.*
- *Generación de código a partir de los modelos.*
- *Ingeniería Inversa (crear modelo a partir código). [15]*

Resulta algo complicado de utilizar esta herramienta, debido a que propone el empleo del proceso de desarrollo unificado, el cual es altamente complejo e incorpora algunas funcionalidades a través de otras aplicaciones pero no las integra.

### Visual Paradigm

Es una potente herramienta CASE para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML, proporciona a los desarrolladores una plataforma que les permita diseñar un producto con calidad de una forma rápida. Facilita la interoperabilidad con otras herramientas CASE como el Rational Rose y se integra con las siguientes herramientas Java: Eclipse/IBM WebSphere, Jbuilder, NetBeans IDE, Oracle Jdeveloper, BEA Weblogic. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal. Genera código y realiza ingeniería inversa para diez lenguajes de programación, Java, C++, CORBA IDL, PHP, XML Schema y ADA. En adición se genera código para C#, Visual Basic.net, Object Definition Language(ODL), Flasch Action Script, Delphi, Perl y Phytion. Se integra con el Visio para importar imágenes del mismo para realizar los diagramas de despliegue. Genera documentación para el proyecto en HTML, MS Word y PDF. Además exporta e importa los diagramas en el estándar XML y como imágenes (ya sea con extensiones jpg o png). Es gratis en su edición Community y es multiplataforma.

Debido a que el sistema operativo que se está utilizando es Ubuntu (distribución de GNU/Linux) se decidió utilizar el Visual Paradigm 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, es además mucho más fácil de utilizar que el Rational Rose y permite crear Bases de Datos en MySQL.

### **1.8.3 Lenguaje de Modelado**

#### **UML**

El Proceso Unificado de Modelado (UML) es un lenguaje gráfico para visualizar y documentar los elementos de los sistemas orientados a objetos. Dicho lenguaje es una notación unificada con la que se permite lograr un entendimiento que propicie el intercambio entre los usuarios y los desarrolladores. Se ha convertido en un estándar de la industria del software, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos Grady Booch, Ivar Jacobson y Jim Rumbaugh. El UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo, diagrama de actividad). Como se utilizó el Visual Paradigm como herramienta CASE se hace uso de UML 2.1.

### **1.8.4 Lenguajes de programación**

#### **Java**

Ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas. Es orientado a objetos, trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando. Es robusto, realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. Es multithreaded (multihilos), permite muchas actividades simultáneas en un programa. Los threads son básicamente pequeños procesos o piezas independientes de un gran proceso.

### **1.8.5 Herramientas de desarrollo**

#### **Netbeans**

Para el desarrollo de la aplicación fue necesario escoger la herramienta de desarrollo a utilizar según el lenguaje de programación que se seleccionó, por lo tanto se hizo uso del IDE Netbeans que no es más que una herramienta para el desarrollo de aplicaciones de escritorio usando Java. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios. Además es un proyecto GNU (libre), tiene un excelente diseñador de interfaces integrado, es muy rápido y fácil de usar.

#### **MatLab**

Es un entorno de computación y desarrollo de aplicaciones, totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. MatLab integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían tradicionalmente, sin necesidad de hacer uso de la programación tradicional.

MatLab dispone también en la actualidad, de un amplio abanico de programas de apoyo especializados, Toolboxes como así se le denominan, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación destacando entre ellos el 'toolbox' de procesos de imágenes, señal, control y redes neuronales.

### 1.8.6 Gestor de Base de datos

#### MySQL

Es un gestor de base de datos sencillo de usar y rápido. También es uno de los motores de base de datos más usados en Internet.

Las características principales de MySQL son:

- *“Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente.*
- *Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.*
- *Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera.*
- *Es una base de datos muy rápida, segura y fácil de usar.”[16]*

### 1.8.7 Software Weka

Es una herramienta de aprendizaje automático y minería de datos (*data mining*), escrita en lenguaje Java. Ha logrado convertirse en uno de los software más popular en la comunidad Open Source no sólo por la cantidad de algoritmos implementados, sino por su facilidad para poder ser extendido o adaptado a cualquier entorno.

Las características son:

- ✓ *“Multiplataforma (Java), con 4 interfaces de usuario distintas.*
- ✓ *Diseño arquitectónico basado en componentes independientes, lo que facilita su uso de forma individual y su adaptación a cualquier entorno.*
- ✓ *Madurez y fiabilidad equiparables al software comercial de más alta gama.*
- ✓ *Licencia permisiva (GPL) y código fuente disponible.”[17]*

## 1.9 Conclusiones

- ✓ Para el desarrollo del sistema se utilizó la metodología RUP debido a que es la más adaptable para proyectos largos y permite entregar un software con calidad, al reducir los errores durante el tiempo de desarrollo.
- ✓ El lenguaje utilizado es Java por ser multiplataforma y la herramienta que se consideró más adecuada para la implementación es Netbeans por la rapidez de compilado y su fácil trabajo con ventanas.
- ✓ Para el almacenamiento y gestión de los datos que se almacenan se consideró utilizar MySql porque trabaja en múltiples plataformas, se puede acceder a él de forma gratuita y es de mucha calidad, rápido y confiable.
- ✓ Por el nivel de cálculo que se requiere se utiliza computación Grid, para agilizar el proceso de cálculo y reducir el tiempo de simulación.
- ✓ Para la resolución del modelo matemático se hace uso del asistente matemático Matlab debido a los métodos que tiene implementado.
- ✓ Se utiliza la herramienta computacional Weka para la creación de los clusters, debido a la cantidad de algoritmos de clustering que tiene implementado.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

En este capítulo se describen los casos de uso encontrados, se definen los actores y trabajadores de negocio y se brinda además una representación gráfica del mismo. Además se conocerán las reglas del negocio que fueron definidas para el software. Se hará una descripción del sistema a automatizar y se mostrará un diagrama de casos de uso del sistema para un mejor entendimiento del mismo. Se definirán los actores del sistema y los requerimientos funcionales del sistema que se implementó. Se plantearán los requerimientos mínimos que debe tener el sistema de cómputo donde se vaya a instalar la aplicación.

### **2.1 Modelo del negocio actual**

El investigador del Centro de Ingeniería Molecular realiza un estudio para definir el sistema biológico que va a simular y a estudiar rigurosamente. Luego de estar definido el sistema biológico se procede a elaborar el modelo matemático que describe al sistema biológico, una vez terminado este paso el bioinformático (es el mismo investigador, pero desempeñando otro rol) procede a realizar un programa para comenzar a simular a partir de dicho modelo con los parámetros y condiciones iniciales definidos. Terminada la simulación el investigador procede a realizar un análisis de la simulación, en caso de querer realizar otra simulación (con distintos parámetros y condiciones iniciales), pues se realiza y una vez obtenida dicha simulación la analiza para así finalizar el estudio de un sistema biológico.

### **2.2 Reglas del negocio**

Reglas del negocio a considerar

#### 1. Reglas del Modelo de Datos:

- Se deben realizar previamente las simulaciones para hacer el análisis.
- Los valores de tiempo inicial, final y los valores de condiciones iniciales y parámetros que se insertan en el programa correspondiente al modelo matemático tienen que ser números para poder obtener la simulación.

#### 2. Reglas de derivación:

- Las simulaciones que le resulten interesantes al investigador con sus correspondientes datos, serán almacenados en un fichero para su posterior consulta.

#### 3. Reglas de Restricción:

- En cada una de las máquinas que se vayan a utilizar para la simulación, tiene que estar instalado el asistente matemático Mathematica.

#### 4. Reglas de flujo:

- El investigador le entrega al bioinformático un cierto sistema biológico y su correspondiente modelo matemático.
- Se realizan pruebas para comprobar que el modelo matemático describe correctamente al sistema biológico, si no corresponde, se vuelve a definir el modelo matemático.
- Cuando se tiene el modelo matemático correcto el bioinformático realiza un programa para dicho modelo en la herramienta computacional Mathematica.
- Se obtienen las simulaciones a partir de los parámetros y condiciones iniciales entrados en el programa.
- Luego el investigador realiza un análisis de las simulaciones y define cuáles le resultan interesantes, guarda dichos resultados y sus correspondientes valores de condiciones iniciales y parámetros en un fichero para consultarlos cuando lo desee.

### 2.3 Actores del negocio

**Tabla 1. Descripción de los actores del negocio**

Nombre del actor	Descripción
Investigador	Es el encargado de definir el sistema biológico al cual realizar el estudio y el modelo matemático que lo define para luego realizar simulaciones. Lo hace con el fin de estudiar a fondo un sistema biológico para conocer su comportamiento.

## 2.4 Diagrama de casos de uso del negocio

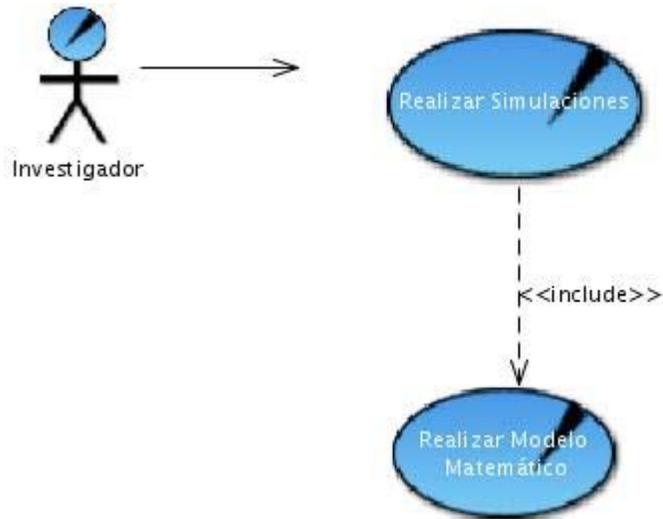


Fig. 2.1. Diagrama de casos de uso del negocio.

## 2.5 Trabajadores del negocio

Tabla 2. Descripción de los trabajadores del negocio

Nombre del trabajador	Descripción
Bioinformático	Es el encargado de hacer un programa para resolver un modelo matemático, realizando así las simulaciones que se necesiten. Lo hace con el fin de obtener simulaciones para un posterior estudio de las mismas.

## 2.6 Descripciones de Casos de uso del negocio

### Realizar simulaciones

<b>Nombre del CU</b>	<b>Realizar Simulaciones</b>	
<b>Actores</b>	Investigador	
<b>Propósito</b>	Permite al investigador hacer un estudio riguroso acerca del comportamiento de un sistema biológico.	
<b>Resumen</b>	El caso de uso se inicia cuando el investigador selecciona un sistema biológico interesante para estudiar detenidamente haciendo simulaciones a partir de un modelo matemático definido previamente, el caso de uso termina cuando el investigador decide no realizar más simulaciones.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del negocio</b>	
1. Elabora el modelo matemático que lo describe (Ver Caso de Uso Realizar Modelo Matemático).	.	
2. Define los valores de los parámetros y de las condiciones iniciales.	3. El bioinformático entra dichos valores en el programa que resuelve el modelo matemático.	
	4. El bioinformático realiza las simulaciones.	
	5. El bioinformático obtiene los resultados de las simulaciones.	
6. El investigador define los tipos de análisis a realizarle a los resultados obtenidos.	7. El bioinformático obtiene los resultados del análisis según el tipo que se definió.	
8. El investigador estudia el resultado del análisis de las simulaciones y le interesa almacenar el resultado de las mismas.	9. El bioinformático almacena los valores de los parámetros y de las condiciones iniciales y el resultado de las simulaciones en un fichero.	
10. El investigador decide no realizar más simulaciones.	11. Termina el caso de uso.	

<b>Curso Alterno</b>	
<b>Acciones del Actor</b>	<b>Respuesta del negocio</b>
8. Al investigador no le interesa el resultado de las simulaciones, pasando así a la acción 2.	
10. El investigador decide realizar más simulaciones, pasando así a la acción 2.	

### **Realizar Modelo Matemático**

<b>Nombre del CUS</b>	<b>Realizar Modelo Matemático</b>
<b>Actores</b>	Investigador
<b>Propósito</b>	Permite al investigador realizar un modelo matemático y comprobar que se ajusta al sistema biológico en cuestión.
<b>Resumen</b>	El caso de uso se inicia cuando el investigador selecciona un sistema biológico interesante para estudiar detenidamente, para el cual realiza un modelo matemático que lo describa, resuelve dicho modelo para ver que se ajusta al sistema biológico, comprobando que el modelo obtenido es el que lo define.

### **Curso Normal de los Eventos**

<b>Acciones del Actor</b>	<b>Respuesta del negocio</b>
1. El investigador define el sistema biológico a estudiar.	
2. Define los parámetros y condiciones iniciales del sistema.	
3. Realiza el sistema de ecuaciones diferenciales ordinarias.	4. El bioinformático realiza un programa para resolver dicho modelo matemático.
5. Define valores de parámetros y condiciones iniciales para probar que el modelo se ajusta al sistema.	6. Resuelve el modelo con dichos datos.

	7. El bioinformático obtiene el resultado del modelo.
8. Analiza el resultado obtenido y comprueba que el modelo es el adecuado.	
	9. Termina el caso de uso.
<b>Curso Alterno</b>	
<b>Acciones del Actor</b>	<b>Respuesta del negocio</b>
8. El modelo obtenido no es el adecuado pasando así a la acción 3.	

## 2.7 Diagramas de Actividades

### Caso de uso <Realizar Simulaciones>

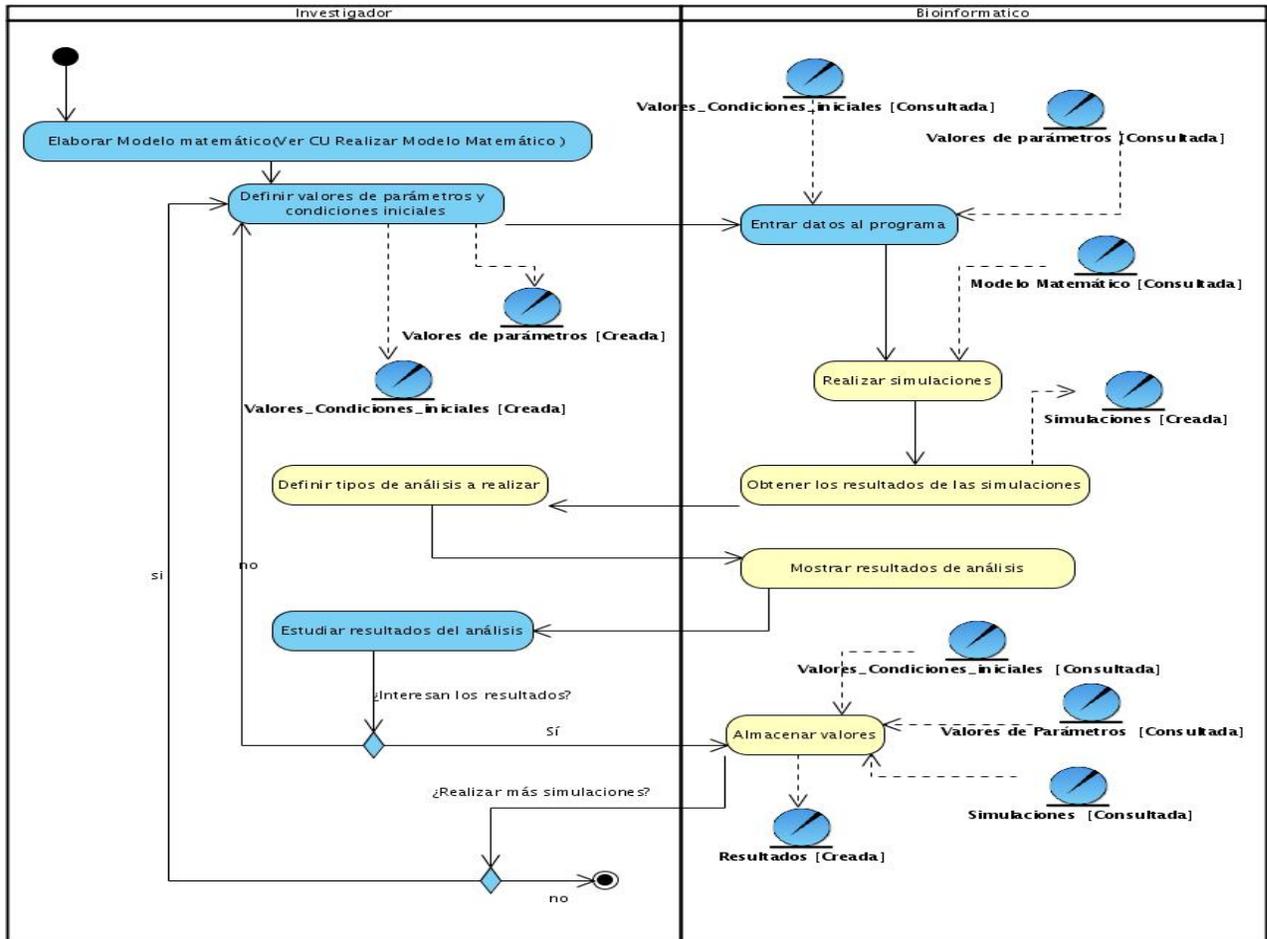
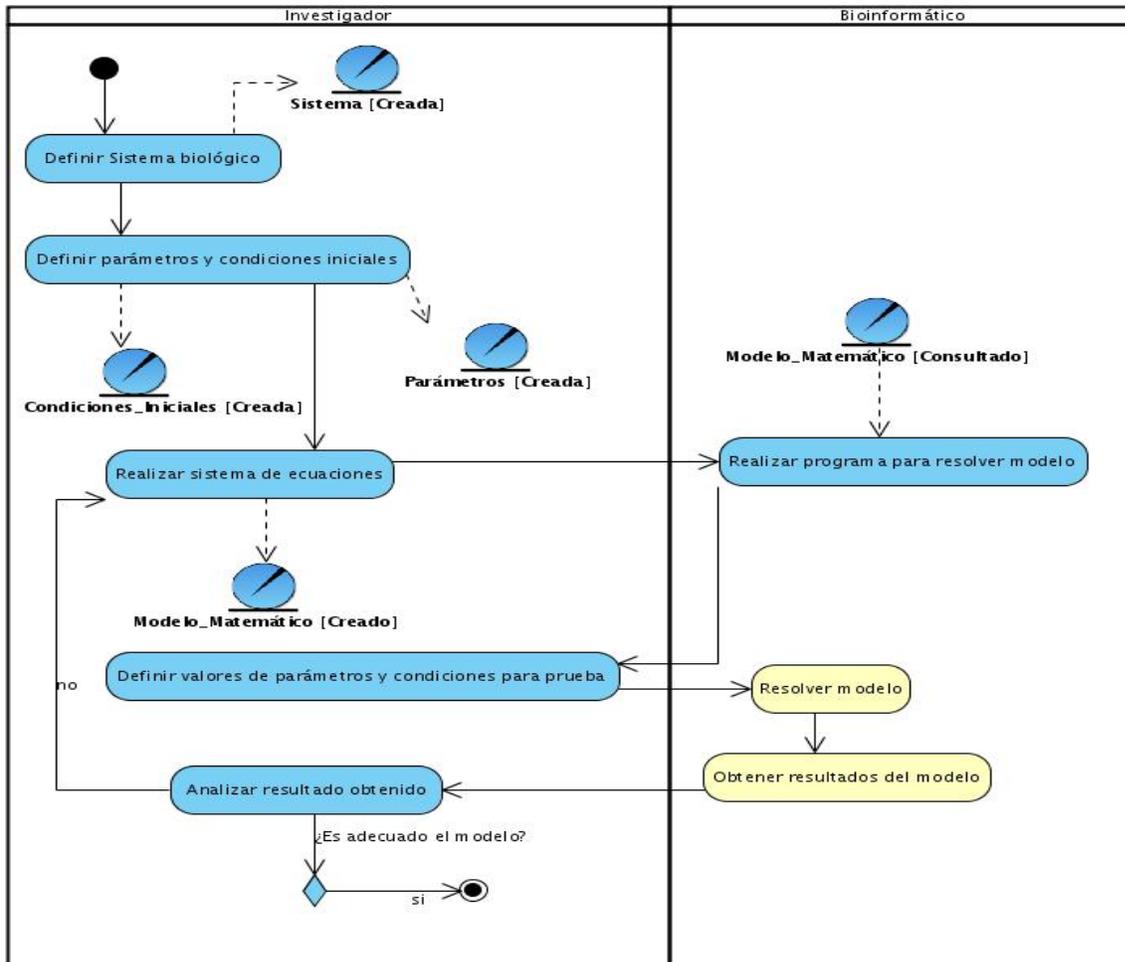


Fig. 2.2. Diagrama de actividades del caso de uso <Realizar Simulaciones>.

