

Universidad de las Ciencias Informáticas

Facultad 6, Bioinformática



Análisis y diseño de una aplicación Web para mejorar la eficacia en el estudio de los Sistemas Biológicos.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autoras: Haymée Llerena Esperón
Haylín Corujo Numa

Tutores: MSc. Yanet Villanueva Armenteros.
MSc. Noel Moreno Lemus.
Lic. Yanelis Benítez Fernández

Ciudad de La Habana, 2007

"Año 49 de la Revolución"

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 firmamos la presente a los 2 días del mes de julio de 2007.

Haymée Llerena Esperón

Nombre completo del primer autor

Haylín Corujo Numa

Nombre completo del segundo autor

MSc. Yanet Villanueva Armenteros

Nombre completo del primer tutor

MSc. Noel Moreno Lemus

Nombre completo del segundo tutor

AGRADECIMIENTOS

A nuestras madres que son nuestro rayito de luz y fuente de inspiración, gracias por hacer hasta lo imposible para que nuestros sueños se hagan realidad.

Agradecemos a nuestra tutora y profesora Yanet Villanueva Armenteros por su ayuda incondicional y maternal.

Agradecemos a nuestro tutor Noel Moreno Lemus por su asesoría.

Agradecemos a todas aquellas personas que de una forma u otra han participado en nuestra formación como profesionales y a todas aquellas personas que han contribuido en la realización de este trabajo, en especial a Liusmila Nieto, Oscar Cedeño.y al profesor Febe Ángel Ciudad Ricardo.

Agradecemos a la Revolución por darnos la oportunidad de estudiar y superarnos.

DEDICATORIA

Haymee Llerena Esperón.

A mi mamá Betty por su inmenso amor, cariño, ayuda y dedicación.

A mis abuelos Georgina y Rogelio, por su constante preocupación.

A mi tía Rosa por ser una madre más, a ella todo mi amor y cariño.

A mi primo y a mi tío por su apoyo y preocupación.

A mis amigas Haylín, Yanet, Indira y Martha por su gran ayuda y comprensión.

A mis hermanitas Vivian y Milena por quererme tanto.

Haylín Corujo Numa.

A mi mamá y mi hermano que han seguido mis estudios y mi formación como profesional, apoyándome y orientándome en cada momento.

A mi padre Waldo Corujo Fong por su ayuda.

A mis queridas hermanitas más pequeñas Claudia Belmonte Pera y Vivian Hernández Villanueva que han sabido ser pacientes y consecuentes.

A Beatriz Esperón Estrada, Rosa M. Esperón Estrada y a Yanet Villanueva Armenteros que han sido como unas madres para mí. Gracias por su ayuda y apoyo, las adoro.

A mis abuelos Gladys Cedeño, Ángela Fong, Felipe Corujo, Georgina Estrada, Rogelio Esperón y a mi abuelito que extraño mucho Francisco Numa.

A mis tías Kodalys Corujo, Kenia Corujo, Maritza Cedeño, Elia y Mercedita. Gracias por su ayuda.

A mis primas y primos Yanet Pera, Nadieska Corujo, Lisette, Georgito, Geydis Corujo, Luisita, Ayasil Fong, Yoilan Pera.

A mis amigos en especial a: Haymee Llerena, Gisselle Silveira, Elyen Vital, Indira Plá, Martha Nieves y Milena Nieves por soportarme en los peores momentos y por ser tan pacientes y comprensivas.

RESUMEN

La investigación surge en el marco de trabajo del Proyecto: Aplicaciones de Software para mejorar la eficacia en el estudio de los de Sistemas Biológicos, desarrollado conjuntamente con el Centro de Inmunología Molecular (CIM) y la Universidad de las Ciencias Informáticas (UCI). En esta investigación se hace el análisis y diseño de una aplicación Web que integra un grupo importante de algoritmos utilizados en la simulación y gestión de la información de Sistemas Biológicos. Esta aplicación permitirá que todos los investigadores tengan la posibilidad de acceder a los sistemas informáticos para realizar todos los cálculos necesarios para las investigaciones vinculadas a Sistemas Biológicos, sin tener que disponer físicamente de grandes recursos computacionales. Se garantiza la portabilidad de la aplicación Web y se desarrollará con herramientas de software libre.

ÍNDICE

AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN.....	IV
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 La Biología de Sistemas.	6
1.2 Modelos matemáticos. Aplicaciones.	7
1.3 La simulación de Sistemas Biológicos.	8
1.4 Tendencias de las aplicaciones de software para la simulación de Sistemas Biológicos. Limitaciones.....	10
1.5 Metodologías, herramientas y técnicas.	12
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	25
2.1 ¿Modelo de negocio o del dominio?.....	25
2.2 Definición de las entidades y los conceptos principales	25
2.3 Requerimientos.....	27
2.3.1 Requerimientos Funcionales	27
2.3.2 Requerimientos No Funcionales.....	28
2.4 Patrones de Casos de Uso.	30
2.5 Definición de los actores	31
2.6 Diagrama de casos de uso del Sistema.	32
2.7 Descripción de los Casos de Usos del Sistema.	32
CAPITULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	54
3.1 Descripción de la arquitectura (Vista de casos de usos).	54
3.2 Modelo de Análisis.....	55
3.2.1 Diagramas de clases del análisis.....	55
3.3 Modelo de Diseño.	59
3.3.1 Patrones.....	60
3.3.2 Principios del diseño.....	61
3.3.3 Clases de la Capa de Acceso a Datos:.....	62
3.3.4 Clases de la Capa de Presentación:.....	63
3.3.5 Arquitectura Web de tres niveles.	63
3.3.6 Diagramas de clases de diseño.....	64
3.4 Base de Datos.	70
CONCLUSIONES	74
RECOMENDACIONES	75
REFERENCIAS BIBLIOGRÁFICAS	76
BIBLIOGRAFÍA.....	77
ANEXOS.....	79
GLOSARIO DE TÉRMINOS.....	90

INTRODUCCIÓN

En los últimos años la Biotecnología ha avanzado considerablemente, su espectro de aplicación es amplio y sus perspectivas de desarrollo son incalculables. Los avances más significativos se manifiestan en la esfera de la salud con la producción, mediante diversas tecnologías, de antibióticos, vitaminas, enzimas, hormonas, vacunas, interferones, productos derivados de la sangre, anticuerpos monoclonales, entre otros.

El incremento del conocimiento sobre la Biotecnología y la ingeniería genética ha originado nuevos tipos de vacunación, como las vacunas de ADN, que han provocado una verdadera revolución en este campo.

En sentido general la Biotecnología ha permitido el avance acelerado de la producción de vacunas más efectivas, menos reactogénicas y en cantidades suficientes para su uso masivo.