**Caso de uso <Realizar Modelo Matemático>**



**Fig. 2.3 Diagrama de actividades del caso de uso <Gestionar Modelo Matemático>.**

## 2.8 Modelo de objetos

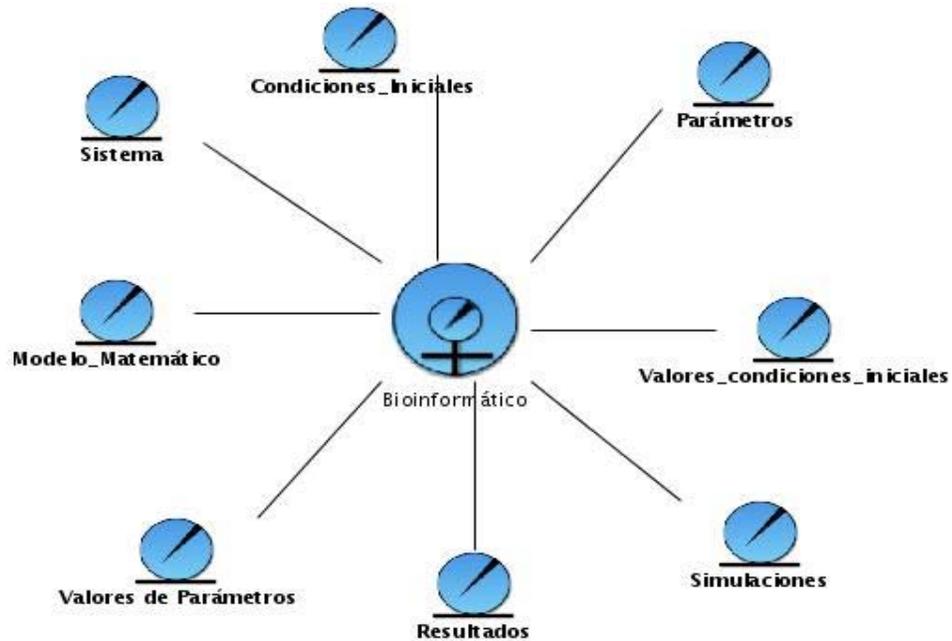


Fig. 2.4. Modelo de objetos

## 2.9 Actores del sistema a automatizar

Los actores de un sistema pueden ser otros sistemas o hardware externo que se relacionan o interactúan con dicho sistema, no necesariamente tiene que ser una persona. Cada actor juega un rol determinado al interactuar con el sistema y diferentes usuarios pueden asumir el mismo rol de un actor. Luego de definir el concepto de actores del sistema se puede establecer el o los actores del sistema en cuestión.

**Tabla 3. Definición de actores del sistema a automatizar**

Nombre del actor	Descripción
Bionfornático	El bioinformático es aquella persona que va a interactuar con el sistema para realizar las simulaciones y luego hacer el análisis de los resultados.

### 2.10 Requisitos Funcionales

Declaración de los servicios que el sistema debe proporcionar, Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requisitos funcionales del sistema son:

R1- Gestionar la información del sistema biológico que se esté estudiando.

- R1.1- Insertar sistema biológico a estudiar.
- R1.2- Seleccionar sistema biológico existente.
- R1.3- Eliminar un sistema biológico existente.
- R1.4- Modificar sistema biológico existente.

R2- Gestionar modelo matemático.

- R2.1- Insertar el modelo matemático.
- R2.2- Eliminar el modelo matemático.
- R2.3- Modificar el modelo matemático.

R3- Realizar estudios de la dinámica del sistema.

- R3.1- Mostrar gráficas de población vs población.
- R3.2- Mostrar gráficas de población vs tiempo.

R4- Realizar simulaciones múltiples utilizando tecnología Grid.

R5- Analizar los resultados de las simulaciones utilizando algoritmos de clustering.

- R5.1- Agrupar utilizando algoritmos de clustering de particionado.
- R5.2- Agrupar utilizando el algoritmo de clustering Jerárquicos.
- R5.3- Permitir mostrar la gráfica de dinámica de población de una simulación que esté contenida en uno de los clusters.

R6- Analizar los resultados de las simulaciones utilizando reglas definidas por el usuario.

- R6.1- Mostrar la gráfica de las fronteras de las reglas entre dos parámetros que varían.

R7- Editar la configuración de las conexiones para la simulación distribuida.

R8- Autenticar usuario para acceder a la aplicación.

## 2.11 Diagrama de casos de uso del sistema a automatizar

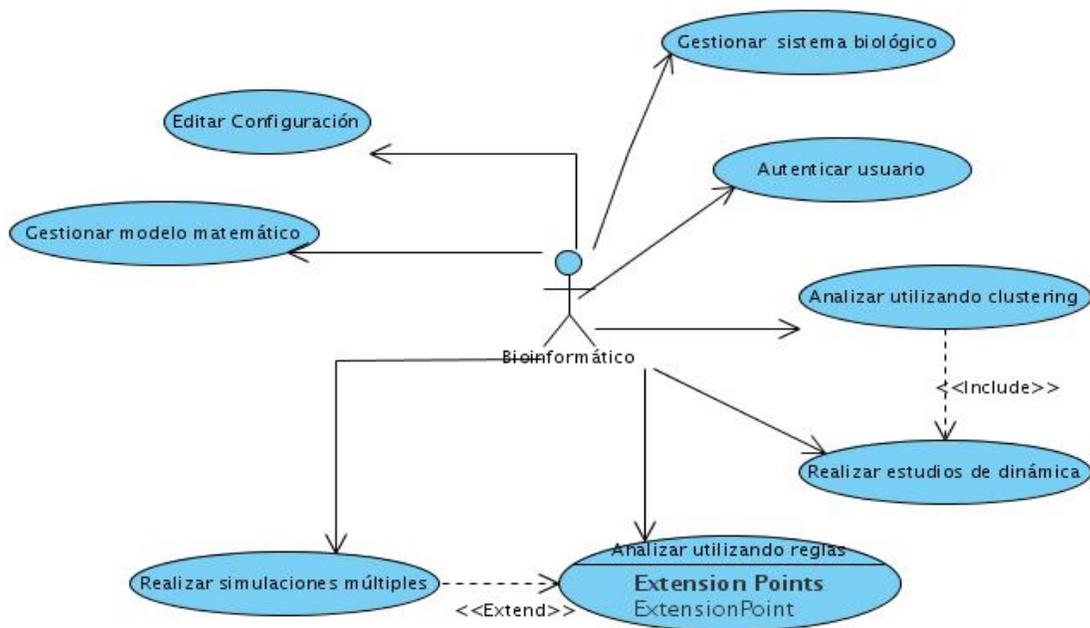


Fig. 2.5. Diagrama de casos de uso del sistema.

## 2.12 Requisitos no funcionales

Restricciones que afectan a los servicios o funciones del sistema, tales como restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son las que hacen al producto atractivo, usable, rápido o confiable.

### Requisitos no funcionales

- Apariencia o interfaz externa
  - ✓ La aplicación deberá tener una interfaz externa amigable, que sea sencilla y fácil de entender por el usuario para así evitar que el usuario se pierda dentro de la aplicación.
- Software:
  - ✓ No hay restricciones en cuanto al sistema operativo a instalar en la PC que esté instalada la aplicación, puesto que la misma será multiplataforma.
  - ✓ La máquina virtual de Java 1.5 tiene que estar instalada.
  - ✓ En cada una de las máquinas clientes que se utilicen para realizar las simulaciones distribuidas deberán tener instalado cualquier distribución de Linux como sistema operativo.
  - ✓ El asistente matemático Matlab tiene que estar instalado en cada una de las máquinas que se vaya a utilizar como clientes de la plataforma distribuida.
  - ✓ El cliente de la Grid deberá estar instalado en cada una de las PCs que se utilizarán para la simulación distribuida.
- Hardware:
  - ✓ Se debe contar con 256 MB de memoria RAM como mínimo, aunque lo ideal serían 512 MB.
  - ✓ Procesadores Pentium IV.
  - ✓ Disco duro de 20 GB como mínimo (depende de la cantidad de información a almacenar).
- Requerimientos en el diseño de implementación.
  - ✓ Lenguaje de programación Java.

- ✓ Netbeans como IDE de desarrollo.
- ✓ Visual Paradigm como herramienta CASE.
- ✓ MySQL como gestor de bases de datos.
- Seguridad
  - ✓ Confidencialidad: Se requiere de usuario y contraseña para poder acceder a la información de la base de datos.
  - ✓ Se requiere de usuario y contraseña para poder acceder al servidor de la Grid.
  - ✓ Disponibilidad: En caso de tener el usuario y la contraseña se le garantiza poder acceder a la información almacenada en todo momento.
- Usabilidad:
  - ✓ El sistema le ofrecerá al investigador la posibilidad de realizar simulaciones distribuidas y análisis de los resultados de las simulaciones, en este sentido se centra el diseño de la aplicación y en específico de las interfaces que harán posible el intercambio de datos de manera que le resulte al usuario de fácil entendimiento.
- Rendimiento: Para hacer más rápida la obtención de los resultados de las simulaciones se implementará una aplicación que haga uso del cálculo distribuido de las mismas para agilizar el proceso de obtención de dichos resultados.
- Soporte:
  - ✓ 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.

- Portabilidad:
  - ✓ El sistema deberá funcionar en cualquier sistema operativo sobre el cual se haya instalado la máquina virtual de Java 1.5 o superior.
- Confiabilidad:
  - ✓ El sistema no debe permitir que existan fallos, pero de ocurrir se debe garantizar que la pérdida de información sea mínima.
  - ✓ De ocurrir algún fallo en las conexiones para realizar los cálculos distribuidos, el sistema está diseñado para detectar este error y enviar la parte del trabajo que no se concluyó a otra máquina que esté disponible para que esta la continúe.

### **2.13 Casos de uso del sistema**

1. Gestionar sistema biológico.
2. Gestionar modelo matemático.
3. Realizar estudios de dinámica.
4. Realizar simulaciones múltiples.
5. Analizar utilizando clustering.
6. Analizar utilizando reglas.
7. Editar configuración.
8. Autenticar usuario en BD.

## 2.14 Descripción de los casos de uso del sistema.

### Descripción Caso de Uso Realizar simulaciones múltiples

<b>Nombre del CU</b>	<b>Realizar simulaciones múltiples</b>	
<b>Actores</b>	Bioinformático	
<b>Propósito</b>	Resolver un modelo matemático que describa a un sistema biológico.	
<b>Resumen</b>	El bioinformático desea resolver un modelo matemático dado un sistema biológico para ello establece una conexión con la base de datos de la cual va a tomar los datos necesarios y entra los valores de los parámetros, de las condiciones iniciales, así como el rango de tiempo en el cual quiere analizar el comportamiento del sistema biológico, selecciona el método numérico para resolver el modelo.	
<b>Referencias</b>	R4	
<b>Precondiciones</b>	Tener conexión con la Base de Datos y estar autenticado en la misma previamente.	
<b>Poscondiciones</b>	Almacenar los resultados en la BD.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El bioinformático selecciona el sistema biológico y el modelo matemático que desea simular, luego selecciona la interfaz que le permite obtener las simulaciones.	2. El sistema muestra la interfaz correspondiente.	
3. El bioinformático entra el rango de valores en los que variarán las condiciones iniciales y los parámetros, así como la escala a utilizar en la variación de cada uno de ellos, además el rango de tiempo en el que va a realizar las simulaciones, así como el método numérico a utilizar.	4. Genera todas las combinaciones posibles entre los valores de las condiciones iniciales y los parámetros.	

	5. Verificar en la BD que las simulaciones no estén realizadas.
	6. Si no se han hecho esas simulaciones con anterioridad, guarda en un fichero todas las combinaciones que se vayan a realizar para distribuirlas por la Grid.
	7. El sistema muestra una ventana para autenticarse en la Grid.
	8. El sistema realiza las simulaciones y las inserta en la BD.
	9. Termina el caso de uso.
<b>Curso Alternativo de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	5. La aplicación busca en la BD y encuentra que ya se han realizado las simulaciones por lo que no simula ese modelo matemático.
<b>Prioridad:</b>	Crítico

- ✓ Descripción Caso de Uso Autenticar usuario. [Anexo 1].
- ✓ Descripción Caso de Uso Realizar estudios de dinámica. [Anexo 2].
- ✓ Descripción Caso de Uso Analizar utilizando reglas. [Anexo 3].
- ✓ Descripción Caso de Uso Gestionar sistema biológico. [Anexo 4].
- ✓ Descripción Caso de Uso Gestionar modelo matemático. [Anexo 5].
- ✓ Descripción Caso de Uso Analizar utilizando clustering. [Anexo 6].
- ✓ Descripción Caso de Uso Editar Configuración. [Anexo 7].

## **2.15 Conclusiones**

En este capítulo se hizo un estudio de los procesos que se llevan a cabo en el CIM para realizar las simulaciones biológicas, procesos que fueron representados en un modelo de negocio. Se evidenció la necesidad de implementar una herramienta que facilite la ejecución de dichos procesos, para lo cual se obtuvo las actividades a automatizar y el flujo de información en cada una de las actividades descritas. Se definió el actor que interactúa con el sistema. Se tuvieron en cuenta además, las funcionalidades que debe tener el software y las cualidades del mismo planteadas como requisitos funcionales y requisitos no funcionales respectivamente. Se hizo una descripción de alto nivel de cada uno de los casos de uso contemplados en el diagrama de casos de uso del sistema.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se realiza el diagrama de clases del diseño dando respuesta a la solución que se propone, este es el artefacto principal del Flujo de Trabajo Análisis y Diseño de RUP. Se describen los principios de diseño que se tuvieron en cuenta durante el desarrollo de la aplicación. Se describe además el estilo arquitectónico a utilizar y se define el diagrama de despliegue con el objetivo de mostrar la distribución física de los nodos de cómputo que necesita la aplicación.

### 3.1 Diagrama de clases del Análisis.

- ✓ **Caso de uso Analizar utilizando clustering. [Anexo 8].**
- ✓ **Caso de uso Analizar utilizando reglas. [Anexo 9].**
- ✓ **Caso de uso Autenticar usuario. [Anexo 10].**
- ✓ **Caso de uso Gestionar modelo matemático. [Anexo 11].**
- ✓ **Caso de uso Gestionar sistema biológico. [Anexo 12].**
- ✓ **Caso de uso Realizar estudios de dinámica. [Anexo 13].**
- ✓ **Caso de uso Editar Configuración. [Anexo 14].**
- ✓ **Caso de uso Realizar simulaciones múltiples. [Anexo 15].**

### 3.2 Estilo arquitectónico utilizado

Arquitectura en capas: Se utilizó la arquitectura en capas debido a que organiza el modelo de diseño a través de capas que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Se utilizaron tres niveles (es decir, tres capas). Las mismas se mencionan a continuación:

- ✓ **Capa de presentación:** representa las interfaces de la aplicación.

- ✓ Capa de acceso a datos: representa las clases controladoras y entidades del negocio.
- ✓ Capa de datos: representa la base de datos donde esté almacenada la información del sistema.

### 3.3 Patrones de diseño utilizados

#### 3.3.1 ¿Qué son los Patrones de Diseño?

"Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma". [18]

#### 3.3.2 Principales Patrones de diseño utilizados.

##### 3.3.2.1 Patrón Adaptador

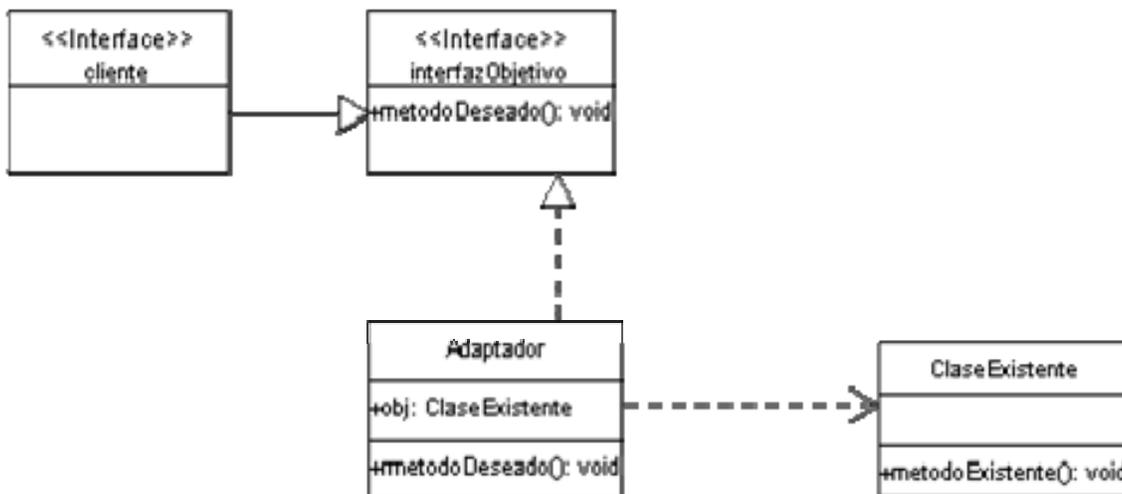
- **Intención**

"Convierte la interfaz de una clase en otra más compatible con nuestras necesidades." [18]

- **Aplicaciones**

- ✓ "Para utilizar la interfaz de una librería que no coincide con la que se requiere.
- ✓ Para extender la funcionalidad de una librería existente." [18]

- **Estructura**



- **Participantes**

- ✓ *“Cliente: Clase que hace uso de otras clases a través de una interfaz que no tiene por qué corresponder con el implementado en dichas clases.*
- ✓ *InterfazObjetivo: Clase que declara la interfaz a la que llama la clase cliente.*
- ✓ *Adaptador: Clase que implementa el enlace entre la interfaz deseada (u objetivo) y la que realmente existe (ofrecida por la ClaseExistente)*
- ✓ *ClaseExistente: También llamada adaptada, es la clase original de que se dispone.”[18]*

- **Patrones relacionados**

*“Strategy: También es similar a Adaptor, pero Strategy además pretende cambiar la funcionalidad del método llamado.” [18]*

### 3.3.2.2 Patrón Vista/Controlador

*“El controlador tiene la responsabilidad de gestionar un sistema de mensajes.” [18]*

- **Motivo**

*“El modelo vista-controlador se basa en delegar en una clase controlador la atención de los eventos y la modificación de los datos y a su vez se delega en una clase vista la representación de dicha información, es decir la actualización de los cambios en el sistema de representación (sistema de ventanas habitualmente).” [18]*

- **Aplicaciones**

- ✓ *“Se utiliza sobre todo en entornos gráficos de usuario interactivos, ya que el número y posibilidades de mensajes, así, como el elevado número de objetos visuales dificultan la gestión de la información y las operaciones sobre ellos.*
- ✓ *El modelo Vista-controlador es útil también cuando es necesario proporcionar múltiples representaciones de los mismos datos, en este contexto el controlador gestiona la información, y múltiples clases Vista ofrecen las diferentes representaciones de la misma información, por ejemplo diferentes diagramas (de barras, de tarta, etc.) de una aplicación estadística u hoja de cálculo.” [18]*

### 3.3.2.3 Patrón Fachada - Facade

- **Intención**

*“Simplifica el acceso a un conjunto de clases o interfaces.” [18]*

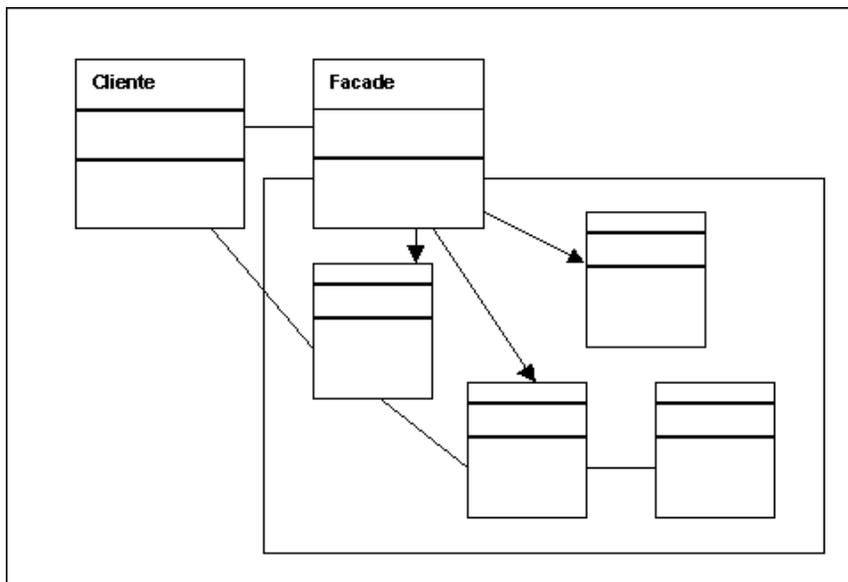
- **Motivo**

*“Reducir la dependencia entre clases, Facade ofrece un punto de acceso al resto de las clases, si estas cambian o se sustituyen por otras sólo hay que actualizar la clase Facade sin que el cambio afecte a las aplicaciones clientes. Facade no oculta las clases sino que ofrece una forma más sencilla de acceder a ellas, en los casos en que se requiere se puede acceder directamente a ellas.” [18]*

- **Aplicaciones**

- ✓ *“Para ofrecer un acceso sencillo a subsistemas complejos.*
- ✓ *Para descomponer un sistema en capas, utilizando Facade para ofrecer un interfaz de acceso entre las mismas” [18]*

- **Estructura**



### 3.4 Diagrama de clases del Diseño.

- ✓ **Caso de uso Analizar utilizando clustering. [Anexo 16].**
- ✓ **Caso de uso Analizar utilizando reglas. [Anexo 17].**
- ✓ **Caso de uso Autenticar usuario. [Anexo 18].**
- ✓ **Caso de uso Gestionar modelo matemático. [Anexo 19].**
- ✓ **Caso de uso Gestionar sistema biológico. [Anexo 20].**
- ✓ **Caso de uso Realizar estudios de dinámica. [Anexo 21].**
- ✓ **Caso de uso Editar Configuración. [Anexo 22].**
- ✓ **Caso de uso Realizar simulaciones múltiples. [Anexo 23].**

### 3.5 Principios de diseño

#### 3.5.1 Interfaz de usuario

La herramienta computacional diseñada es una aplicación desktop, por lo que la interfaz diseñada para el sistema está basada en el estándar de ventanas. El tipo de letra utilizado es *Dialog*, tamaño 12 estilo *Plain*. El diseño de la aplicación es sencillo y simple de utilizar. Todos los botones tienen una altura de 25 y el ancho variará de acuerdo al contenido del mismo. El sistema muestra una barra de menú en la parte superior donde están la mayoría de las funciones a realizar, además un explorador en la parte izquierda donde estarán los sistemas y modelos contenidos en la Base de Datos, que al seleccionarlos se podrán realizar los estudios deseados. El icono asociado a la aplicación será el logotipo que identifica al sistema.

#### 3.5.2 Tratamiento de errores

En el diseño de la aplicación se tuvo en cuenta el tratamiento de errores, mostrando mensajes de fácil comprensión y lo más descriptivos posibles para una mejor información al usuario. Además se les alerta de posibles riesgos que puedan ocurrir con las operaciones que el usuario realice. Para los mismos se utilizan los iconos correspondientes a los estándares, ejemplo:



Fig. 3.1. Icono de información.



Fig. 3.2. Icono de error.



Fig. 3.3. Icono de confirmación.

### 3.6 Modelo de clases persistentes

Las clases persistentes representan información de larga duración o que persiste en el tiempo, es decir, es la información que se requiere almacenar para gestionarla en cualquier momento. En este caso las clases persistentes se convierten en tablas de la base de datos, lo cual facilitará el manejo de la información y permitirá tener un registro completo del estudio de sistemas biológicos.

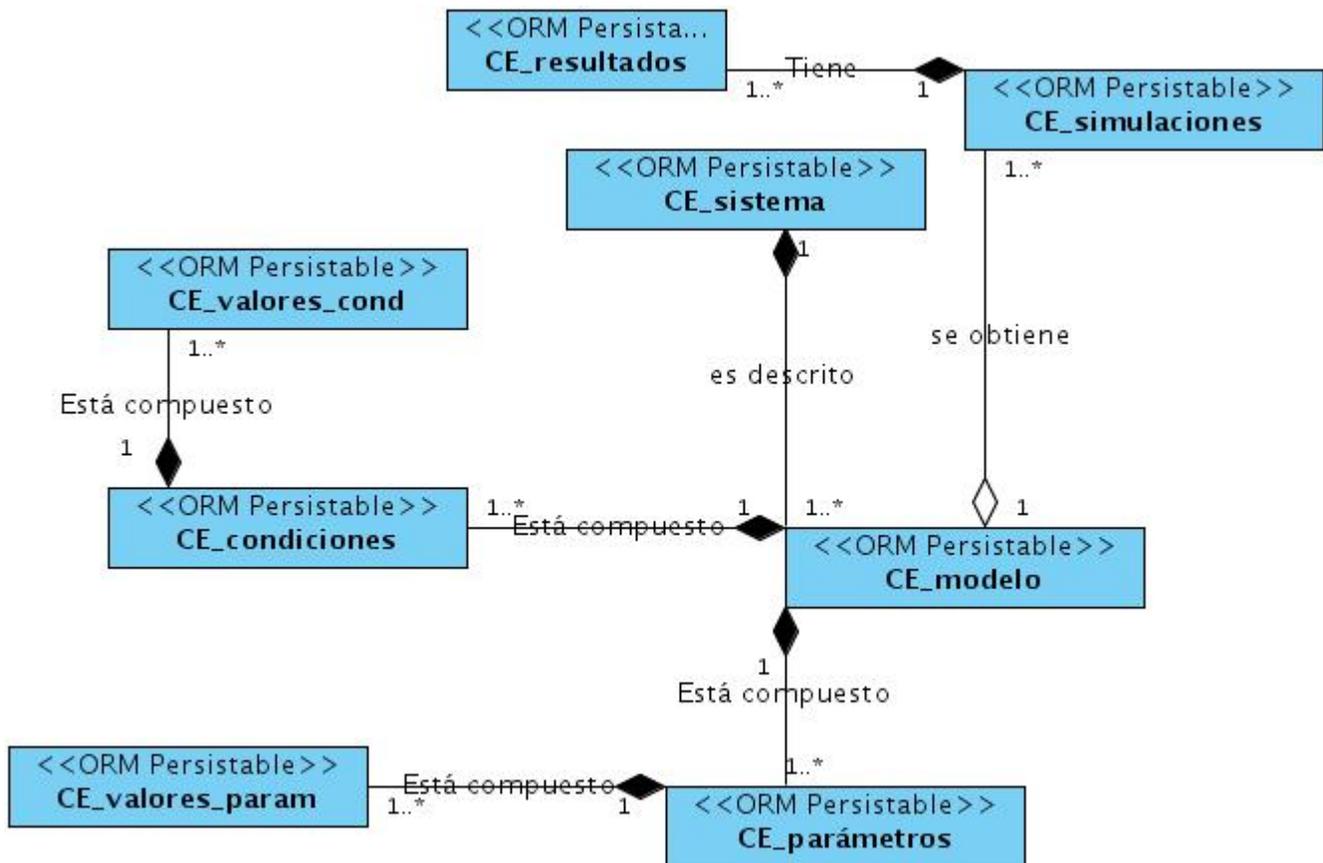


Fig. 3.4. Diagrama de clases.



### 3.8 Modelo físico de datos

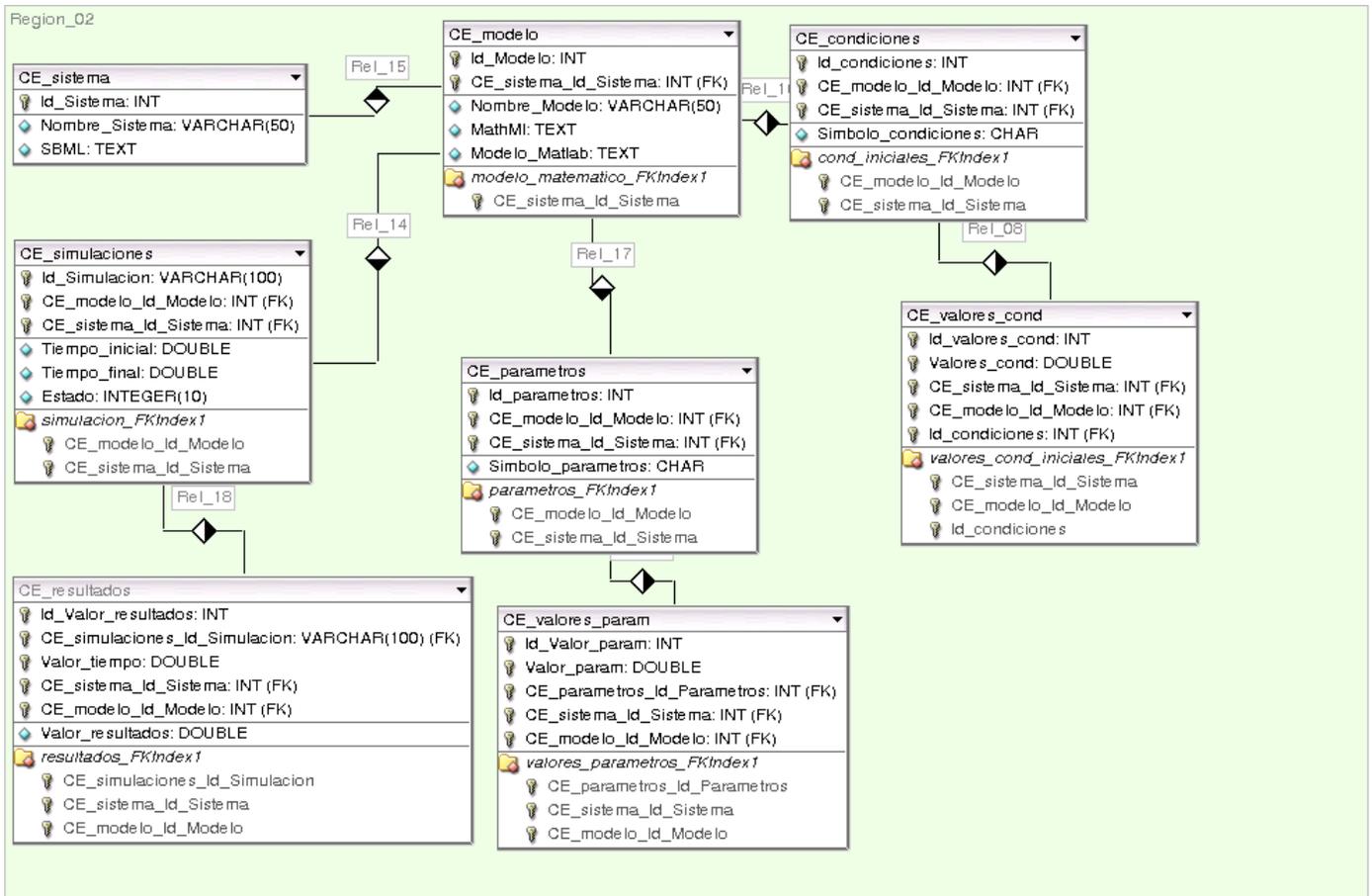
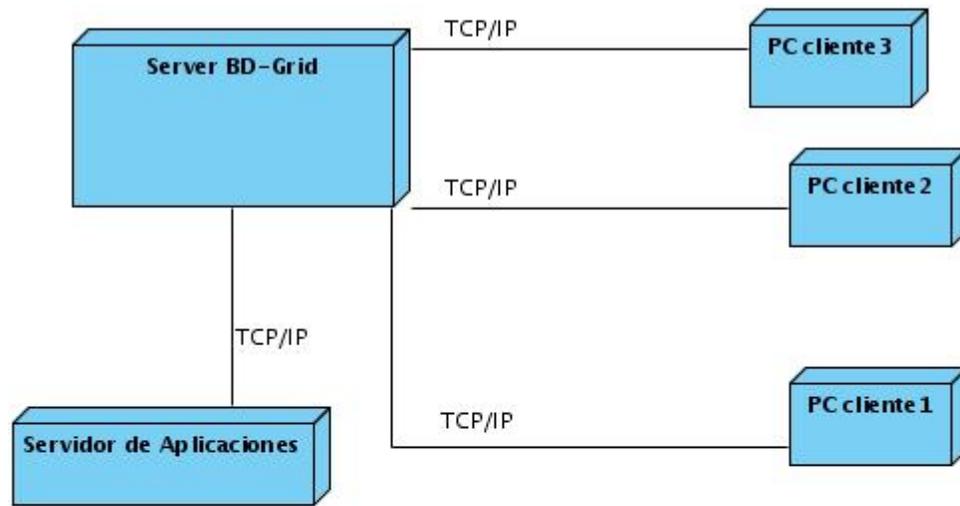


Fig. 3.6. Modelo físico de datos.

### 3.9 Diagrama de Despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad del mismo entre los nodos de cómputo que utiliza. No es más que la representación física de todos los recursos que utiliza el sistema, por ejemplo, nodos con capacidad de procesamiento, módems e impresoras.



En este diagrama se representa la distribución física de los elementos hardware. Están representadas 3 PCs clientes para que se sepa que el sistema es distribuido, pero en realidad no se conoce la cantidad de PCs clientes que se utilizarán. Cada Pc cliente debe tener instalado la máquina virtual de java 1.5, Matlab 7.0 y un cliente de la Grid. En el servidor de aplicaciones se instala el software. En el Server BD-Grid se encuentran la base de datos y el servidor de la Grid pero no necesariamente tienen que estar en un mismo servidor (pueden estar aislados).

**Fig. 3.7. Diagrama de despliegue.**

### **3.10 Conclusiones**

Como resultado de este capítulo se conoce cómo fue implementado el software, partiendo de las clases del diseño definidas, así como también, el diseño que tiene la BD. Se conoció cómo se llevan a cabo las funcionalidades que requieren el sistema que se desarrolló y la distribución física de los nodos de procesamiento que se necesitan para su correcto funcionamiento y mejor rendimiento.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

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

### 4.1 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

#### Estereotipos de los componentes

- ✓ Ejecutable: Es un programa que se puede ejecutar en un nodo.
- ✓ Biblioteca: Es una biblioteca de objetos estática o dinámica.
- ✓ Tabla: Es una tabla de una BD.
- ✓ Archivo: Es un fichero que contiene código fuente.
- ✓ Documento: Es un documento.
- ✓ Pagina Web: Es una página que se obtiene de la ejecución del sistema.