La calidad de las producciones biotecnológicas ha interesado a empresas de países altamente desarrollados como Japón y Francia. Entre esos productos biotecnológicos se destacan:

- Vacuna contra la meningitis tipo B y C.
- Vacuna polivalente contra *Haemophilus influenzae* tipo B
- Vacuna recombinante contra la hepatitis B.
- Interferones como el alfa-interferón.
- Más de 100 tipos de anticuerpos monoclonales.
- Estreptoquinasa recombinante para el tratamiento del infarto de miocardio agudo (IMA).
- Enzimas de utilización industrial.
- Policosanol (PPG) para el tratamiento de las lipidemias aterógenas.
- Sistemas diagnósticos para diversas enfermedades como el SIDA-VIH y la lipoproteína (a).
- Factor de crecimiento epidérmico.
- Equipos médicos y programas de computación.
- Sistema Ultra- Micro Analítico (SUMA).
- Eritropoyetina recombinante humana para su utilización en pacientes con insuficiencia renal crónica.

Dentro de las ramas que estudia la Biotecnología, una de las más jóvenes y que ha comenzado a ganar terreno dentro de los científicos que a nivel mundial se dedican a su estudio, está la Biología de Sistemas.

El método reduccionista es el más utilizado por los científicos para estudiar los Sistemas Biológicos, consiste en la descomposición del sistema en sus partes constituyentes, las cuales por separado, son mucho más sencillas de entender y modelar.

Esta estrategia reduccionista se ha demostrado 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 el funcionamiento del sistema como un todo. Teniendo en cuenta esta limitación, surge la necesidad de un nuevo método de estudio para abordar los Sistemas Biológicos, dando paso al surgimiento de un área de investigación conocida como: Biología de Sistemas; planteada en modo teórico a comienzos de los años 40, y que ahora supone uno de los campos más activos que aprovecha 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 solamente necesita de un conocimiento amplio de la Bioquímica, sino de rápidas y potentes herramientas de cálculo masivo. La investigación implica una metodología científica susceptible, que implica no solamente a las personas sino a 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.

La Biología de Sistemas tiene dos componentes fundamentales: uno experimental y uno computacional. El lado experimental proporciona los datos necesarios para la creación y validación de los modelos matemáticos. En cambio el lado computacional está dividido en dos grandes grupos, la modelación y la minería o análisis de los datos ya existentes o generados durante las simulaciones.

Esta nueva disciplina trata de entender los Sistemas Biológicos en diferentes niveles de abstracción, desde el nivel molecular hasta los ecosistemas y haciendo uso de diferentes tipos de modelos matemáticos y técnicos computacionales, que van desde el Sistema de Ecuaciones Diferenciales

(SED) hasta la minería de datos, pasando por cadenas de Markov, Redes Booleanas, Redes Bayesianas, entre otras.

La complejidad que manifiestan las investigaciones en esta área ha hecho necesario el uso de herramientas computacionales así como su desarrollo sistemático. Hasta hace poco tiempo este desarrollo no llegaba a todos los investigadores, pues no existían aplicaciones de software fáciles de usar, que permitieran modelar y analizar Sistemas Biológicos.

En las últimas décadas se han desarrollado varias aplicaciones Web para el estudio de Sistemas Biológicos. Ejemplo: ImmunoGrid, Cellware, Webcell, Vcell, entre otras. La mayor limitación que tienen estas aplicaciones Web 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, así como la gestión de los resultados obtenidos de la simulación.

Dentro de los centros cubanos que utilizan las simulaciones de Sistemas Biológicos en sus investigaciones, se encuentran: 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); todos, centros de Biotecnología creados en la década de los 80.

Sin embargo, estos centros no cuentan muchas veces con las herramientas computacionales necesarias para la realización de estas investigaciones, debido a diferentes causas, dentro de las cuales se destacan, el alto precio de las mejores aplicaciones de software desarrolladas en esta área, los elevados requerimientos de hardware de las mismas, o la necesidad del uso de infraestructuras paralelas o distribuidas, para la realización de muchos de los análisis requeridos en este tipo de estudios. Otra limitante es la no existencia, en muchas ocasiones, de aplicaciones de software que se adapten a las necesidades de simulación que exigen los modelos matemáticos que en dichos centros se desarrollan. Además, la mayoría de las aplicaciones de software existentes adolecen de herramientas capaces de gestionar la información procedente de las simulaciones realizadas.

Teniendo en cuenta lo antes expuesto el **problema científico** que abordaremos en nuestra investigación es: ¿Cómo mejorar la eficacia en las investigaciones de los Sistemas Biológicos?

Por lo tanto nuestro **objeto de estudio** son: Las Aplicaciones Cliente-Servidor.

El **Campo de acción** son: Las Aplicaciones Web en el estudio de los Sistemas Biológicos.

Para solucionar esta problemática se ha propuesto como **objetivo** de la investigación: Realizar el análisis y diseño de una aplicación Web para mejorar la eficacia en las investigaciones de los Sistemas Biológicos.

De este objetivo general se derivan los siguientes **objetivos específicos**:

- Elaborar el modelo conceptual.
- Identificar requisitos funcionales y no funcionales.
- Diseñar los prototipos de interfaz.
- Diseñar las clases a implementar.

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

- Revisión de las tendencias actuales en el estudio de los Sistemas Biológicos.
- Estudio de las aplicaciones de software desarrolladas para la simulación de Sistemas Biológicos.
- Revisión de las técnicas, herramientas y metodologías para diseñar aplicaciones Web.
- Elaboración del modelo conceptual.
- Determinación de los requisitos funcionales y no funcionales.
- Diseño del diagrama de caso de uso del sistema.
- Descripción de los casos de uso del sistema.
- Diseño de los prototipos de interfaz.
- Realización de los diagramas de clases del análisis.
- Realización de los diagramas de clases del diseño.

Resultado esperado de la investigación

- El diseño de las clases y la interacción entre ellas, de la aplicación Web en desarrollo.
- Diseño de una aplicación Web para mejorar la eficacia en las investigaciones de los Sistemas Biológicos. La aplicación Web que se diseñará no sólo mostrará la dinámica del sistema sino

que permitirá al usuario realizar innumerables simulaciones en n-dimensiones a través de una interfaz amigable, permitiendo a través de algoritmos y herramientas informáticas el análisis de la información de todas las simulaciones realizadas. La aplicación Web que se va a diseñar no sólo debe centrarse en problemas y enfermedades del sistema inmune, sino que incluirá cualquier problema biológico que pueda ser modelado mediante Sistemas de Ecuaciones Diferenciales.

Capítulo 1: En este capítulo se hace un análisis de las tendencias en la simulación de Sistemas Biológicos, objeto de estudio de la investigación. Describe las características de las diferentes aplicaciones de software desarrolladas en el mundo para el estudio de los Sistemas Biológicos. Se establece la metodología de trabajo, las herramientas y técnicas que se usarán en la investigación.

Capítulo 2: En este capítulo se describen los requisitos funcionales y no funcionales, los actores que intervienen en el sistema, los casos de uso y sus descripciones. Se describe la solución propuesta mediante el modelo de dominio teniendo en cuenta la definición de las entidades y los conceptos principales, así como la representación gráfica de dicho modelo utilizando los componentes de la metodología RUP.

Capítulo 3: En este capítulo se describen a través del modelo de análisis los requisitos funcionales de forma tal que facilite una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a la estructura general del sistema. Se desarrolla el modelo de diseño creando una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos, interfaces y clases.

CAPÍTULO 1: Fundamentación teórica.

En este capítulo se hace un análisis de las tendencias en la simulación de Sistemas Biológicos, objeto de estudio de la investigación. Describe las características de las diferentes aplicaciones de software desarrolladas en el mundo para el estudio de los Sistemas Biológicos. Se establece la metodología de trabajo, las herramientas y técnicas que se usarán en la investigación.

1.1 La Biología de Sistemas.

Una de las ciencias más jóvenes, asociada a la Biotecnología, es la Biología de Sistemas. A diferencia de los métodos clásicos de estudio usados por los biólogos, la Biología de Sistemas se centra en estudiar Sistemas Biológicos en su totalidad, usando herramientas de modelación, comparación y simulación.

¿Qué es la Biología de Sistemas?

Intenta describir los mecanismos que regulan la vida de los seres vivos de forma que, conocida la descripción, se pueda predecir que 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. [1]

Se puede predecir cómo responde un organismo ante determinadas enfermedades, conocer cuáles son los mecanismos que funcionan de forma alterada y tratar de mejorar así el estado de cualquier patología.

La Biología de Sistemas ha dejado su fase de retraso detrás, y ahora se está desarrollando como una nueva disciplina científica que permite la solución de un número cada vez mayor de problemas biológicos.

¿Cuáles son las aplicaciones de esta disciplina?

- En las cardiopatías, la Biología de Sistemas intenta recoger todo el conocimiento sobre esta patología que ha atesorado el bioquímico, el inmunólogo, el médico para encontrar nuevas terapias de prevención y curación.

- En la gota, sólo se dispone de un fármaco, si se estudia bien la ruta metabólica, se podrían crear nuevos fármacos con menos efectos secundarios.
- En el cáncer del hígado, la Biología de Sistemas pretende descubrir cómo una célula del hígado se transforma en una célula que provoca un tumor, modificando el funcionamiento de las células para paralizar el crecimiento del tumor.

La Biología de Sistemas es una disciplina que contribuye a la solución de los problemas biológicos, permitiendo integrar el método reduccionista con las técnicas computacionales más avanzadas logrando mayores resultados en un menor tiempo.

1.2 Modelos matemáticos. Aplicaciones.

La Biología de Sistemas hace un uso intensivo de la modelación matemática en los problemas que abarca, ya que el estudio de un Sistema Biológico se hace más factible si somos capaces de crear un modelo matemático que lo describa.

¿Qué es 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.

La modelación matemática es una de las herramientas que se utilizan 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 de investigación.

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. Los pasos básicos en la formulación de un modelo incluyen la conceptualización, realización y solución del mismo; cada paso debe ser corroborado, con lo cual se valida el modelo. [2]

Para la modelación de Sistemas Biológicos se puede hacer uso de distintos tipos de modelos matemáticos. Dentro de los más utilizados se destacan los autómatas celulares y la dinámica de población. Un Autómata Celular (AC) es un arreglo D-dimensional con un Autómata de Estados Finitos (AEF) en cada posición o celda del arreglo. La dinámica de población permite ver la evolución en el espacio y en el tiempo de un Sistema Biológico, teniendo en cuenta diversos factores, como la dispersión, densidad y crecimiento.

En la investigación se utiliza la dinámica de población, ya que puede expresar los Sistemas Biológicos con Ecuaciones Diferenciales Ordinarias (EDO) con condiciones iniciales.

1.3 La simulación de Sistemas Biológicos.

Existen diferentes definiciones de Simulaciones de Sistemas Biológicos.

Robert Shannon define: *"La simulación es el diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentalmente con este modelo con el propósito de entender el comportamiento del sistema del mundo real o evaluar varias estrategias con los cuales puedan operar el sistema"*.

Thomas H. Naylor define: *"Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo"*.

Se asume en la investigación la definición expresada por Thomas H. Naylor pues a través de esta técnica numérica se puede 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. Incluye tanto la construcción del modelo como su uso analítico para estudiar el problema. La simulación es una técnica muy

poderosa y ampliamente usada en las ciencias para analizar y estudiar sistemas complejos. Ofrece la posibilidad de comprimir el tiempo, esfuerzo y cantidad de recursos necesarios para tomar decisiones.

El modelado “in silico” (en oposición a “in vitro” [en vidrio] o “in vivo” [en vivo]) permite que los investigadores predigan los efectos de los fármacos en el cuerpo humano. La simulación de Sistemas Biológicos permite a las compañías farmacéuticas reducir de manera significativa el número de experimentos de laboratorios requeridos para identificar posibles objetivos de los fármacos. Son muchas las ventajas de estos nuevos métodos de investigación biológica, casi todas ellas se derivan del hecho de que los tiempos requeridos para realizar un experimento son sumamente pequeños.

Resumiendo, la simulación de Sistemas Biológicos:

- Es un proceso relativamente eficiente y flexible.
- Acelera el descubrimiento de posibles drogas.
- Minimiza los tiempos de obtención de un nuevo producto (2-3 años).
- Reduce los costos de investigación y producción.
- Elimina los problemas éticos vinculados al uso de humanos como animales de laboratorio.
- Brinda facilidades para variar parámetros y condiciones experimentales.

La principal desventaja de estos métodos de simulación es que se requiere de un personal altamente preparado en el tema.

Si se unificaran todos los algoritmos necesarios para realizar simulaciones de Sistemas Biológicos se requeriría de una gran potencia de cálculo y recursos computacionales, por lo que sólo pocas entidades las podrían realizar.

De aquí, la necesidad de agrupar algoritmos utilizados en los estudios de Sistemas Biológicos y facilitar, a las instituciones que no los posean, el potencial de cálculo necesario en los análisis que así lo requieran.

Para solucionar este problema se desea diseñar una aplicación Web que integre un grupo de algoritmos utilizados en la simulación de Sistemas Biológicos y gestione la información de las simulaciones realizadas sobre dichos sistemas.

La simulación es una vía para que el investigador pueda comprender mejor el funcionamiento de los Sistemas Biológicos, permitiendo un mejor análisis de los mismos, y así obtener resultados más eficientes en sus investigaciones.

1.4 Tendencias de las aplicaciones de software para la simulación de Sistemas Biológicos.

Limitaciones.

En la actualidad a nivel internacional se han desarrollado varias aplicaciones de software para el estudio de Sistemas Biológicos:

ImmunoGrid

ImmunoGrid es una implementación virtual del sistema inmune humano usando tecnología Grid. Simula procesos inmunes. 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. Hace uso de bases de datos para almacenar, manipular y modelar datos inmunológicos, lo que facilita la búsqueda de modelos matemáticos y predictivos.