A continuación se muestra el diagrama de componentes de la aplicación software en cuestión:

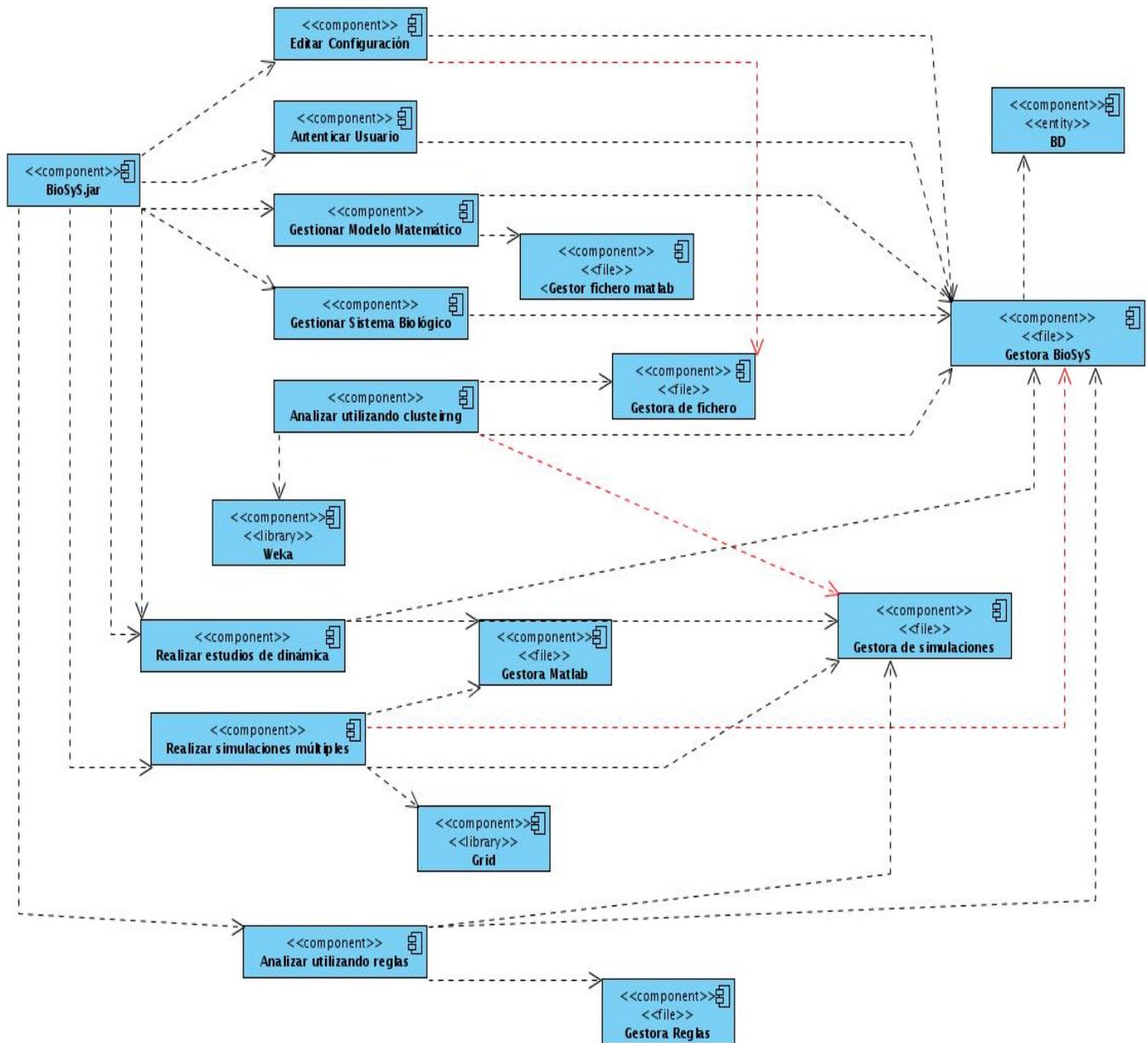


Fig. 4.1. Diagrama de componentes.

## 4.2 Casos de prueba de caja negra

Como la aplicación desarrollada es un módulo de una plataforma computacional, el tipo de prueba realizado es el de Prueba de Unidades. El objetivo de realizar las pruebas a la aplicación es comprobar que esta cumpla correctamente los requisitos funcionales establecidos, se utilizó el método de caja negra, con lo cual se verificaron no sólo que las funciones del software son operativas, sino también que no existieran errores en las interfaces. La técnica utilizada fue la técnica algebraica que se conoce como clases de equivalencia. Con dicha técnica se definen datos que son equivalentes a otros y si se logra partir un rango excesivamente amplio de posibles valores reales a un conjunto reducido de clases de equivalencia, entonces es suficiente probar un caso de cada clase, pues los demás datos de la misma clase son equivalentes.

### CU Realizar simulaciones múltiples

	Casos Válidos	Casos no Válidos
Variar Population Initial value	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Valor cero.</li> </ul>
Variar Population Final value	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Valor cero.</li> <li>El valor final deberá ser mayor que el valor inicial entrado.</li> </ul>
Initial Time	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Sin valor.</li> </ul>
Final Time	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Sin valor.</li> </ul>

**Caso de uso:** Realizar simulaciones múltiples

**Caso de prueba:** Se entran los valores en los que variará cierto parámetro, el valor final entrado será menor que el valor inicial.

**Entrada:** Se selecciona el sistema y el modelo matemático que se desea simular. Se entran del parámetro que se desea variar, los valores inicial y final. Se entran además el rango de variación de las condiciones iniciales, el rango de tiempo y el método matemático.

System=Interleuquina

Mathematical Model=dy2\_eval

**De Condiciones Iniciales se variarán:**

Population x initial value=1 final value=4 Points Number=1

Population y initial value=1 final value=8 Points Number=1

Population z initial value=2 final value=15

**De parámetros se variará c:**

Population c initial value=8 final value=2 Points Number=2

Initial Time= 0

Final Time= 10

Mathematical Method= ode45

**Resultado:** Se mostrará un mensaje de error diciendo que el valor final del parámetro que se desea variar debe ser mayor que el valor inicial entrado.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Realizar simulaciones múltiples

**Caso de prueba:** Se entran los valores en los que variará cierto parámetro.

**Entrada:** Se selecciona el sistema y el modelo matemático que se desea simular, se entran del parámetro que desea variar, los valores inicial y final. Se entran además el rango de variación de las condiciones iniciales, el rango de tiempo y el método matemático.

System=Interleuquina

Mathematical Model=dy2\_eval

**De Condiciones Iniciales se variarán:**

Population x initial value=1 final value=12 Points Number=1

Population y initial value=1 final value=8 Points Number=1

Population z initial value=2 final value=5 Points Number=1

**De parámetros se variará c:**

Population c initial value=8 final value=16 Poits Number=2

Initial Time= 1

Final Time= 10

Mathematical Method= ode45

User=administrator

Pass=cassandra

Server=10.34.30.30

**Resultado:** Se realizan todas las simulaciones después de haber definido las máquinas que realizarán el trabajo distribuido.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

## CU Autenticar usuario

	Casos Válidos	Casos no Válidos
Data Base Server	<ul style="list-style-type: none"> <li>• Formato de la expresión regular.</li> </ul>	<ul style="list-style-type: none"> <li>• No tener el formato de la expresión regular.</li> <li>• Sin valor.</li> </ul>
Port	<ul style="list-style-type: none"> <li>• Número.</li> </ul>	<ul style="list-style-type: none"> <li>• Letras.</li> <li>• Sin valor.</li> </ul>
Data Base	<ul style="list-style-type: none"> <li>• Cadena de caracteres.</li> </ul>	<ul style="list-style-type: none"> <li>• Sin valor.</li> </ul>
user	3 Cadena de caracteres.	4 Sin valor.
password	5 Cadena de caracteres.	6 Letras. 7 Sin valor.

**Caso de uso:** Autenticar usuario

**Caso de prueba:** Un usuario que no tiene permiso para acceder a la BD intenta autenticarse.

**Entrada:** Se entra el usuario y la contraseña. Se entran además el IP del servidor donde se encuentra la BD, el puerto y el nombre de la BD.

Data Base Server=10.34.20.25

Port=5801

Data Base=biosys

user= carlos

password= laberinto12

**Resultado:** Se muestra un mensaje indicando que no se pudo realizar la conexión con la BD.

**Condiciones:** Ser usuario de la BD.

<b>Caso de uso:</b> Autenticar usuario
<b>Caso de prueba:</b> Un usuario de la BD desea entrar al sistema y para ello se autentifica.
<p><b>Entrada:</b> Se entra el usuario y la contraseña. Se entran además el IP del servidor donde se encuentra la BD, el puerto y el nombre de la BD.</p> <p>Data Base Server=10.34.20.25  Port=5801  Data Base=biosys  user= proyecto  password= proyecto</p>
<b>Resultado:</b> El usuario accede a la aplicación de software.
<b>Condiciones:</b> Ser usuario de la BD.

### CU Realizar estudios de dinámica

	Casos Válidos	Casos no Válidos
Population value	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Sin valor.</li> </ul>
Parameters value	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Sin valor.</li> </ul>
Initial Time	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Sin valor.</li> </ul>
Final Time	<ul style="list-style-type: none"> <li>Números.</li> </ul>	<ul style="list-style-type: none"> <li>Letras.</li> <li>Sin valor.</li> </ul>

**Caso de uso:** Realizar estudios de dinámica

**Caso de prueba:** Se entrará una letra para uno de los valores de los parámetros. Se entrarán además los valores de las condiciones iniciales y los tiempos inicial y final.

**Entrada:** Se selecciona el sistema y el modelo matemático. Se entran valores de parámetros y condiciones iniciales, se entran además los valores de tiempo inicial y final. Se elige el método matemático a utilizar.

System=Interleuquina

Mathematical Model=dy2\_eval

Population x value=1

Population y value=1

Population z value=1

Parameters a value=1

Parameters b value=1

Parameters c value=1

Parameters d value=R

Parameters e=0

Parameters f value=0

Parameters g value=0

Parameters h value=0

Parameters i value=0

Parameters k value=0

Parameters l value=0

Initial Time=1

Final Time=10

Mathematical Method= ode45

**Resultado:** El campo del valor del parámetro al que se le entró una letra se señalará en rojo y el investigador no podrá variar o entrar más valores del parámetro, pero si ejecuta la opción de Graficar sí se realizará esta tarea sólo que se hará tomando el valor sin actualizar, es decir, se trabajará con el valor anteriormente almacenado para este campo.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Realizar estudios de dinámica

**Caso de prueba:** Se entrarán los valores de los parámetros. Se entrarán además los valores de las condiciones iniciales y los tiempos inicial y final.

**Entrada:** Se selecciona el sistema y el modelo matemático. Se entran valores de parámetros y condiciones iniciales, se entran además los valores de tiempo inicial y final. Se elige el método matemático a utilizar.

System=Interleuquina

Mathematical Model=dy2\_eval

Population x value=1

Population y value=1

Population z value=1

Parameters a value=2

Parameters b value=1

Parameters c value=3

Parameters d value=2

Parameters e=3

Parameters f value=5

Parameters g value=1

Parameters h value=10

Parameters i value=3

Parameters k value=6

Parameters l value=5

Initial Time=0

Final Time=10

Mathematical Method= ode45

**Resultado:** Se obtiene la gráfica de dinámica de la población con los resultados de la simulación realizada.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

### CU Analizar utilizando reglas

	Casos Válidos	Casos no Válidos
Population value	<ul style="list-style-type: none"><li>• Números.</li></ul>	<ul style="list-style-type: none"><li>• Letras.</li><li>• Sin valor.</li></ul>
Population Initial value	<ul style="list-style-type: none"><li>• Números.</li></ul>	<ul style="list-style-type: none"><li>• Letras.</li><li>• Sin valor.</li></ul>
Initial Time	<ul style="list-style-type: none"><li>• Números.</li></ul>	<ul style="list-style-type: none"><li>• Letras.</li><li>• Sin valor.</li></ul>
Final Time	<ul style="list-style-type: none"><li>• Números.</li></ul>	<ul style="list-style-type: none"><li>• Letras.</li><li>• Sin valor.</li></ul>
Edit rulers	<ul style="list-style-type: none"><li>• Expresiones lógicas.</li></ul>	<ul style="list-style-type: none"><li>• No ser una expresión lógica.</li></ul>

**Caso de uso:** Analizar utilizando reglas

**Caso de prueba:** No se entrarán valores para los tiempos inicial y final.

**Entrada:** Se seleccionan el sistema y el modelo. Se seleccionan los parámetros que se van a variar, los valores iniciales de las poblaciones, se inserta la regla y el error. Se dejarán en blanco los campos para los valores del tiempo inicial y el tiempo final.

System=Interleuquina

Mathematical Model=dy2\_eval

Population x value=1

Population y value=1

Population z value=1

Population a -->Varia Initial value=1 Final value=12 Points Number value=2

Population b -->No Varia Initial value=1

Population c -->No Varia Initial value=1

Population d -->Varia Initial value=2 Final value=16 Points Number value=3

Population e-->No Varia Initial value=1

Population f -->No Varia Initial value=1

Population g -->No Varia Initial value=1

Population h -->No Varia Initial value=1

Population i -->No Varia Initial value=1

Population j -->No Varia Initial value=1

Population k -->No Varia Initial value=1

Population l-->No varía Initial value=1

Initial Time=\_\_\_\_\_

Final Time=\_\_\_\_\_

Edit rulers= $x>y>z$

$=y<z$

Error=2

**Resultado:** Se mostrará un mensaje de error indicando que los valores para los tiempos deben ser introducidos.

**Condiciones:** Tener conexión con la Base de Datos, el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Analizar utilizando reglas

**Caso de prueba:** Se entrarán los valores de los parámetros y de las condiciones iniciales que se van a variar y las reglas que defina el usuario.

**Entrada:** Se seleccionan el sistema y el modelo. Se seleccionan los parámetros que se van a variar, los valores iniciales de las poblaciones, se inserta la regla y el error. Se dejarán en blanco los campos para los valores del tiempo inicial y el tiempo final.

System=Interleuquina

Mathematical Model= $dy_2\_eval$

Population x value=10

Population y value=9

Population z value=10

Population a --> No Varia Initial value=1  
Population b -->No Varia Initial value=1  
Population c -->No Varia Initial value=11  
Population d --> No Varia Initial value=1  
Population e-->Varia Initial value=1 Final value=10 Points Number value=10  
Population f -->No Varia Initial value=1  
Population g -->No Varia Initial value=1  
Population h -->No Varia Initial value=11  
Population i -->No Varia Initial value=1  
Population j -->Varia Initial value=1 Final value=100 Points Number value=10  
Population k -->No Varia Initial value=1  
Population l-->No varia Initial value=11

Initial Time=0

Final Time=10

Edit rulers=x>y

=x<z

Error=0,001

**Resultado:** Se obtiene una gráfica con los resultados de las simulaciones agrupados según las reglas definidas por el usuario, separadas por una frontera entre ellas y van a representar también el comportamiento de un parámetro con respecto a otro.

**Condiciones:** Tener conexión con la Base de Datos, el usuario deberá estar autenticado en la misma previamente.

## CU Gestionar sistema biológico

	Casos Válidos	Casos no Válidos
Name	<ul style="list-style-type: none"> <li>Números.</li> <li>Cadena de caracteres.</li> </ul>	<ul style="list-style-type: none"> <li>Sin valor.</li> </ul>

**Caso de uso:** Gestionar sistema biológico(Sección: Modificar un sistema biológico)

**Caso de prueba:** Se selecciona el sistema al cual se le desea cambiar el nombre, pero se deja en blanco el campo del nombre.

**Entrada:**

System=Interleuquina

Name=\_\_\_\_\_

**Resultado:** Se deja sin llenar el campo donde va el nombre del sistema por lo que el sistema muestra un mensaje de error al hacer clic en la opción Update.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Gestionar sistema biológico(Sección: Modificar un sistema biológico)

**Caso de prueba:** Se selecciona el sistema al cual se le desea cambiar el nombre y se entra el nuevo nombre.

**Entrada:**

System=Interleuquina

Name=InterleuquinaB

**Resultado:** Al hacer clic en la opción Update se renombró el sistema.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

## CU Gestionar modelo matemático

	Casos Válidos	Casos no Válidos
Variable	<ul style="list-style-type: none"> <li>• Cadena de caracteres.</li> <li>• La cantidad de variables y de ecuaciones diferenciales deberán ser la misma.</li> </ul>	<ul style="list-style-type: none"> <li>• Sin valor.</li> <li>• Diferentes las cantidades de variables y de ecuaciones diferenciales.</li> </ul>

**Caso de uso:** Gestionar modelo matemático(Sección: Modificar un modelo matemático)

**Caso de prueba:** Se selecciona el sistema y el modelo matemático, se carga el modelo y se adiciona otra variable a las que ya estaban.

**Entrada:**

System=Interleuquina

Mathematical Model=dy2\_eval

Add Condiciones Iniciales y(4) variable=w

**Resultado:** El modelo tenía 3 variables, al adicionarle una más serían 4 y el sistema de ecuaciones diferenciales contaba con 3 ecuaciones por lo que se mostrará un mensaje diciendo que falta una ecuación.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Gestionar modelo matemático

**Caso de prueba:** Se selecciona el sistema y el modelo matemático, se carga el modelo y se inserta otra variable y otra ecuación.

**Entrada:**

System=Interleuquina

Mathematical Model=dy2\_eval

Add Condiciones Iniciales y(4) variable=w

Mathematical Format Ecuaciones diferenciales:  $dw = b+f*(z/(g+z))*rb-rf$

**Resultado:** Se agregó otra variable por lo que se adicionó la ecuación diferencial para dicha variable, modificándose el modelo anterior con los nuevos datos.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

### CU Analizar utilizando clustering

	Casos Válidos	Casos no Válidos
Time	<ul style="list-style-type: none"><li>• Números.</li></ul>	<ul style="list-style-type: none"><li>• Letras.</li><li>• Sin valor.</li></ul>
Choose cluster type	<ul style="list-style-type: none"><li>• Statics parameters.</li><li>• Statics Initial Conditions.</li><li>• All Simulation.</li></ul>	<ul style="list-style-type: none"><li>• ____</li></ul>

**Caso de uso:** Analizar utilizando clustering

**Caso de prueba:** Se introduce un valor de tiempo para el cual se quiere obtener todas las simulaciones, pero para dicho valor no se tiene una simulación (esto es porque al realizar las simulaciones según el paso que se tomó, el 5 no es uno de los valores de tiempo con el cual se obtiene un resultado al simular).

**Entrada:** Se selecciona el sistema, el modelo matemático y se elige que el tipo de agrupación sea utilizando todas las simulaciones.

System=Interleuquina

Mathematical Model=dy2\_eval

Time=5

Choose cluster type: All Simulation-->seleccionado

**Resultado:** Se introduce el valor cinco para el tiempo, pero el sistema mostrará un mensaje especificando que para ese tiempo no se tiene ninguna simulación.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Analizar utilizando clustering

**Caso de prueba:** Se introduce el valor de tiempo 10 para el cual se quiere obtener las simulaciones que cumplen con ciertas condiciones fijas.

**Entrada:** Se selecciona el sistema, el modelo matemático y se seleccionan aquellas simulaciones que cumplan con tener ciertas condiciones iniciales fijas.

System=Interleuquina

Mathematical Model=dy2\_eval

Time=10

Choose cluster type: Statics Initial Conditions-->seleccionado

Condition x Value=1

Condition y Value=1

Condition z Value=1

**Resultado:** Para los valores entrados se obtuvieron las simulaciones agrupadas por número de cluster, luego se seleccionó una simulación y se escogió el tipo de gráfica a mostrar y se visualizó.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

## CU Editar configuración

	Casos Válidos	Casos no Válidos
IP	<ul style="list-style-type: none"> <li>• Formato de la expresión regular.</li> </ul>	<ul style="list-style-type: none"> <li>• No tener el formato de la expresión regular.</li> <li>• Sin valor.</li> </ul>
Matlab Command	<ul style="list-style-type: none"> <li>• Cadena de caracteres.</li> </ul>	<ul style="list-style-type: none"> <li>• Sin valor.</li> </ul>

**Caso de uso:** Editar configuración

**Caso de prueba:** Al adicionar una PC se deberá entrar el IP de dicha computadora, en este caso se entrarán dos letras.

**Entrada:** Se entran el IP de la PC y el comando para levantar Matlab.

IP=pp

Matlab Command=matlab

**Resultado:** Se mostrará un mensaje especificando que no pueden ser letras, sino números separados por “.”.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

**Caso de uso:** Editar configuración

**Caso de prueba:** Al adicionar una PC se entra el IP de dicha computadora.

**Entrada:** Se entran el IP de la PC y el comando para levantar Matlab.

IP=10.34.20.22

Matlab Command=matlab

**Resultado:** Se adiciona la PC a la Grid.

**Condiciones:** Tener conexión con la Base de Datos y el usuario deberá estar autenticado en la misma previamente.

### **4.3 Conclusiones**

En este capítulo se trató la validación del software mediante los casos de prueba de caja negra para validar el correcto funcionamiento de la aplicación comprobándose que no existen errores en las funciones operativas del software. Además se definió el diagrama de componentes con el objetivo de brindar una idea de cómo se implementó el software en término de componentes.

## CONCLUSIONES

- ✓ Se desarrolló una aplicación que permite realizar simulaciones distribuidas de sistemas biológicos que son modelados mediante Sistemas de Ecuaciones Diferenciales, almacena los resultados y realiza meta-análisis sobre estos datos.
- ✓ Se realizó el análisis y diseño de una aplicación para la simulación distribuida y análisis de sistemas biológicos.
- ✓ Se implementó una aplicación en Java para la simulación y análisis de sistemas biológicos.
- ✓ Se realizaron pruebas de caja negra para verificar el correcto funcionamiento de la aplicación.

## RECOMENDACIONES

- Incorporar otras técnicas de análisis con el objetivo de hacer investigaciones más profundas.
- Implementar funciones en Java que permitan la resolución del modelo matemático.
- Profundizar en el estudio de los algoritmos de clustering del software Weka con el objetivo de reprogramarlo y brindar otras facilidades al usuario, como por ejemplo la selección de la fórmula de distancia a utilizar.
- Implementar la Ayuda de la aplicación con el objetivo de informarle al usuario cómo trabajar con la aplicación y de mostrar información importante referente a los sistemas biológicos.

## REFERENCIAS BIBLIOGRÁFICAS

1. VIC, U. D. *Descripción del departamento Biología de Sistemas España*: [Consultado el: 18 de mayo de 2007]. Disponible en: [www.uvic.cat/eps/dept/biologiasistemas/es/inici.html](http://www.uvic.cat/eps/dept/biologiasistemas/es/inici.html).
2. TERRA. *La Biología de Sistemas mejorará el tratamiento de enfermedades* [Consultado el: 16 de enero de 2007]. Disponible en: <http://www.imbiomed.com/Innsz/Nnv46n4/espanol/Wnn44-07.html>.
3. RAMÍREZ, K. O. *Plataforma Computacional para el Desarrollo de la Biología de Sistemas. Trabajo de diploma para optar por el título de Ingeniería en Informática*. Facultad de Matemática-Computación. Universidad de La Habana, 2004.
4. RO, C. *Modelo matemáticos en medicina y biología. Bases teóricas y fundamentos* [Consultado el: 16 de enero de 2007]. (Revista Invest Clin). Disponible en: <http://www.imbiomed.com/Innsz/Nnv46n4/espanol/Wnn44-07.html>.
5. MANUEL ÁLVAREZ BLANCO; ARNALDO GÓMEZ MONTENEGRO, et al. *Matemática numérica*. 2003.
6. MARTA BONACINA y BORTOLATO, G. *Uso del ordenador en la simulación de procesos* [Consultado el: 18 de mayo de 2007]. Disponible en: <http://www.fceia.unr.edu.ar/fceia1/publicaciones/numero7/articulo1/articulo1resumen.htm>.
7. INSPIRED, R. *El futuro de la industria farmacéutica* [Consultado el: 16 de enero de 2007]. Disponible en: [http://www-03.ibm.com/industries/education/doc/content/bin/G299\\_0887\\_00\\_LSP.pdf](http://www-03.ibm.com/industries/education/doc/content/bin/G299_0887_00_LSP.pdf).

8. PARRA, R. A. M. *Ventajas / Desventajas de la Simulación* [Consultado el: 19 de enero de 2007]. Disponible en: <http://docencia.50webs.com/simula03.htm>.
9. VIRGÓS, E. M. *Grid* [Consultado el: 19 de enero de 2007]. Disponible en: [http://www.matematicalia.net/index.php?option=com\\_content&task=view&id=293&Itemid=191](http://www.matematicalia.net/index.php?option=com_content&task=view&id=293&Itemid=191).
10. MASSÓ, J. *Aplicación de la tecnología Grid en el mercado financiero* [Consultado el: 18 de mayo de 2007]. Disponible en: <http://www.risklab-madrid.uam.es/es/seminarios/pasados/marzo2003.pdf>.
11. MARTÍNEZ, C. M. *Dinámica de la población* [Consultado el: 18 de enero de 2007]. Disponible en: <http://platea.pntic.mec.es/~cmarti3/CTMA/BIOSFERA/crecto.htm>.
12. FERNÁNDEZ, D. E. A. *Clustering (Agrupamiento)* Universidad Católica de Córdoba, Fac. Ingeniería: [Consultado el: 23 de enero de 2006]. Disponible en: <http://www.uccor.edu.ar/paginas/seminarios/Cursos/DM-Medicine/Clustering.ppt>.
13. SCHIAFFINO, P. D. S. *Descubrimiento de Conocimiento en Bases de Datos* [Consultado el: 17 de enero de 2007]. Disponible en: <http://www.exa.unicen.edu.ar/catedras/dbdiscov/clase2c.pdf>.
14. SANCHEZ, M. A. M. *Metodologías de Desarrollo de Software* [Consultado el: 17 de enero de 2007]. Disponible en: <http://64.233.167.104/search?q=cache:lzgHcDIV3CEJ:www.informatizate.net/articulos/pdfs>

[/metodologias de desarrollo de software 07062004.pdf+metodolog%C3%ADas+para+el+desrrollo+de+software&hl=es&gl=cu&ct=clnk&cd=1&client=firefox.](#)

15. IVAR JACOBSON; GRADY BOOCH, et al. El Proceso Unificado de Desarrollo de Software. 1999. vol. volumen 1.
16. GRACIA, J. *Introducción a MySQL* Última actualización: 9 de junio del 2005. [Consultado el: 19 de enero de 2007]. Disponible en: <http://www.webestilo.com/mysql/intro.phtml>.
17. METAEMOTION. Meta Weka: OpenSource Data Mining [Consultado el: 19 de enero de 2007]. Disponible en: <http://blog.metaemotion.com/2007/02/14/weka-opensource-data-mining/>.
18. Tejada, D. H. Disponible en: <http://teleprogramadores.com>

## BIBLIOGRAFÍA

1. *Cálculo diferencial e integral II , tema No.9. Integrales curvilíneas, de superficie y de volúmen* [Consultado el: 5 de febrero de 2007]. Disponible en: [http://64.233.167.104/search?q=cache:LNBwtM\\_CcaQJ:exa.unne.edu.ar/investigacion/calculo2/public\\_html/anamat1\\_doc/tema9.doc+tender+al+infinito%2Bpunto+estacionario&hl=es&gl=cu&ct=clnk&cd=3&client=firefox](http://64.233.167.104/search?q=cache:LNBwtM_CcaQJ:exa.unne.edu.ar/investigacion/calculo2/public_html/anamat1_doc/tema9.doc+tender+al+infinito%2Bpunto+estacionario&hl=es&gl=cu&ct=clnk&cd=3&client=firefox).
2. ALEJANDRO MEDINA SANTIAGO; EVA VALDEZ ALEMÁN, *et al. Simulación de sistemas analógicos para la solución de un grupo de problemas de optimización* [Consultado el: 5 de febrero de 2007]. Disponible en: [http://www.uvmnet.edu/investigacion/episteme/numero6-06/reportes/a\\_simulacion.asp](http://www.uvmnet.edu/investigacion/episteme/numero6-06/reportes/a_simulacion.asp).
3. ANDREW EMERSON; SILVIA GIULIANI, *et al. ImmunoGrid. Large scale simulation of the immune system* [Consultado el: 22 de febrero de 2007]. Disponible en: <http://www.immunogrid.org>.
4. BIOPHYSICS, C. B. A. *ByoDyn* [Consultado el: 20 de febrero de 2007]. Disponible en: <http://diana.imim.es/ByoDyn>.
5. BRIGITTA TADMOR y TIDOR, B. *Interdisciplinary research and education at the biology–engineering–computer science interface: a perspective*. september, 2007, vol. 10,
6. CELLWARE. *Cellware 3.0.2 is available* [Consultado el: 30 de agosto de 2006]. Disponible en: <http://www.cellware.org/index.html>.
7. DAN PELLEGG y MOORE, A. *X-means: Extending K-means with Efficient estimation of the Number of Clusters* [Consultado el: 17 de enero de 2007]. Disponible en: <http://www.cs.cmu.edu/~dpelleg/download/xmeans.ps>.
8. DORADO, I. C. *Simulación de sistemas* [Consultado el: 16 de enero de 2007 ]. Disponible en: <http://www.monografias.com/trabajos20/simulacion-sistemas/simulacion-sistemas.shtml>.
9. ECOLÓGICO, P. E. P. *La dinámica poblacional* [Consultado el: 18 de enero de 2007]. Disponible en: [http://www.peruecologico.com.pe/lib\\_c2\\_t04.htm](http://www.peruecologico.com.pe/lib_c2_t04.htm).

10. EXPERIMENTAR. *¿Qué es un modelo?* [Consultado el: 16 de enero de 2007]. Disponible en: <http://www.experimentar.gov.ar/newexperi/NOTAS/maquinaviva/modelo.htm>.
11. FERNÁNDEZ, D. E. A. *Clustering (Agrupamiento)* Universidad Católica de Córdoba, Fac. Ingeniería: [Consultado el: 23 de enero de 2006]. Disponible en: <http://www.uccor.edu.ar/paginas/seminarios/Cursos/DM-Medicine/Clustering.ppt>.
12. FLEMING, R. J. S. *Uso de la simulación computarizada para avanzar la investigación agrícola en los países en vía de desarrollo* Estados Unidos de Norteamérica: [Consultado el: 19 de enero de 2007]. Disponible en: <http://www.public.iastate.edu/~rjsalvad/manuscripts/general/simulacion.html>.
13. FRASER, B. D. *Caos en sistemas* [Consultado el: 5 de febrero de 2007 ]. Disponible en: <http://segre.upc.es/nllab/iterated-es.html>.
14. INSPIRED, R. *El futuro de la industria farmacéutica* [Consultado el: 16 de enero de 2007]. Disponible en: [http://www-03.ibm.com/industries/education/doc/content/bin/G299\\_0887\\_00\\_LSP.pdf](http://www-03.ibm.com/industries/education/doc/content/bin/G299_0887_00_LSP.pdf).
15. IVAR JACOBSON; GRADY BOOCH, *et al. El Proceso Unificado de Desarrollo de Software*. 1999. vol. volumen 1.
16. JAMES M. MCCOLLUM y KEMP, W. A. *BioGrid Documentation* Última actualización: April 30, 2005.
17. JAVA, T. D. *Características de Java* [Consultado el: 19 de enero de 2007]. Disponible en: <http://www.cica.es/formacion/JavaTut/Intro/tabla.html>.
18. JAVARM. *La Biología de lo complejo* [Consultado el: 16 de enero de 2007]. Disponible en: <http://geneura.ugr.es/~jmerelo/atalaya/print.cgi?id=/historias/341&nombre=javarm>.
19. LABORATORY, L. A. N. *Overview and Benefits* [Consultado el: 16 de nov de 2006]. Disponible en: <http://cellsignaling.lanl.gov/bionetgen/benefits.shtml>.
20. LEÓN, A. R. S. *Herramienta para la modelación de sistemas biológicos*. Universidad de las Ciencias Informáticas, 2006.

21. MAÑAS, J. A. *Puebra de Programas* [Consultado el: 4 de junio de 2007]. Disponible en: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
22. MARÍA JOSÉ FERRÓN y GONZÁLEZ, J. P. *Sistemas de Programas(CI-3711)* [Consultado el: 4 de junio de 2007]. Disponible en: <http://www ldc.usb.ve/~teruel/ci3711/test1/index.html>.
23. MARTÍNEZ, C. M. *Dinámica de la población* [Consultado el: 18 de enero de 2007]. Disponible en: <http://platea.pntic.mec.es/~cmarti3/CTMA/BIOSFERA/crecto.htm>.
24. MATHWORKS, T. *SimBiology 2.1.1* [Consultado el: 9 de septiembre de 2006]. Disponible en: <http://www.mathworks.com/products/simbiology/description5.html>.
25. METAEMOTION. *Meta Weka: OpenSource Data Mining* [Consultado el: 19 de enero de 2007]. Disponible en: <http://blog.metaemotion.com/2007/02/14/weka-opensource-data-mining/>.
26. MIGUEL GARRE; JUÁN JOSÉ CUADRADO, *et al. Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software* Dept. de Ciencias de la Computación ETS Ingeniería Informática, Universidad de Alcalá: [Consultado el: 23 de enero de 2007]. Disponible en: <http://www.sc.ehu.es/jiwdocoj/remis/docs/GarreAdis05.pdf>.
27. MONOGRAFÍAS. *Matlab* [Consultado el: 19 de enero de 2007]. Disponible en: <http://www.monografias.com/trabajos5/matlab/matlab.shtml>.
28. NETBEANS. *NetBeans* [Consultado el: 19 de enero de 2007]. Disponible en: [http://www.netbeans.org/index\\_es.html](http://www.netbeans.org/index_es.html).
29. PARADIGM, V. *Documentation* [Consultado el: 18 de enero de 2007]. Disponible en: <http://www.visual-paradigm.com/>.
30. PARRA, R. A. M. *Ventajas / Desventajas de la Simulación* [Consultado el: 19 de enero de 2007]. Disponible en: <http://docencia.50webs.com/simula03.htm>.

31. PATRICIO LETELIER y PENADÉS, M. C. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* Universidad Politécnica de Valencia: [Consultado el: 17 de enero de 2007]. Disponible en: <http://www.willydev.net/descargas/masyxp.pdf>.
32. PENNSYLVANIA, B. G. A. T. U. O. *Software*. vol. 2006, Disponible en: <http://biocomp.cis.upenn.edu/index.php3>.
33. PETER BLAIS y YOU, L. *Dynetica User's Guide (version 1.2beta)*. Department of Biomedical Engineering Duke University: [Consultado el: 8 de diciembre de 2006]. Disponible en: <http://labs.genome.duke.edu/YouLab/software/dynetica/Dynetica%20User%20Guide%20v.1.pdf>.
34. PROJECT, N. P. *Introduction to JSim Sensitivity Analysis* [Consultado el: 20 de febrero de 2007]. Disponible en: [http://physiome.org/jsim/docs/User\\_Sens.html](http://physiome.org/jsim/docs/User_Sens.html).
35. RAMÍREZ, K. O. *Plataforma Computacional para el Desarrollo de la Biología de Sistemas. Trabajo de diploma para optar por el título de Ingeniería en Informática*. Facultad de Matemática-Computación. Universidad de La Habana, 2004.
36. RO, C. *Modelo matemáticos en medicina y biología. Bases teóricas y fundamentos* [Consultado el: 16 de enero de 2007]. (Revista Invest Clin). Disponible en: <http://www.imbiomed.com/Innsz/Nnv46n4/espanol/Wnn44-07.html>.
37. SANCHEZ, M. A. M. *Metodologías de Desarrollo de Software* [Consultado el: 17 de enero de 2007]. Disponible en: [http://64.233.167.104/search?q=cache:lzgHcDIV3CEJ:www.informatizate.net/articulos/pdfs/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.pdf+metodolog%C3%ADas+para+el+desrrollo+de+software&hl=es&gl=cu&ct=clnk&cd=1&client=firefox](http://64.233.167.104/search?q=cache:lzgHcDIV3CEJ:www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf+metodolog%C3%ADas+para+el+desrrollo+de+software&hl=es&gl=cu&ct=clnk&cd=1&client=firefox).
38. SCHIAFFINO, P. D. S. *Descubrimiento de Conocimiento en Bases de Datos* [Consultado el: 17 de enero de 2007]. Disponible en: <http://www.exa.unicen.edu.ar/catedras/dbdiscov/clase2c.pdf>.
39. TEJADA, D. H. Disponible en: <http://teleprogramadores.com>.

40. TEKNOMOS, K. *Numerical Example of K-Means Clustering* [Consultado el: 17 de enero de 2007]. Disponible en: <http://people.revoledu.com/kardi/tutorial/kMean/NumericalExample.htm>.
41. TERRA. *La Biología de Sistemas mejorará el tratamiento de enfermedades* [Consultado el: 16 de enero de 2007]. Disponible en: <http://www.imbiomed.com/Innsz/Nnv46n4/espanol/Wnn44-07.html>.
42. TICOM. *Las Aplicaciones de Red Distribuidas crean Supercomputadoras Virtuales* [Consultado el: 25 de enero de 2007]. Disponible en: <http://www.ticomperu.com/supercomputadorasvirtualesincompleto.htm>.
43. VARGAS, L. F. G. *Una Introducción a los Automatas Celulares* [Consultado el: 24 de enero de 2007]. Disponible en: <http://yupana.autonoma.edu.co/publicaciones/yupana/005/autocelular/Automatas.html>.
44. VCELL. *Parameter Scan(new in VCell 4.2 beta)* [Consultado el: 9 de septiembre de 2006]. Disponible en: [http://www.vcell.org/login/parameter\\_scans/parameter\\_scans.html](http://www.vcell.org/login/parameter_scans/parameter_scans.html).
45. VIRGÓS, E. M. *Grid* [Consultado el: 19 de enero de 2007]. Disponible en: [http://www.matematicalia.net/index.php?option=com\\_content&task=view&id=293&Itemid=191](http://www.matematicalia.net/index.php?option=com_content&task=view&id=293&Itemid=191).
46. ZAMORANO, J. P. *Clasificación de patrones: Métodos no supervisados* [Consultado el: 23 de enero de 2007]. Disponible en: [http://www.iula.upf.es/materials/050418porta\\_5.pdf](http://www.iula.upf.es/materials/050418porta_5.pdf).

ANEXOS

**Anexo 1: Descripción Caso de uso Autenticar usuario**

<b>Nombre del CU</b>	<b>Autenticar usuario</b>	
<b>Actores</b>	Bioinformático	
<b>Propósito</b>	Verificar que el usuario tiene acceso a la base de datos.	
<b>Resumen</b>	El caso de uso se inicia cuando el bioinformático desea realizar la investigación de un sistema biológico, para lo cual abre la aplicación y necesita ser usuario de la base de datos para gestionar la información correspondiente.	
<b>Referencias</b>	R8	
<b>Precondiciones</b>		
<b>Poscondiciones</b>	Se tendrá almacenada la información de los sistemas biológicos que se estudien.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El bioinformático abre la aplicación para realizar un estudio.	2. El sistema muestra una ventana para que el bioinformático se autentique.	
3. El bioinformático entra los datos necesarios (servidor, puerto, nombre de la base de datos, usuario y contraseña).	4. El sistema comprueba que tiene acceso a la BD.	