Limitaciones: Se centra sólo en problemas y enfermedades del sistema inmune. Los tipos de modelos matemáticos son complejos, hay que definirlos solamente en formato de texto plano.

La aplicación Web que se diseñará no sólo debe centrarse en problemas y enfermedades del sistema inmune, sino que debe incluir cualquier problema biológico que pueda ser modelado mediante sistemas de ecuaciones diferenciales.

Cellware

Cellware no sólo se ha diseñado para conducir la modelación y simulación de los caminos reguladores y metabólicos del gen sino también para ofrecer un ambiente integrado para la diversidad de representaciones matemáticas, la valoración del parámetro y la optimización. Además, tiene una exhibición gráfica y una capacidad de uso fácil para trabajar con modelos grandes y complejos que serían proporcionadas por defecto.

Limitaciones: No hace análisis exhaustivo de las simulaciones realizadas, es decir, aunque permite realizar múltiples análisis, la interpretación de los resultados queda en manos de los usuarios.

La aplicación no sólo debe conducir la modelación y simulación sino que además permitirá realizar análisis profundos de los Sistemas Biológicos que se simulan, facilitando el proceso de interpretación de los resultados por parte de los investigadores.

Webcell

Webcell es un ambiente de la simulación integrado para manejar la información cuantitativa y cualitativa sobre las redes celulares, y para explorar su estado-sostenido y las conductas dinámicas interactivamente sobre la red.

Una interfaz amigable, les permite eficazmente a los usuarios crear, visualizar, simular y guardar sus modelos de red de reacción, mientras se facilita la modelación cinética y la simulación de Sistemas Biológicos de interés.

Los métodos de análisis apoyados para tales modelos incluyen, el análisis de la senda estructural, el análisis del mando metabólico (MCA), análisis de conservación y simulación dinámica.

Limitaciones: Se centra solo en la dinámica del sistema no permitiendo realizar otro tipo de análisis.

La aplicación que se diseñará no sólo mostrará la dinámica del sistema sino que permitirá al usuario realizar innumerables simulaciones a través de una interfaz amigable, y permitirá a través de algoritmos ya implementados, el análisis de todas las simulaciones realizadas.

Vcell

Vcell permite crear modelos biológicos del cual se generará un código matemático que 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 84 servidores. También permite realizar nuevas representaciones geométricas si el usuario lo desea. Los modelos pueden ser rehusados, actualizados y publicados para el uso de otros usuarios, o compartidos privadamente.

Los resultados de las simulaciones son guardados en una base de datos y se pueden exportar en varios formatos. Permite además seleccionar qué parámetros obtener de la simulación de sus combinaciones y estos parámetros pueden ser variados seleccionando los valores o poniendo un rango de números y la cantidad de números a escoger dentro de dicho rango.

Limitaciones: No realiza análisis de los resultados.

La aplicación que se diseñará permitirá al investigador realizar análisis de las simulaciones.

En la actualidad la simulación de Sistemas Biológicos requiere de grandes recursos computacionales y es un campo en el que las aplicaciones que se han desarrollado tienen muchas limitaciones, se considera necesario diseñar una aplicación Web que integre un grupo importante de algoritmos utilizados en el estudio de Sistemas Biológicos y ponga en manos de los investigadores las herramientas de simulación y análisis necesarias para una adecuada comprensión de este tipo de sistemas. Teniendo en cuenta todas las limitaciones de las aplicaciones de software desarrolladas es necesario que la aplicación Web que se desea diseñar permita hacer simulaciones de Sistemas Biológicos en n-dimensiones y además realizar análisis profundos a partir de la información de los Sistemas Biológicos que se simulen.

1.5 Metodologías, herramientas y técnicas.

Para la selección del lenguaje de programación, así como la utilización de los diversos componentes; se consultaron diferentes características como portabilidad, multiplataforma y facilidad de uso, resultando seleccionados los que más se acercan a los requerimientos de la plataforma.

Metodologías de desarrollo.

La creciente informatización de los procesos productivos y sociales provoca que las organizaciones y empresas requieran cada vez más de aplicaciones de software confiables y de calidad tanto en su desarrollo como en su mantenimiento. Es por ello que en los últimos años se han venido publicando estándares, notaciones y procesos de desarrollo de Software que establecen las buenas prácticas para el desarrollo de aplicaciones de software.

La Programación Extrema (XP) es una metodología ligera de software que se basa en la simplicidad, la comunicación, realimentación o reutilización del código desarrollado. Es una de las metodologías de desarrollo de software utilizada para proyectos de corto plazo

Rational Unified Process (RUP) unifica los mejores elementos de metodologías anteriores, está preparado para desarrollar grandes y complejos proyectos. Es un proceso de desarrollo orientado a objetos, dirigido por casos de uso que reflejan lo que los usuarios futuros necesitan y desean, centrado en la arquitectura mostrando la visión común del sistema completo, es iterativo e incremental, donde cada fase se desarrolla en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. RUP utiliza UML (Unified Model Language), como lenguaje de representación visual.

Como la metodología XP es para proyectos cortos y simples y la investigación es compleja y a largo plazo se ha seleccionado para el desarrollo de la investigación la metodología de desarrollo RUP. Además esta metodología de desarrollo utiliza UML el cual es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de aplicaciones de software. UML se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Este lenguaje nos va a ser de mucha utilidad a la hora de representar los diagramas para el desarrollo de la aplicación Web.

De los flujos de trabajo que propone RUP, en la investigación se desarrollan los siguientes: Modelado del Negocio, Requerimientos y Análisis y Diseño. Para cada uno de estos flujos de trabajo RUP propone la realización de varios artefactos, de los cuales se desarrollan en la investigación los siguientes:

- *Modelado del Negocio: Modelo de Casos de Uso del negocio (Modelo Conceptual), Glosario del negocio.*
- *Requerimientos: Modelo de caso de uso, Glosario, Actor, Caso de uso, Especificación de Requerimientos del Software.*
- *Análisis y Diseño: Clase del Análisis, Clase del Diseño, Modelo de Análisis, Modelo de Diseño, Realización de Caso de uso, Prototipo de interfaz de usuario, Modelo de Datos. Descripción de la Arquitectura. [\(Ver Anexo 1\)](#)*

¿Rational Rose o Visual Paradigm?

Rational Rose brinda soporte a Unified Modeling Language (UML). Es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Ofrece un lenguaje de modelado común que agiliza la creación del software. Es un software propietario.

Visual Paradigm es una herramienta CASE que utiliza "UML": como lenguaje de modelación. Visual Paradigm-UML soporta los últimos estándares de anotaciones de JAVA y UML, provee soporte para la generación de código y la ingeniería inversa para Java. Además, Visual Paradigm -UML se integra con Eclipse, Borland® JBuilder®, NetBeans IDE/Sun™ ONE, IntelliJ IDEA™, Oracle JDeveloper y BEA WebLogic Workshop™ para soportar las fases de implementación en el desarrollo de software. Tiene dentro de sus características que es multiplataforma, portable y posee gran facilidad de uso.

Como Rational Rose es una herramienta CASE de software propietario y no genera código en PHP decidimos utilizar el Visual Paradigm porque es una herramienta CASE de software libre, de fácil integración con PHP que es el lenguaje que se utilizará para desarrollar la aplicación Web.

¿ASP o PHP?

Active Server Pages (ASP) es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). ASP realiza muchas tareas apoyándose en objetos que deben ser comprados a ciertas empresas especializadas, o en algunos casos programados por el servidor. La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Es una plataforma que no se ha desarrollado como lo esperaba Microsoft.

PHP (acrónimo recursivo de "PHP: Hypertext Preprocessor", originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un lenguaje del lado del servidor, que se utiliza básicamente para generar páginas dinámicas y que facilita de forma sencilla el acceso a diferentes bases de datos. También se utiliza para generar imágenes, generar pdf. PHP lo desarrollan programadores Web introduciendo nuevas funciones en las versiones mejoradas del lenguaje.

Ventajas de PHP:

- *Libre y Abierto (Código fuente disponible, es gratuito).*
- *Multiplataforma: inicialmente fue diseñado para entornos UNIX por lo que ofrece más prestaciones en este sistema operativo, pero es perfectamente compatible con Windows. ASP, sólo está orientado a sistemas Windows.*
- *Soporte para varios servidores Web.*
- *Fácil acceso a Bases de Datos.*
- *Mucha documentación (Ejemplos, manuales).*
- *Presenta una integración perfecta entre Apache-PHP-MySQL.*
- *Posee una sintaxis bastante clara.*
- *Fácil aprendizaje.*
- *Seguro.*
- *PHP está orientado a objetos. [3]*

Como ASP es un lenguaje que ha sido comercializado con IIS se decidió utilizar PHP porque presenta una compatibilidad total con Apache, servidor que se va a utilizar.

Zend Studio

Se trata de un programa orientado a desarrollar aplicaciones Web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para trabajar con toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

El editor

La parte del programa que nos permite escribir los scripts es bastante útil para la programación en PHP. La interfaz está compuesta por varias partes, en las que encontramos un explorador de archivos, los menús, una ventana de depuración, y otra para mostrar el código de las páginas.

Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Otras ayudas que ofrece a la hora de escribir son las típicas en editores avanzados, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, identificación automática del código, entre otras funciones.

Sin embargo, si nos ponemos exigentes, las funciones de edición podrían perfeccionarse aun más, aspectos que deben resolverse en próximas versiones.

En la parte novedosa de edición que no se había probado en otros programas están los Bookmarks, que nos permiten dirigirnos rápidamente a un punto exacto del código de cualquier archivo. De todos modos, es importante decirlo, las cualidades de Zend Studio que seguro cautivarán a los programadores a los que va dirigido no se deben buscar en la parte de edición sino en las posibilidades de gestión de proyectos y depuración.

La herramienta de depuración

Sin duda más de una vez los programadores de PHP se han visto en problemas por no encontrar un error en algún script que está haciendo que devuelva resultados inesperados. En estos casos lo que se suele hacer es escribir el contenido de diversas variables en la página Web.

Para facilitar el trabajo Zend Studio dispone de una herramienta muy interesante de debug o depuración. Gracias a ella se pueden ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión. Se pueden colocar puntos de parada de los scripts y realizar las acciones típicas de depuración.

Además de la ventana para visualizar el contenido de las variables, dispone de otras donde muestra la salida del script según se va generando, y otra donde se pueden ver las alertas y errores. Las posibilidades se completan con distintos tipos de depuración, en local, en remoto o a partir de una URL.

¿Por qué Dreamweaver?

Dreamweaver es la herramienta líder de desarrollo de aplicaciones Web, es la opción profesional para crear sitios Web y aplicaciones, dado que proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, características que permiten a los desarrolladores y diseñadores más o menos expertos crear rápidamente sitios Web y aplicaciones basados en estándares internacionales.

Desde el avanzado soporte de diseño basado en hoja de estilo (CSS) a las funciones de codificación manual, Dreamweaver proporciona las herramientas profesionales que requiere un entorno integrado y agilizado. Los desarrolladores pueden utilizar Dreamweaver con su tecnología de servidor preferida para crear potentes aplicaciones en Internet destinadas a conectar a los usuarios a las bases de datos, las fuentes de datos dinámicos y los sistemas heredados. Es compatible con todas las principales tecnologías de servidor como, por ejemplo, ColdFusion, PHP, ASP, ASP.NET y JSP, para que los desarrolladores, más o menos expertos, puedan dar vida a sus diseños.

¿Qué es CSS?

CSS es la hoja de estilo en cascada, es un lenguaje Web que controla la presentación y posición del contenido basado en la Web. El formato CSS es la forma recomendada para separar la presentación del contenido. Con CSS, los diseñadores pueden crear hojas de estilo que definen la forma en que aparecen determinados elementos como los encabezamientos y los vínculos. Estas hojas de estilo pueden aplicarse a cualquier página Web. Las ventajas de trabajar con CSS son las siguientes:

- Simplificación: Esta separación entre presentación y contenido simplifica el mantenimiento, porque al modificar un dato en una hoja de estilo se actualiza de manera integral.
- Expresividad: Los lenguajes de la Web, como HTML, no han evolucionado significativamente desde finales de los noventa, pero la creatividad de nuestros clientes tiene fronteras. En los últimos años, los navegadores han seguido desarrollando su soporte para CSS, lo que permite

a los diseñadores y desarrolladores de la Web explorar nuevos horizontes e impulsar su creatividad mediante CSS.

¿SQL Server o MySQL?

SQL Server es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con un interface intuitivo y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET, además incorpora un modelo de objetos totalmente programable (SQL-DMO) con el que podemos desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, hacer aplicación para crear bases de datos, tablas, DTS, backups, entre otros, todo lo que se puede hacer desde el administrador del SQL Server.

Existen diferentes arquitecturas para los sistemas de gestión de bases de datos, pero la más extendida, y la que más éxito ha tenido, es la arquitectura relacional MySQL es un servidor de bases de datos relacionales muy rápido y robusto. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

MySQL fue publicado bajo la licencia GPL (GNU Public License) y mantenido por la compañía sueca MySQL AB que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Este gestor se creó con la rapidez en mente, de modo que no tiene muchas de las características de los gestores comerciales más importantes, como Oracle, Sybase o SQL Server. No obstante, eso no ha impedido que sea el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesiten de características muy avanzadas, como es el caso de las aplicaciones Web y por tanto de la propuesta de solución que se ofrece con este trabajo debido a su gran rapidez y facilidad de uso.