	5. Se establece conexión con la BD.
6. El bioinformático tiene acceso a la información contenida en la BD.	7. Termina el caso de uso.
<b>Curso Alternativo de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	4. El sistema comprueba que no tiene acceso a la BD.
	5. El sistema muestra un mensaje indicando que el usuario no tiene acceso, finalizando así el caso de uso.
<b>Prioridad:</b>	Crítico

## Anexo 2: Descripción Caso de uso Realizar estudios de dinámica

<b>Nombre del CU</b>	<b>Realizar estudios de dinámica</b>
<b>Actores</b>	Bioinformático
<b>Propósito</b>	Realizar una simulación simple y obtener las gráficas del resultado obtenido.
<b>Resumen</b>	El caso de uso se inicia cuando el bioinformático decide realizar un estudio de dinámica de la población realizando para esto una simulación simple con cuyo resultado se obtendrán las gráficas de población vs población y población vs tiempo.
<b>Referencias</b>	R3, R3.1,R3.2 Casos de uso asociados: ----

<b>Precondiciones</b>	Estar autenticado en la BD previamente y tener sistemas biológicos almacenados en la BD.
<b>Poscondiciones</b>	Se obtendrán las gráficas deseadas, ya sea de Mapa de Fase o de Dinámica de la Población.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El bioinformático selecciona el sistema biológico y el modelo matemático de la simulación que desea realizar, luego selecciona la interfaz correspondiente para realizar el estudio de la dinámica del sistema.	2. La aplicación muestra una ventana para entrar los datos de la simulación.
3. El bioinformático entra los valores de condiciones iniciales, valores de parámetros y el rango del tiempo en que desea realizar la integración. Además selecciona el método numérico.	4. La aplicación busca la simulación en la base de datos para verificar que no esté hecha previamente.
	5. El sistema levanta el asistente matemático Matlab.
	6. El sistema realiza la simulación y almacena los resultados en la BD.
	7. El sistema muestra una ventana para que se seleccione el tipo de gráfica.
8. El bioinformático selecciona el tipo de gráfica.	9. El sistema muestra la interfaz correspondiente.

	10. Termina el caso de uso.
<b>Curso Alterno de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	4. La simulación se encuentra almacenada en la BD, pasando así a la acción 7.
<b>Prioridad:</b>	Crítico

### Anexo 3: Descripción Caso de uso Analizar utilizando reglas

<b>Nombre del CU</b>	<b>Analizar utilizando reglas</b>
<b>Actores</b>	Bioinformático
<b>Propósito</b>	Realizar análisis de múltiples simulaciones haciendo uso de reglas definidas por el usuario.
<b>Resumen</b>	El bioinformático selecciona un sistema y un modelo del cual quiere obtener todos los resultados de las simulaciones, inserta los datos necesarios para realizar las mismas y luego define reglas para agruparlas, para ello la aplicación se conectará a la BD y realizará las consultas necesarias para devolver los datos deseados. Luego graficará las simulaciones que se encuentren dentro del rango agrupado por las reglas.
<b>Referencias</b>	R6, R6.1 Casos de uso asociados: Realizar simulaciones múltiples.

<b>Precondiciones</b>	Estar autenticado en la BD previamente y tener datos almacenados en la BD.	
<b>Poscondiciones</b>	Datos agrupados según las reglas definidas por el usuario.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El bioinformático accede a la interfaz correspondiente luego de seleccionar con cuál sistema biológico y modelo matemático desea realizar el análisis.	2. El sistema muestra la interfaz correspondiente.	
3. El bioinformático inserta los valores de parámetros y condiciones iniciales, así como los dos únicos parámetros que necesita variar, se insertan además las reglas por las cuales se agruparán, el rango de tiempo de las simulaciones, el error de las bifurcaciones y el método de resolución del sistema de ecuaciones diferenciales.	4. El sistema comprueba que no estén hechas las simulaciones.	
	5. El sistema realiza las simulaciones múltiples (ver caso de uso Realizar simulaciones múltiples).	
	6. La aplicación realiza las consultas necesarias a la BD para devolver los resultados de las simulaciones que se corresponden con los datos entrados.	

	7. El sistema agrupa los resultados según las reglas definidas.
	8. El sistema se conecta a la Grid.
9. El bioinformático sube a la Grid el fichero que contiene los resultados agrupados para obtener la gráfica que muestra las fronteras donde se separa a una regla de otra(s).	10. La Grid distribuye el trabajo de ir calculando los puntos de fronteras para la representación gráfica.
	11. El sistema muestra la gráfica.
	12. Termina el caso de uso.
<b>Curso Alternativo de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	4. El sistema comprueba que ya están hechas las simulaciones, pasando así a la acción 6.
<b>Prioridad:</b>	Crítico

#### Anexo 4: Descripción Caso de uso Gestionar sistema biológico

<b>Nombre del CU</b>	<b>Gestionar sistema biológico</b>	
<b>Actores</b>	Bioinformático	
<b>Propósito</b>	El bioinformático necesita tener almacenada la información de un sistema biológico, permitirá insertar, modificar o eliminar un sistema biológico.	
<b>Resumen</b>	El caso de uso comienza cuando el bioinformático decide gestionar la información de un sistema biológico, ya sea insertar un sistema biológico, modificarlo o eliminarlo.	
<b>Referencias</b>	R1.1, R1.2, R1.3, R1.4 Casos de uso asociado : ---	
<b>Precondiciones</b>	Estar autenticado en la BD previamente.	
<b>Poscondiciones</b>	El sistema tendrá almacenada la información del sistema biológico.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El bioinformático podrá:</p> <ul style="list-style-type: none"> <li>Insertar los datos de un nuevo sistema.</li> <li>Modificar un sistema biológico existente.</li> <li>Eliminar un sistema biológico existente.</li> </ul>	<p>2. El sistema mostrará la información necesaria para:</p> <ul style="list-style-type: none"> <li>si “Insertar los datos de un nuevo sistema” ir a Sección Insertar los datos de un nuevo sistema biológico.</li> <li>si “Modificar un sistema biológico” ir a Sección Modificar un sistema biológico.</li> </ul>	

	si “Eliminar un sistema biológico” ir a Sección Eliminar un sistema biológico.
<b>Sección Insertar los datos de un nuevo sistema biológico</b>	
1. El bioinformático introduce el nombre del sistema biológico y el SBML.	2. El sistema almacenará la información entrada, en la BD.
<b>Sección Eliminar un sistema biológico</b>	
1. El bioinformático seleccionará un sistema biológico y luego escogerá la opción que permita eliminar.	2. El sistema busca en la BD el sistema biológico y luego lo elimina.
<b>Sección Modificar un sistema biológico</b>	
1. El bioinformático selecciona el sistema biológico a modificar.	2. El sistema muestra la información del sistema biológico.
3. El bioinformático modifica sólo el nombre y el SBML del sistema.	4. El sistema actualiza los datos.
	5. Termina el caso de uso.
<b>Prioridad:</b>	Crítico

## Anexo 5: Descripción Caso de uso Gestionar modelo matemático

<b>Nombre del CU</b>	<b>Gestionar modelo matemático</b>	
<b>Actores</b>	Bioinformática	
<b>Propósito</b>	El caso de uso gestionar modelo matemático permitirá insertar un nuevo modelo matemático, visualizar la información de otros ya existentes, eliminarlos y modificarlos.	
<b>Resumen</b>	El caso de uso comienza cuando el bioinformático desea gestionar los datos de un modelo matemático, para esto debe seleccionar el sistema biológico para el cual va a gestionar la información del modelo matemático.	
<b>Referencias</b>	R2.1, R2.2, R2.3 Casos de uso asociados: ---	
<b>Precondiciones</b>	Estar autenticado en la BD previamente.	
<b>Poscondiciones</b>	El modelo matemático será almacenado en la BD.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El bioinformático podrá:</p> <ul style="list-style-type: none"> <li>• Insertar un nuevo modelo matemático.</li> <li>• Eliminar un modelo matemático.</li> <li>• Modificar un modelo matemático.</li> </ul>	<p>2. El sistema mostrará la información necesaria para:</p> <ul style="list-style-type: none"> <li>• si “Insertar un nuevo modelo matemático” ir a Sección Insertar un nuevo modelo matemático.</li> <li>• si “Eliminar un modelo matemático”</li> </ul>	

	<p>ir a Sección Eliminar un modelo matemático.</p> <ul style="list-style-type: none"> <li>• si “Modificar un modelo matemático” ir a Sección Modificar un modelo matemático.</li> </ul>
	3. Termina el caso de uso.
<b>Sección Insertar un nuevo modelo matemático</b>	
1. El bioinformático selecciona el sistema biológico para el cual insertará un modelo matemático, entra el nombre, los parámetros, condiciones iniciales y el MathML correspondiente.	2. La aplicación convierte el modelo matemático que está en formato MathML a formato Matlab.
	3. La aplicación almacena todos los datos introducidos en la BD (modelo matemático, parámetros, condiciones iniciales, MathML, Fichero Matlab).
<b>Sección Eliminar un modelo matemático</b>	
1. El bioinformático selecciona el modelo matemático que desea eliminar.	2. El sistema busca en la BD el modelo matemático y luego lo elimina.
<b>Sección Modificar un modelo matemático</b>	
1. El bioinformático selecciona el modelo matemático que desea modificar.	2. El sistema muestra la información del modelo.
3. El bioinformático sólo modifica el nombre del	4. El sistema actualiza los datos.

modelo.	
	5. Termina el caso de uso.
<b>Prioridad:</b>	Crítico

#### Anexo 6: Descripción Caso de uso Analizar utilizando clustering

<b>Nombre del CU</b>	<b>Analizar utilizando clustering</b>
<b>Actores</b>	Bioinformático
<b>Propósito</b>	Agrupar los resultados de las simulaciones utilizando algoritmos de clustering.
<b>Resumen</b>	El bioinformático define qué datos agrupar, para ello el sistema realiza una consulta a la BD para escoger la matriz de datos a clusterizar. Luego el bioinformático selecciona qué tipo de algoritmo de clustering utilizar y se obtienen los clusters deseados.
<b>Referencias</b>	R5.1, R5.2, R5.3 Casos de uso asociado : ---
<b>Precondiciones</b>	Estar autenticado en la BD previamente y tener simulaciones almacenadas en la BD.
<b>Poscondiciones</b>	Datos agrupados según el algoritmo seleccionado.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El bioinformático accede a la interfaz correspondiente al seleccionar el modelo matemático.</p>	<p>2. El sistema muestra una ventana para escoger según qué datos agrupar.</p>
<p>3. El Bioinformático selecciona qué resultados desea agrupar, para ello tiene la opción de elegirlos por:</p> <p>Estado de equilibrio.</p> <p>Parámetros fijos.</p> <p>Condiciones iniciales fijas.</p>	<p>4. El sistema realiza la consulta para devolver el conjunto de resultados que le interesa al bioinformático agrupar:</p> <ul style="list-style-type: none"> <li>• si “Estado de equilibrio” ir a sección Estado de equilibrio.</li> <li>• si “Parámetros fijos” ir a sección Parámetros fijos.</li> <li>• si “Condiciones iniciales fijas” ir a sección Condiciones iniciales fijas.</li> </ul>
	<p>5. El sistema muestra una ventana para realizar los clusters y escoger el tipo de algoritmo a utilizar.</p>
<p>6. El bioinformático selecciona qué tipo de algoritmo a utilizar para agrupar los datos.</p>	<p>7. La aplicación ejecuta el algoritmo de clustering seleccionado.</p>
	<p>8. La aplicación devuelve una gráfica con los clusters realizados.</p>

<p>9. El bioinformático desea obtener una gráfica, ya sea, mapa de fase o de dinámica de la población, según elija y selecciona una simulación que esté contenida en uno de los clusters para mostrar su gráfica.(Ver caso de uso Realizar estudios de dinámica)</p>	<p>10. El sistema obtiene la gráfica (ver caso de uso Realizar estudios de dinámica).</p>
	<p>11. Termina el caso de uso</p>
<p><b>Sección Estado de equilibrio</b></p>	
<p>1. El usuario selecciona realizar los clusters con todas las simulaciones que llegaron al equilibrio pasando como datos el tiempo final.</p>	<p>2. El sistema genera la consulta correspondiente.</p>
	<p>3. El sistema consulta la BD para obtener las simulaciones que cumplan con dicha restricción.</p>
	<p>4. El sistema genera un fichero temporal con todas las simulaciones según el formato de archivos Weka.</p>
	<p>5. El sistema accede a dicho fichero para obtener los datos a agrupar.</p>
<p><b>Sección Parámetros fijos</b></p>	

1. El usuario selecciona realizar los clusters con las simulaciones que tengan los parámetros fijos establecidos por él y condiciones iniciales variables.	2. El sistema muestra una ventana para especificar el valor de los parámetros que son fijos.
3. El usuario entra los valores de los parámetros y el tiempo final.	4. El sistema genera un ID con el valor de los parámetros.
	5. El sistema genera una consulta para obtener las simulaciones que tengan esos valores de parámetros sin importar el valor de las condiciones iniciales.
	6. El sistema consulta la BD para obtener las simulaciones que cumplan con dicha restricción.
	7. El sistema genera un fichero temporal, según el formato de archivos Weka con todas las simulaciones que cumplieron con lo establecido por el usuario.
	8. El sistema accede a dicho fichero para obtener los datos a agrupar.
<b>Sección Condiciones iniciales fijas</b>	
1. El usuario selecciona realizar los clusters	2. El sistema muestra una ventana

con las simulaciones que tengan las condiciones iniciales fijas establecidas por él previamente y los parámetros variables.	para especificar el valor de las condiciones iniciales que son fijas.
3. El usuario entra los valores de las condiciones iniciales y el tiempo final.	4. El sistema genera un ID con el valor de las condiciones iniciales.
	5. El sistema genera una consulta para obtener las simulaciones que tengan esos valores de condiciones iniciales sin importar el valor de los parámetros.
	6. El sistema consulta la BD para obtener las simulaciones que cumplan con dicha restricción.
	7. El sistema genera un fichero temporal con todas las simulaciones que cumplieron con lo establecido por el usuario.
	8. El sistema accede a dicho fichero para obtener los datos a agrupar.
<b>Prioridad:</b>	<b>Crítico</b>

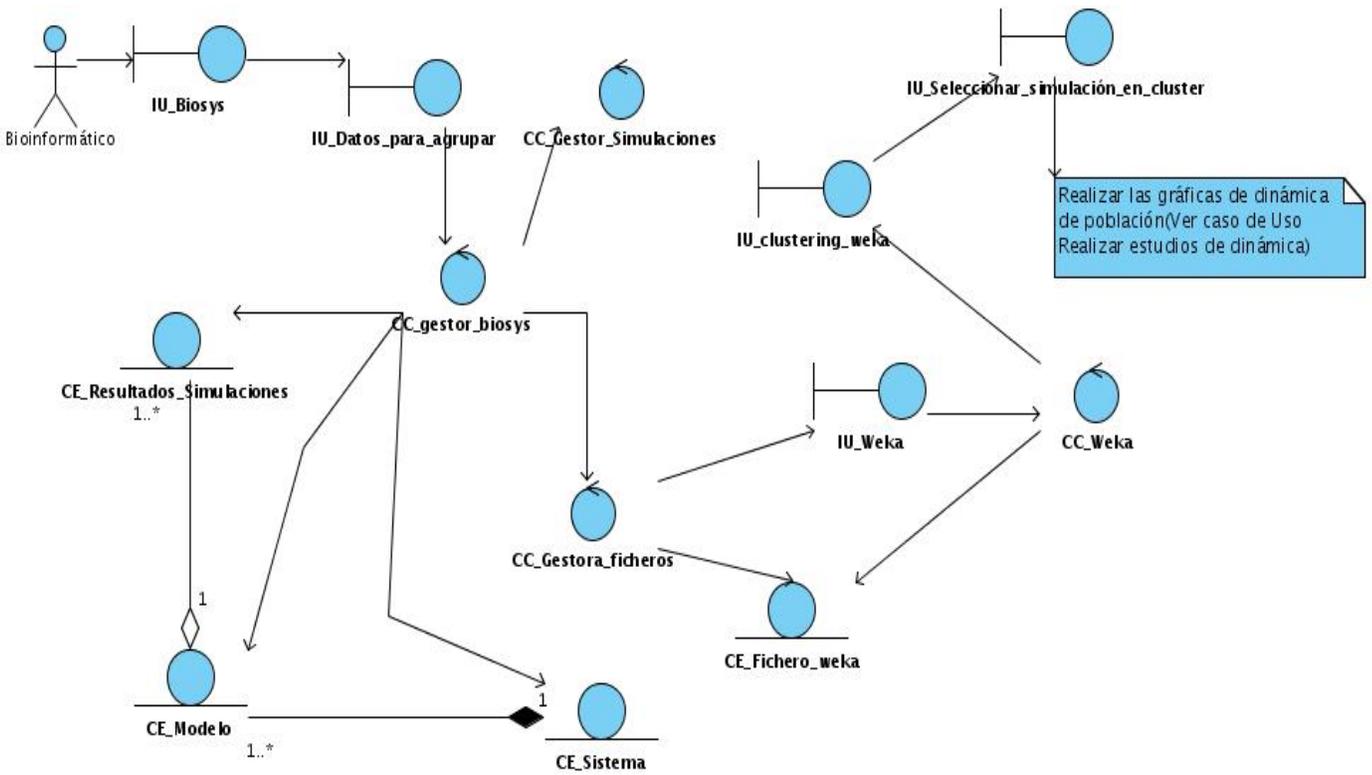
## Anexo 7: Descripción Caso de uso Editar Configuración

<b>Nombre del CU</b>	<b>Editar Configuración</b>	
<b>Actores</b>	Bioinformático	
<b>Propósito</b>	El bioinformático necesita configurar la información de las PCs con las que simulará distribuidamente.	
<b>Resumen</b>	El caso de uso comienza cuando el bioinformático decide configurar la información de las máquinas que se usará para la simulación distribuida, para lo cual tendrá la posibilidad de agregar más máquinas o eliminar las que desee.	
<b>Referencias</b>	R7 Casos de uso asociados: ---	
<b>Precondiciones</b>	Estar autenticado en la BD previamente.	
<b>Poscondiciones</b>	El sistema tendrá almacenada en un fichero de configuración la información de las máquinas que utilice.	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El bioinformático accede a la interfaz correspondiente.	2. El sistema muestra una ventana con toda la información que se requiere para la configuración de la conexión.	
3. El bioinformático podrá: Agregar nuevas máquinas a utilizar.	4. El sistema mostrará la información necesaria para:	