Como SQL Server es un sistema gestor de base de datos de software propietario se decidió usar MySQL porque es un de software libre y se trabaja con mucha más facilidad, además tiene una fácil integración con PHP.

¿IIS o Apache?

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP.

Este servicio convierte a un computador en un servidor de internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

El servidor Apache es un software (libre) servidor HTTP de código abierto para plataformas UNIX, Windows entre otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache es de acuerdo al estudio hecho por Netcraft, el servidor WWW más utilizado del momento. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Apache ha demostrado ser sustancialmente más rápido que muchos otros servidores libres y compite de cerca con los mejores servidores comerciales, es el servidor Web por excelencia.

Entre sus características más sobresalientes están:

1. **Fiabilidad:** Alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.
2. **Gratuidad:** Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.
3. **Extensibilidad:** Se pueden añadir módulos para ampliar las ya de por sí amplias capacidades de Apache. Hay una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Python), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, hasta crear servidores virtuales por IP o por nombre (varias direcciones Web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos incluso pueden ser creados por cualquier persona con conocimientos de programación.

Como IIS es un servidor Web propiedad de Microsoft Corporation se decidió usar Apache, que es desarrollado en Software Libre.

Arquitectura Cliente-Servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas

- Aumento de la productividad:
 - Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos.
 - Mediante la integración de las aplicaciones cliente/servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes.
 - Una interfaz gráfica de usuario consistente reduce el tiempo de aprendizaje de las aplicaciones.
- Menores costos de operación:
 - Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores (habitualmente caros) y dispositivos periféricos (como impresoras) entre máquinas clientes permite un mejor rendimiento del conjunto.
 - Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él y de la ubicación de la información.
 - El movimiento de funciones desde un ordenador central hacia servidores o clientes locales origina el desplazamiento de los costos de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.

➤ Mejora en el rendimiento de la red:

- Las arquitecturas cliente/servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.
- Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones.
- En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.
- Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los ordenadores personales que antes estuvieran aislados.

Web Services

Es un sistema software diseñado para soportar la interoperabilidad máquina - máquina a través de una red. Este tiene una interfaz descrita en un formato que puede ser procesado por una máquina. Otros sistemas interactúan con el Web Service utilizando mensajes SOAP los cuales se encuentran establecidos previamente.

SOAP es un protocolo elaborado para facilitar la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web. SOAP se puede implementar fácilmente en casi cualquier ambiente de programación. Actualmente, existen diversos paquetes de desarrollo SOAP para diversos sistemas operativos y lenguajes de alto nivel. [5]

Entonces se podría decir que un Web Service es una comunicación por medio de mensajes SOAP entre diferentes equipos a través de una red.

Los Web Services representan los principios sobre los cuales se construirá la próxima generación de arquitecturas de e-business, presentando una evolución desde sistemas orientados a objetos en sistemas basados en servicios. Un Web Service es una aplicación modular, contenida en sí misma y

accesible a través de protocolos Web. Los Web Services combinan lo mejor de los desarrollos basados en componentes y la Web. Desde el punto de vista de componentes, un Web Service no tiene que revelar cómo está implementado. Sólo es necesario saber cómo accederlo y utilizar los servicios que él provee. Desde el punto de vista de la Web, son accesibles protocolo de transferencia de hipertexto (HTTP) completamente independientes de proveedores, plataformas y lenguajes, utilizando formatos de datos universalmente aceptados (XML). [6]

Modelo de acceso a Web Services

Están establecidos un conjunto de requerimientos que deben cumplirse para brindar un entorno de Web Services de forma tal que permita explotar óptimamente sus ventajas:

- El Web Service debe ser creado y sus interfaces y métodos de invocación deben ser definidos.
- Debe ser publicado en repositorios de intranet o de Internet para que los potenciales usuarios los localicen.
- Debe ser localizado para poder ser invocado.
- Debe ser invocado por el usuario cliente.
- Un Web Service puede ser dado de baja del repositorio si deja de estar publicado.

Una arquitectura de Web Services requiere tres operaciones fundamentales: publicar, encontrar y utilizar.

Conclusiones del Capítulo

En este capítulo se ha demostrado la necesidad de diseñar una aplicación Web para mejorar la eficacia en las investigaciones de los Sistemas Biológicos. La aplicación Web que se diseñará no sólo mostrará la dinámica del sistema sino que permitirá al usuario realizar innumerables simulaciones en n-dimensiones a través de una interfaz amigable, y permitirá a través de algoritmos y herramientas informáticas, el análisis de la información de todas las simulaciones realizadas. La aplicación Web que se va a diseñar no sólo debe centrarse en problemas y enfermedades del sistema inmune, sino que incluirá cualquier problema biológico que pueda ser modelado mediante Sistemas de Ecuaciones Diferenciales.

Para el desarrollo de la aplicación de software se han seleccionado la metodología de desarrollo Rational Unified Process (RUP), así como las diferentes herramientas y técnicas desarrolladas en software libre:

Lenguajes

- PHP

Herramientas

- Visual Paradigm
- Zend Studio
- Dreamweaver
- MySQL
- Apache

CAPÍTULO 2: Características del Sistema.

En este capítulo se describen los requisitos funcionales y no funcionales, los actores que intervienen en el sistema, los casos de uso y sus descripciones. Se describe la solución propuesta mediante el modelo de dominio teniendo en cuenta la definición de las entidades y los conceptos principales, así como la representación gráfica de dicho modelo utilizando los componentes de la metodología RUP.

2.1 ¿Modelo de negocio o del dominio?

Existen como mínimo dos aproximaciones para expresar el contexto de un sistema en una forma utilizable para desarrolladores de software: modelado del dominio y modelado del negocio.

Un Modelo de Negocio es una técnica para comprender los procesos de negocio de la organización son funciones que se desarrollan en el ambiente o entorno que definimos como negocio. El objetivo del modelado del negocio es describir los procesos (existentes u observados) con el propósito de comprenderlos. El modelo del negocio especifica que procesos del negocio soportará el sistema.

Un Modelo del Dominio describe los conceptos importantes del contexto como objeto del dominio y enlaza estos objetos unos con otros. Captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. [7] [Ver Anexo 2.](#)

En la investigación se decidió desarrollar el modelo del dominio porque se hace difícil encontrar la definición de procesos del negocio que tienen que ver con el objeto de estudio de la investigación, se necesita describir mediante una serie de conceptos y sus relaciones agrupados en un modelo del dominio para un fácil entendimiento de la aplicación.

2.2 Definición de las entidades y los conceptos principales

La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema.

Con la aplicación del Modelo del dominio se detectaron las siguientes entidades y conceptos (objetos):

- Centro de investigación: Institución donde se desarrolla el proyecto.