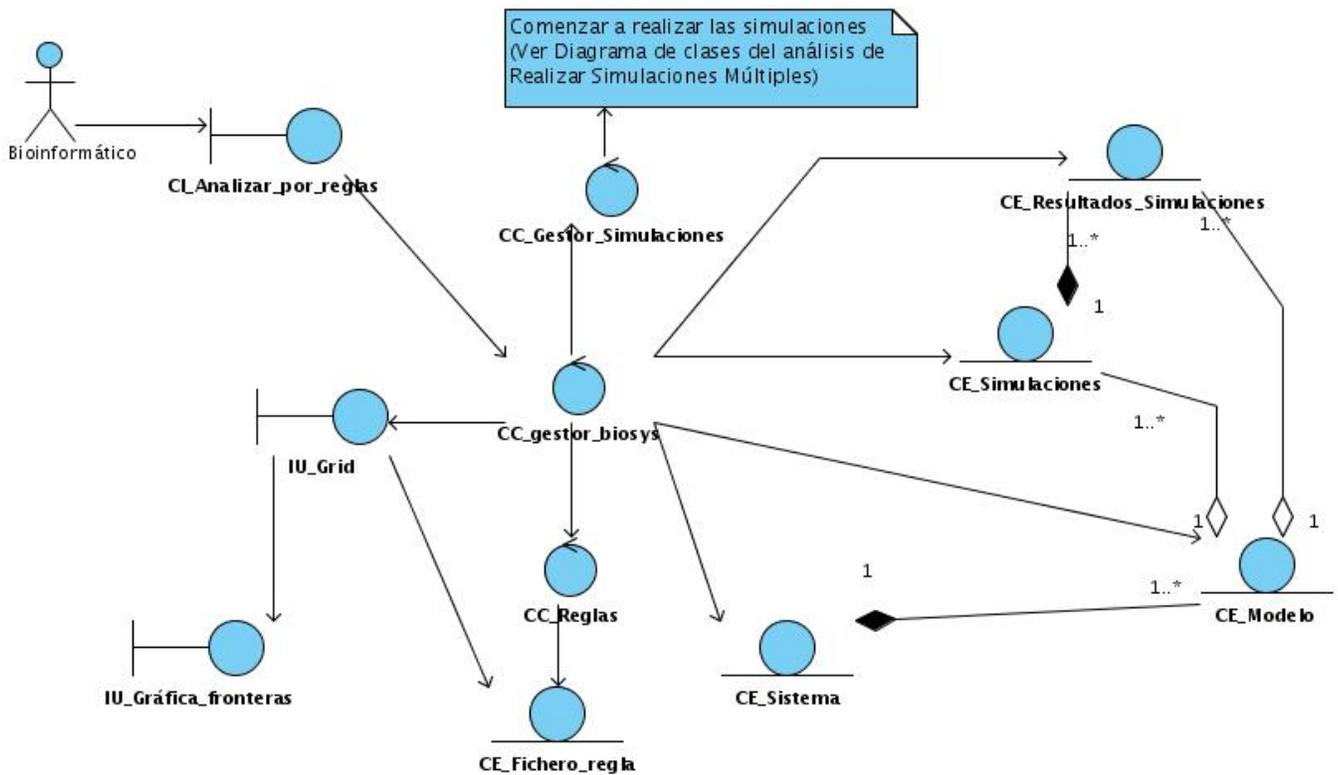
<p>Modificar la configuración de una PC.</p> <p>Eliminar alguna conexión con una PC.</p>	<p>si “Agregar nuevas máquinas a utilizar” ir a Sección Agregar nuevas máquinas a utilizar.</p> <p>si “Modificar la configuración de una PC” ir a Sección Modificar la configuración de una PC.</p> <p>si “Eliminar alguna conexión con una PC” ir a Sección Eliminar alguna conexión con una PC.</p>
	5. Termina el caso de uso.
<b>Sección Agregar nuevas máquinas a utilizar</b>	
1. El bioinformático selecciona agregar una nueva PC.	2. El sistema agrega una nueva fila en donde se introducirán los datos necesarios.
3. El bioinformático introduce el IP de la PC y el comando para ejecutar Matlab en la misma y acepta los cambios.	4. El sistema actualiza el fichero de configuración.
<b>Sección Modificar la configuración de una PC</b>	
1. El bioinformático selecciona a qué PC cambiarle algún dato.	
2. El bioinformático actualiza la información existente y acepta los cambios.	3. El sistema actualiza el fichero de configuración.

Sección Eliminar alguna conexión con una PC	
1. El bioinformático selecciona qué PC eliminar y acepta los cambios.	2. El sistema elimina la PC seleccionada.
	3. El sistema actualiza el fichero de configuración.
<b>Prioridad:</b>	Crítico

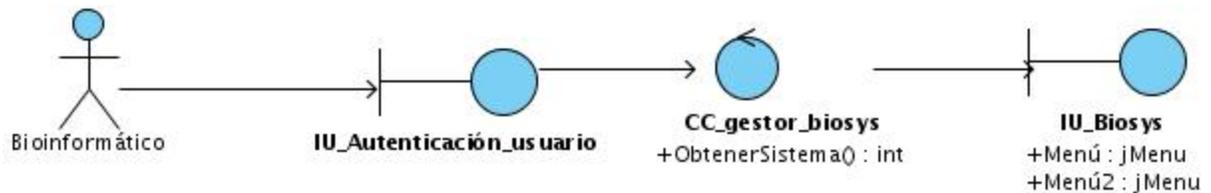
**Anexo 8: Diagrama de clases del análisis del Caso de uso Analizar utilizando clustering.**



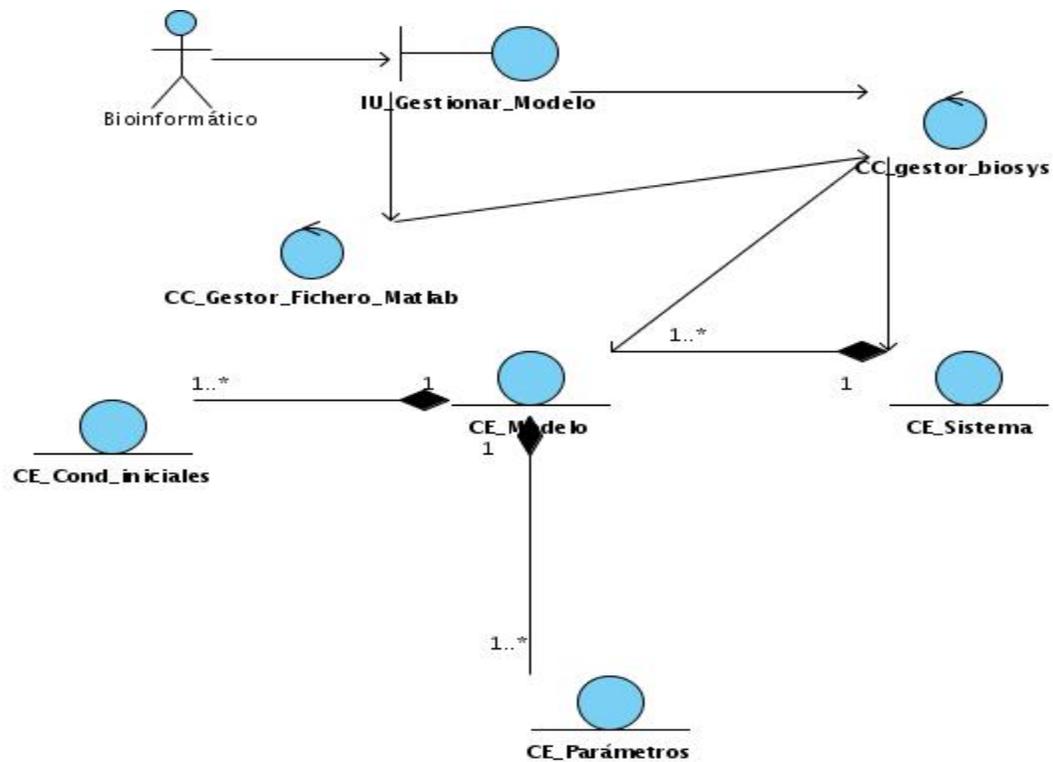
**Anexo 9: Diagrama de clases del análisis del Caso de uso Analizar utilizando reglas.**



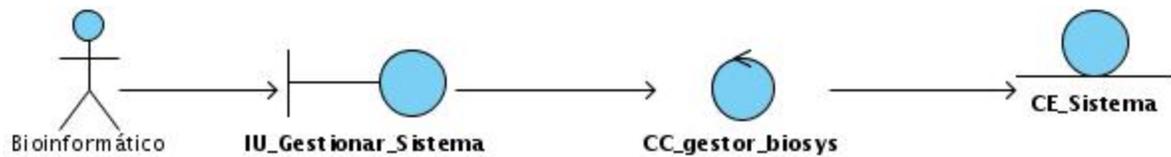
**Anexo 10: Diagrama de clases del análisis del Caso de uso Autenticar Usuario.**



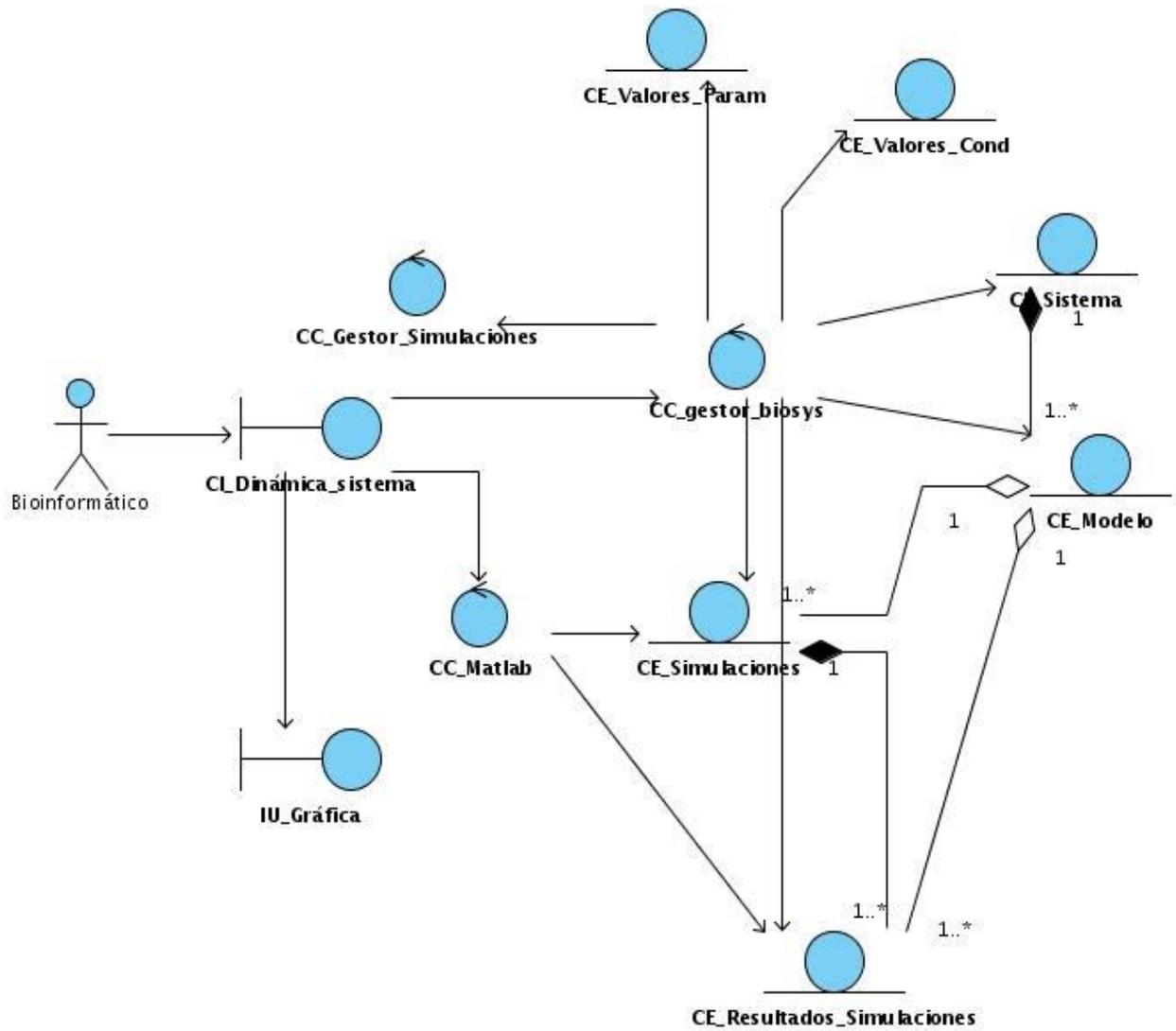
**Anexo 11: Diagrama de clases del análisis del Caso de uso Gestionar Modelo Matemático.**



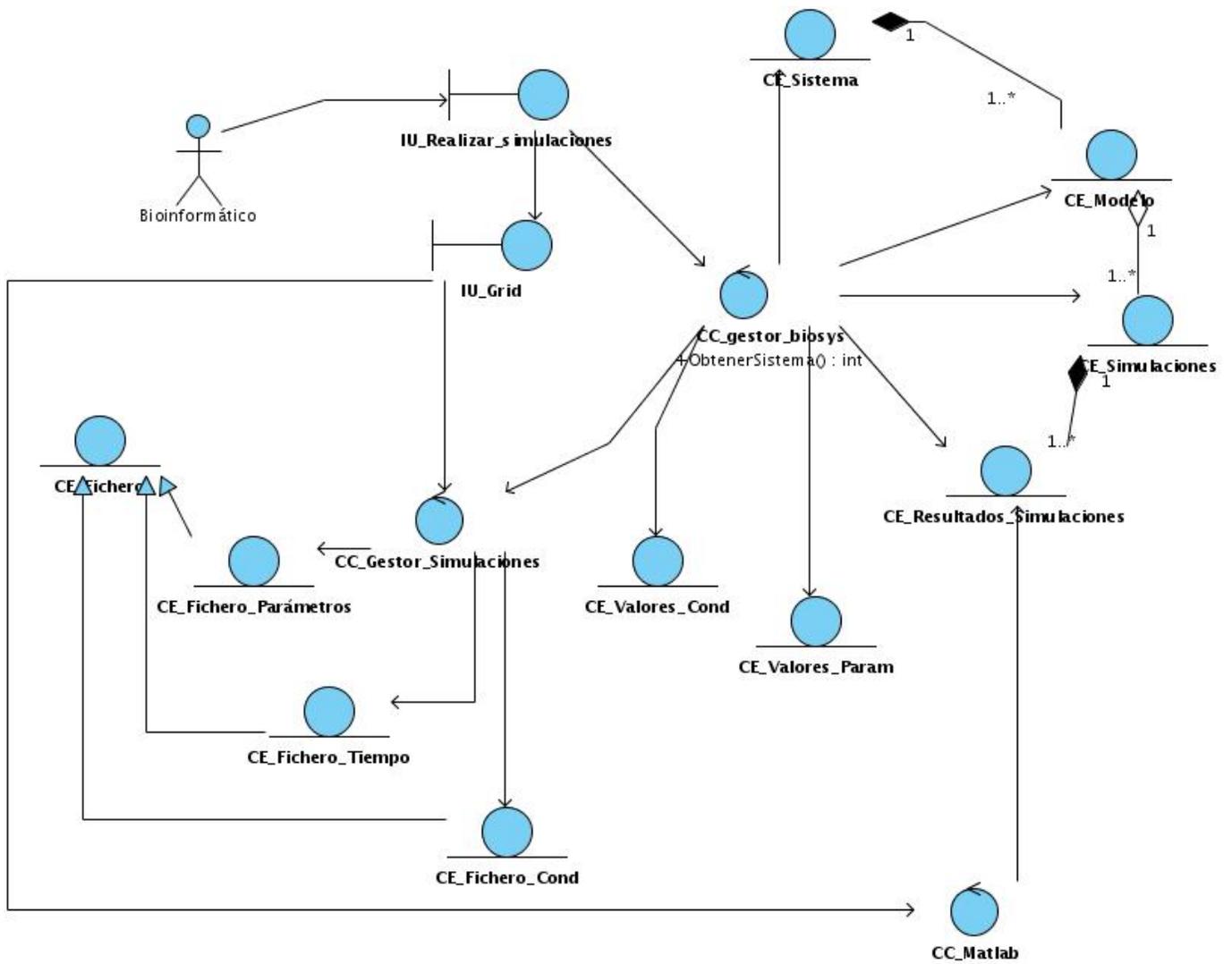
**Anexo 12: Diagrama de clases del análisis del Caso de uso Gestionar Sistema Biológico.**



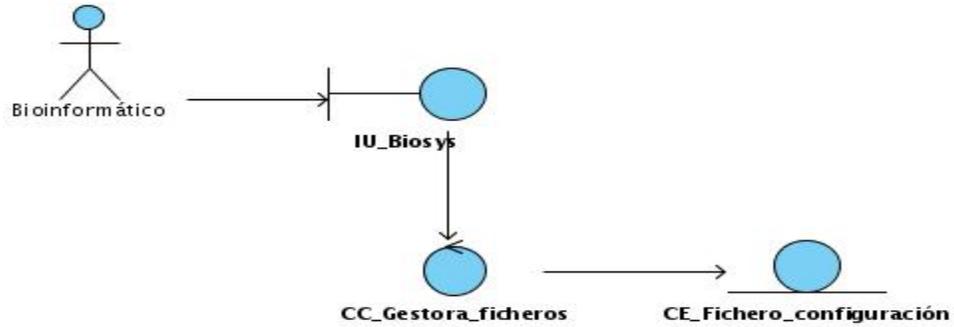
Anexo 13: Diagrama de clases del análisis del Caso de uso Realizar estudios de dinámica.



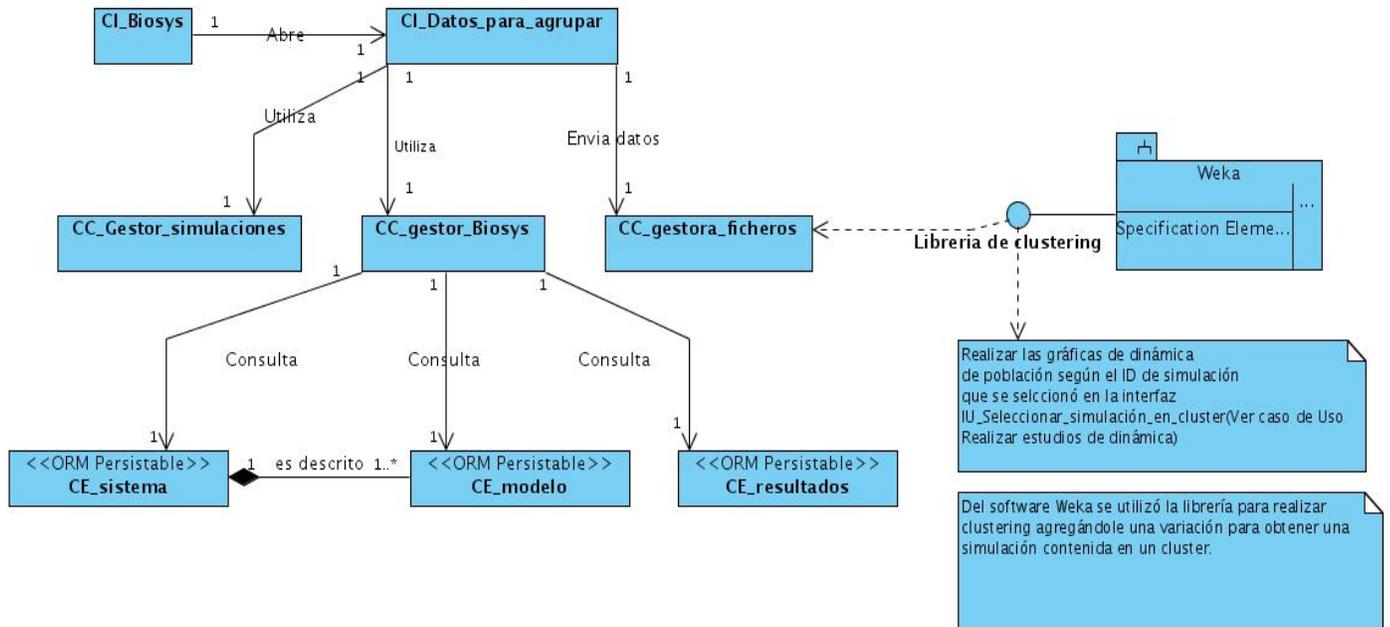
**Anexo 14: Diagrama de clases del análisis del Caso de uso Realizar simulaciones múltiples.**



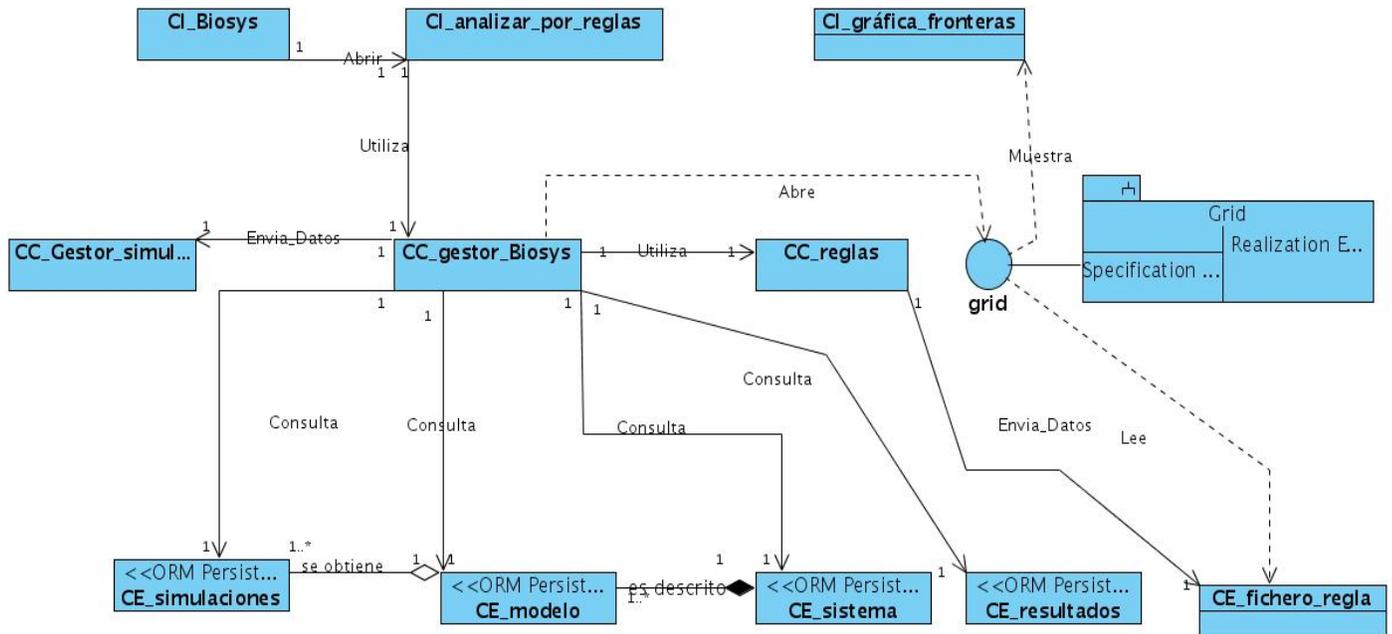
**Anexo 15: Diagrama de clases del análisis del Caso de uso Editar Configuración.**



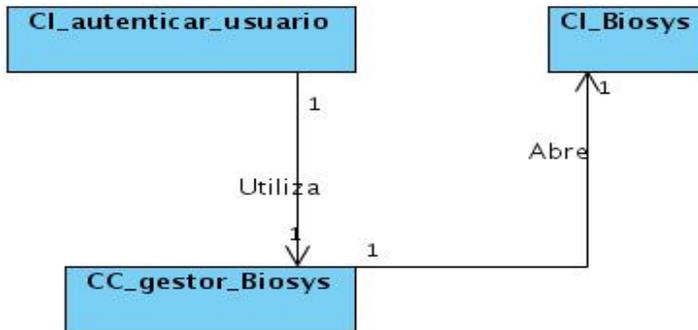
**Anexo 16: Diagrama de clases del Caso de uso Analizar utilizando clustering.**



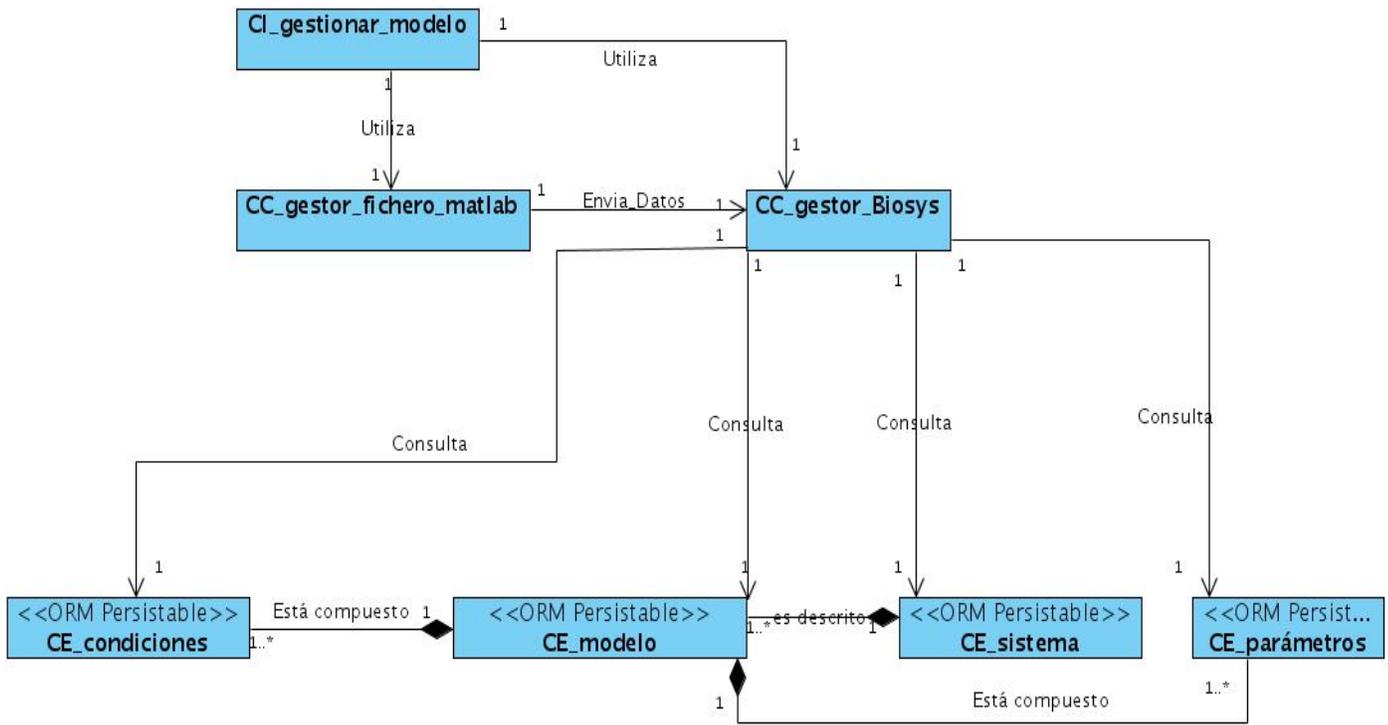
**Anexo 17: Diagrama de clases del Caso de uso Analizar utilizando reglas.**



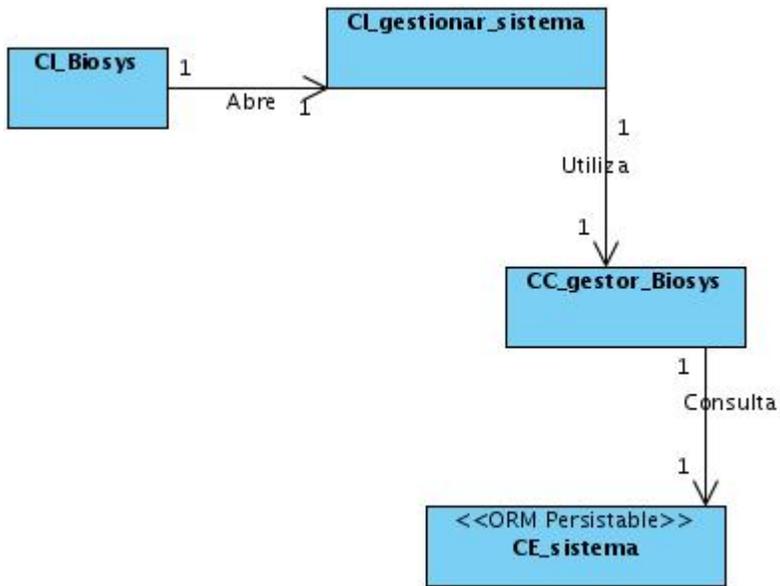
**Anexo 18: Diagrama de clases del Caso de uso Autenticar Usuario.**



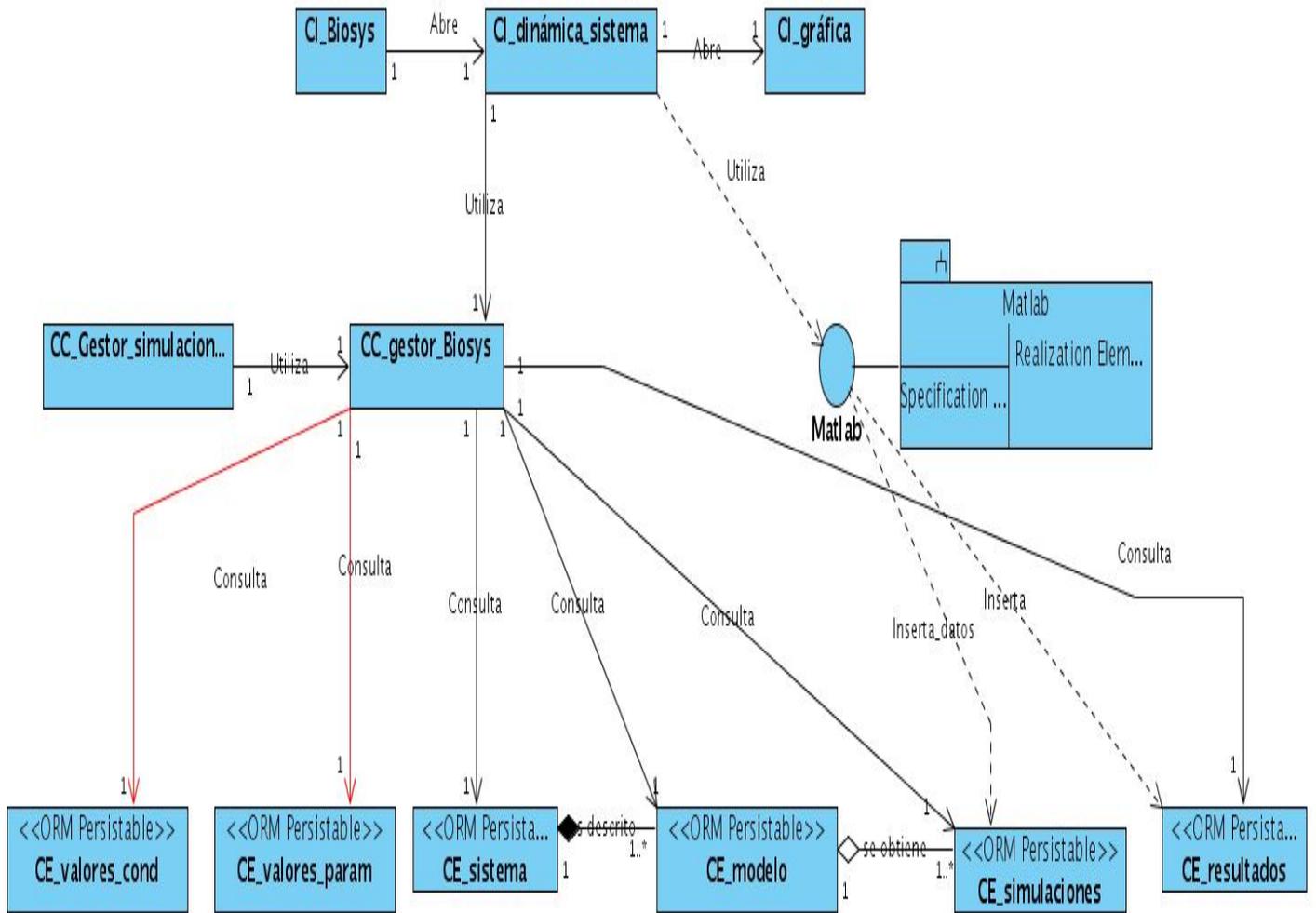
**Anexo 19: Diagrama de clases del Caso de uso Gestionar Modelo Matemático.**



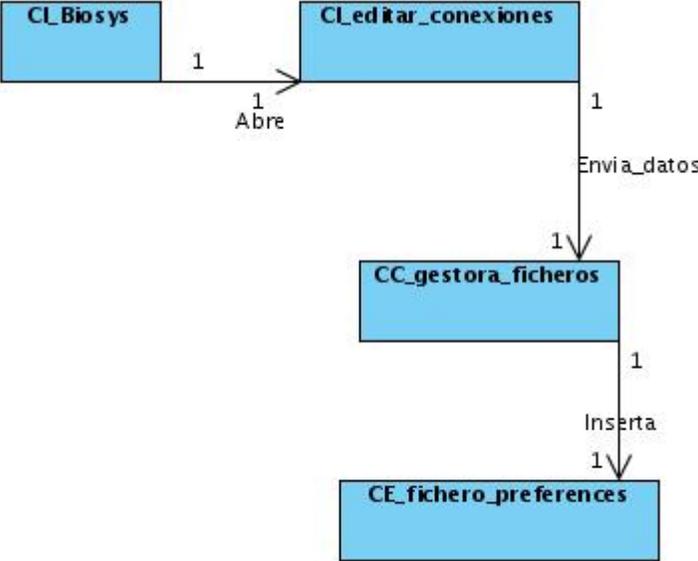
Anexo 20: Diagrama de clases del Caso de uso Gestionar Sistema Biológico.



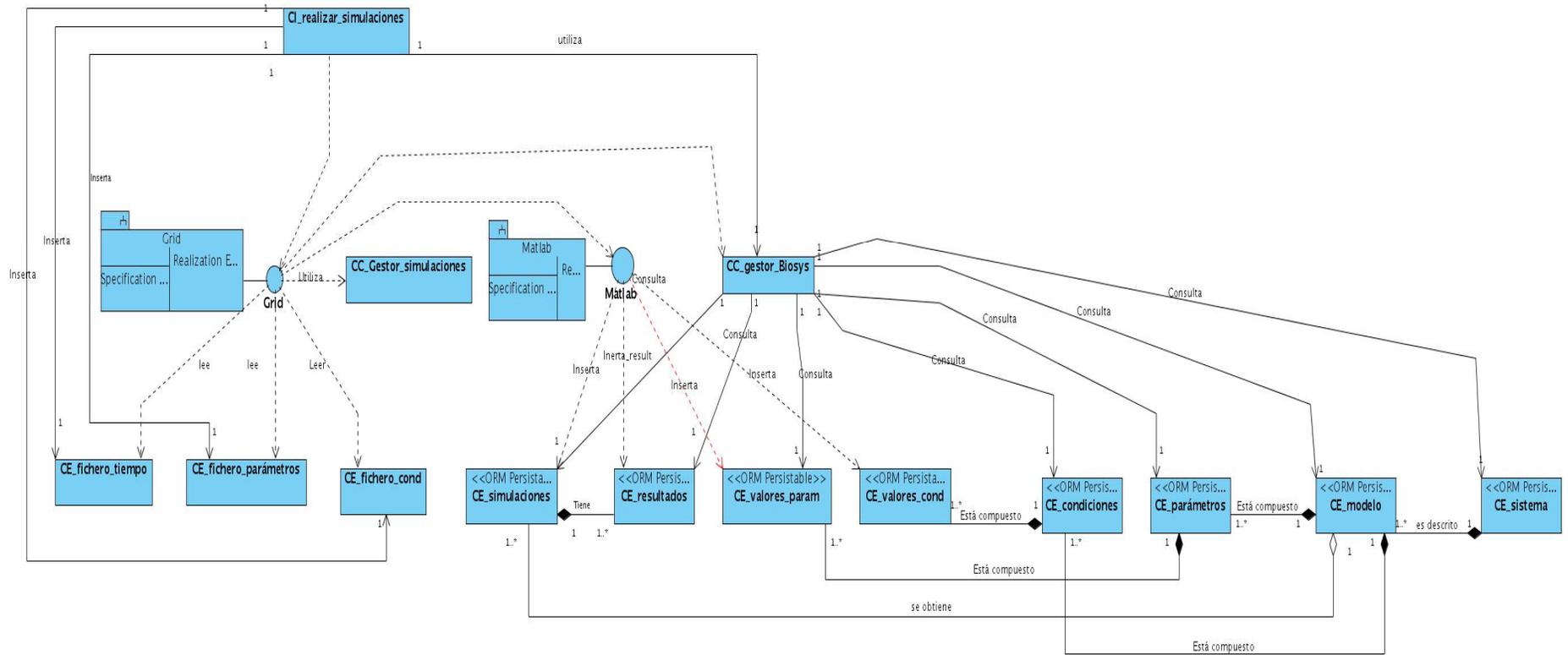
**Anexo 21: Diagrama de clases del Caso de uso Realizar estudios de dinámica.**



Anexo 22: Diagrama de clases del Caso de uso Editar Configuración.



### Anexo 23: Diagrama de clases del Caso de uso Realizar simulaciones múltiples.



## GLOSARIO

- ✓ BS: Biología de Sistemas (ciencia biológica que estudia los sistemas como un todo).
- ✓ Clusters: Grupo de objetos que comparten características semejantes.
- ✓ Clustering: Tareas de la inteligencia artificial que permite agrupar datos de acuerdo a la similitud entre los mismos.
- ✓ BD: Base de Datos.
- ✓ CIGB: Centro de Ingeniería Genética y Biotecnología.
- ✓ CIM: Centro de Inmunología Molecular.
- ✓ SED: Sistema de Ecuaciones diferenciales.
- ✓ OS: Operating System (Sistema Operativo).