- Investigador: Persona que dirige el proyecto.
- Proyecto: Tema sobre el cuál se está investigando.
- Sistema Biológico: Conjuntos de entes biológicos que interactúan entre sí y funcionan como un todo íntegro.
- 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.
- Simulación: Técnica para reproducir artificialmente un fenómeno o el funcionamiento de un sistema, utilizando modelos matemáticos y lógicos específicos para cada proceso.

Diagrama Modelo del Dominio

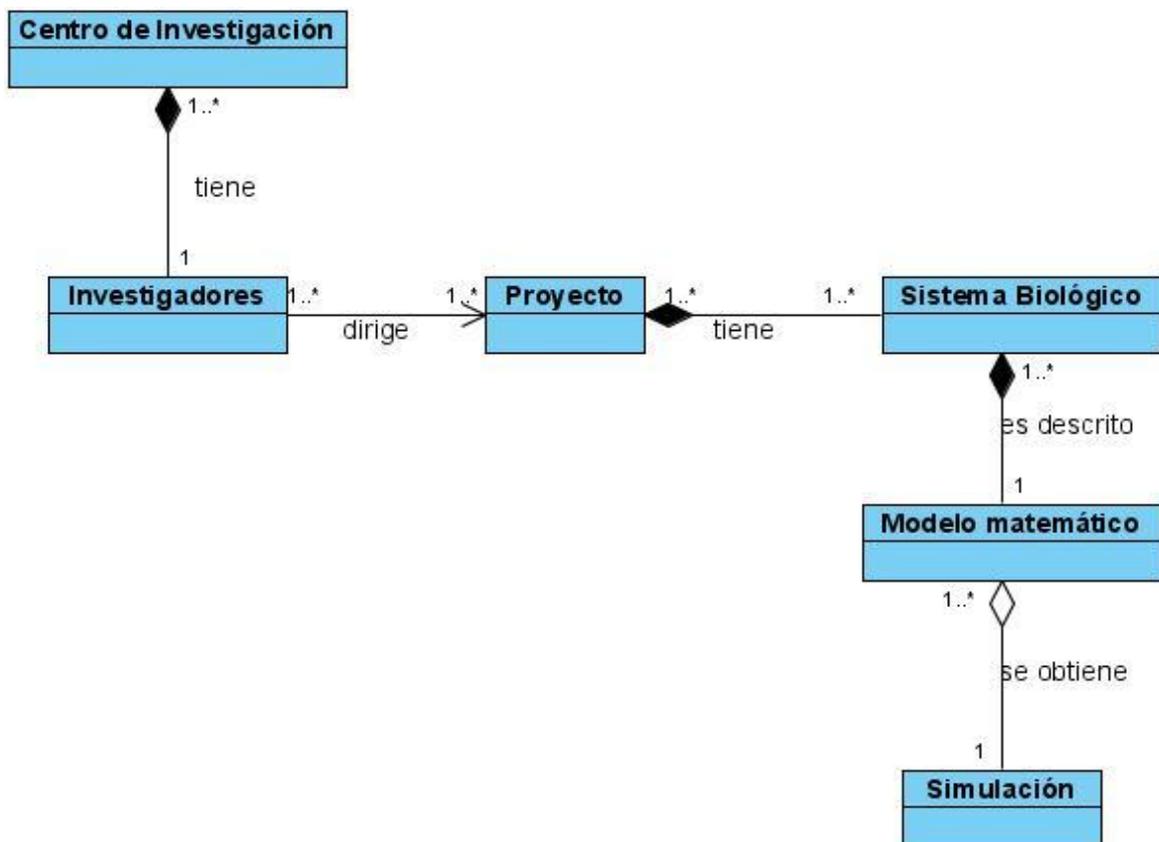


Figura 2.1: Diagrama modelo de Dominio

2.3 Requerimientos

2.3.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Ellos permiten determinar, de una manera clara, responsabilidades que se propone el sistema.

1. Autenticar Usuario
 - 1.1 Comparar usuario y contraseña con los usuarios del sistema.
 - 1.2 Asignar privilegios al perfil del usuario.
2. Registrar Usuario
3. Solicitar Baja
 - 3.1 Enviar mensaje de solicitud de eliminación de un usuario.
4. Administrar Usuario
 - 4.1 Permitir aprobar la solicitud de registro de los datos del usuario.
 - 4.2 Buscar los datos del usuario.
 - 4.3 Modificar los datos del usuario.
 - 4.4 Eliminar usuario.
5. Gestionar Proyecto
 - 5.1 Permitir enviar planilla de solicitud para la creación de un proyecto.
 - 5.2 Permitir aprobar la creación de un proyecto.
 - 5.3 Buscar proyecto.
 - 5.4 Modificar proyecto
 - 5.5 Eliminar proyecto.
 - 5.6 Permitir enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto.
6. Gestionar Sistema Biológico
 - 6.1 Insertar sistema biológico.
 - 6.2 Buscar sistema biológico.

- 6.3 Modificar sistema biológico.
- 6.4 Eliminar sistema biológico.
- 7. Gestionar Modelo Matemático
 - 7.1 Insertar modelo matemático.
 - 7.2 Buscar modelo matemático.
 - 7.3 Modificar modelo matemático.
 - 7.4 Eliminar modelo matemático.
- 8. Gestionar Simulación
 - 8.1 Realizar simulación dado un modelo matemático.
 - 8.2 Buscar simulación.
 - 8.3 Mostrar un listado de las simulaciones.
 - 8.4 Eliminar simulación.
- 9. Administrar Tutoriales
 - 9.1 Insertar tutoriales.
 - 9.2 Buscar tutoriales.
 - 9.3 Publicar tutoriales
 - 9.4 Eliminar tutoriales.
- 10. Administrar Foro de Discusión
 - 10.1 Insertar tema a debatir.
 - 10.2 Insertar preguntas sobre un tema.
 - 10.3 Insertar respuesta a pregunta.
 - 10.4 Buscar.

2.3.2 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características del producto.

Apariencia o interfaz externa:

La aplicación será diseñada con una interfaz amigable y fácil de usar por el usuario.

Usabilidad:

El sistema podrá ser usado por cualquier usuario que conozca el manejo de la computadora y navegue en un ambiente Web.

Rendimiento:

Al estar concebida para un ambiente cliente/servidor, se trata de garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios así como la velocidad de procesamiento de la información. Para garantizar ésto se realiza la validación de los datos y la manipulación de eventos en el cliente y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.

Portabilidad:

El sistema puede ser ejecutado sobre los sistemas operativos Linux y Windows, por su característica de ser multiplataforma.

Seguridad:

El administrador tendrá acceso total de la aplicación pero los demás usuarios tendrán acceso solo al nivel adecuado según el rol ya que las funcionalidades del sistema se muestran de acuerdo al nivel de usuario que esté activo.

Confiabilidad:

El sistema debe ser confiable y preciso en la información que le suministra al usuario para evitar cualquier tipo de error.

Ayuda:

La aplicación debe poseer ayuda, en la que se explique de forma clara el uso de las opciones del sistema para brindarle al usuario cualquier información.

Software:

Se requiere para el uso de la aplicación:

- Sistema operativo Linux o Windows.
- Cualquier navegador Web.
- DirectX
- Lenguaje PHP

Hardware:

Pentium IV de 2.8 GH con 512 MB RAM.

2.4 Patrones de Casos de Uso.

¿Qué es un patrón?

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos al tratar de lograr ciertos objetivos. Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y también describe el núcleo de la solución a ese problema.

En la investigación se utilizaron los siguientes patrones:

- **Patrón CRUD** (Creating, Reading, Updating and Deleting): consiste en un caso de uso, llamado “Información CRUD” o “Administrar Información”, que modela todas las operaciones que se pueden realizar sobre una parte de información de cierto tipo, tal como crearla, leerla, actualizarla y eliminarla. Este patrón debe ser usado cuando todos los flujos contribuyen al mismo valor de negocio.
- **Actores múltiples: Rol común** (Multiple Actors: Common Role): Dos actores juegan el mismo papel hacia el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Este patrón es aplicable cuando, desde el punto de vista de un caso de uso hay solo una entidad externa interactuando con cada instancia del caso de uso.

2.5 Definición de los actores

Actores	Justificación
Usuario	Humano que interactúa con el sistema.
Trabajador del proyecto	Es un usuario previamente registrado que ha sido incluido por parte de un líder de proyecto para que trabaje en el proyecto.
Líder del proyecto	Usuario registrado que crea un proyecto y le pide autorización al administrador para su ejecución. Una vez aprobado el proyecto este usuario puede incluir a otros usuarios para que trabajen en el mismo e invitar a otros a que observen el trabajo. Este usuario es el único autorizado para eliminar el proyecto.
Administrador	Persona que administra todo el sistema permite el registro de usuarios, la creación de proyectos (es decir, revisa que los proyectos que se crean tienen sentido) revisa y autoriza la publicación y eliminación de tutoriales (estos tutoriales pueden ser subidos por cualquier usuario que este registrado en el sistema).

Tabla 2.1: Definición de los actores

Diagrama de actores del sistema

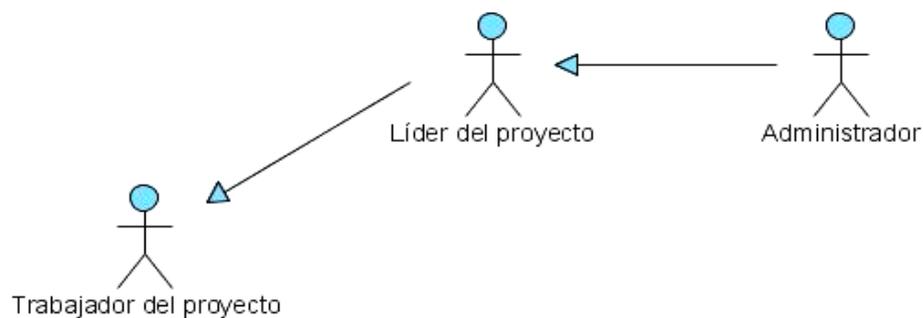


Figura 2.2: Diagrama de actores del sistema

2.6 Diagrama de casos de uso del Sistema.

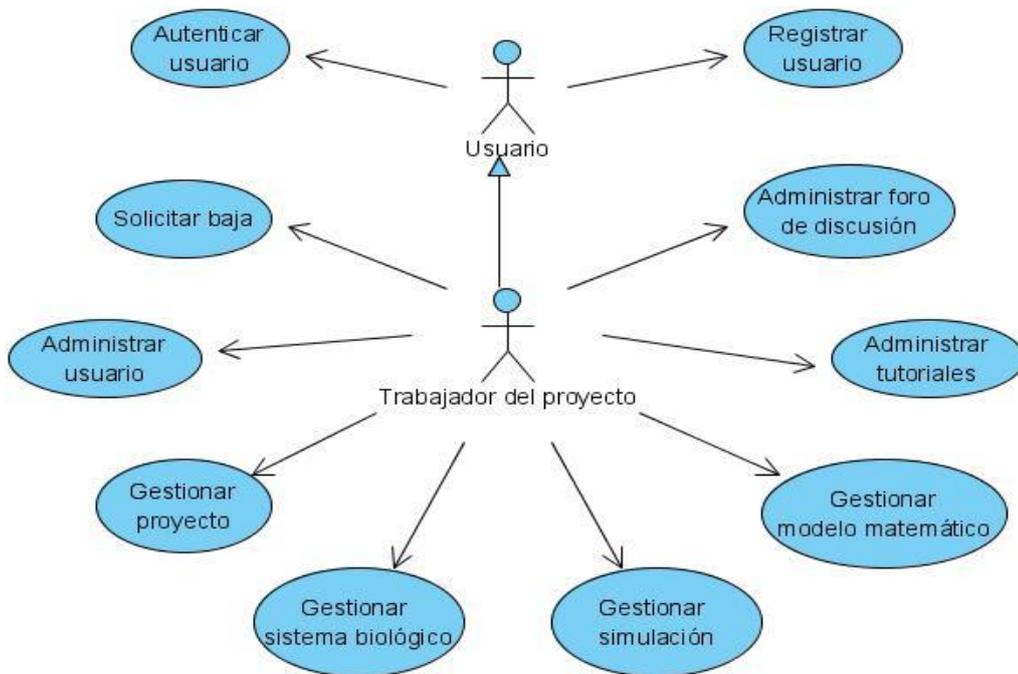


Figura 2.3: Diagrama de Caso de Uso del Sistema

2.7 Descripción de los Casos de Usos del Sistema.

A continuación se muestran las descripciones textuales de los casos de uso críticos del sistema. (Para ver las demás descripciones remitirse al Expediente de Proyecto).

Administrar usuario

Nombre del CU	Administrar usuario
Actores	Trabajador del proyecto (inicia)
Propósito	Permitir Aprobar la solicitud de registro de los datos del usuario, buscar los datos del usuario, modificar los datos del usuario, eliminar usuario.
Resumen	El Caso de Uso se inicia cuando se solicitan cualesquiera de las siguientes opciones:

	<ul style="list-style-type: none"> • Aprobar la solicitud de registro: El administrador aprueba la solicitud de registro de los datos del usuario. • Buscar los datos del usuario: El Trabajado del proyecto provee los parámetros de búsqueda de los datos de un usuario. • Modificar los datos del usuario: El Trabajado del proyecto indica que desea modificar los datos de un usuario, los modifica y el sistema lo guarda en la base de datos. • Eliminar usuario: El administrador indica el usuario que quiere eliminar y el sistema lo elimina de la base de datos. <p>Termina el caso de uso cuando se realiza cualesquiera de las opciones anteriores.</p>
Referencias	R 4.1, R 4.2, R 4.3, R 4.4
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	El sistema aprueba la solicitud de registro de los datos del usuario, busca los datos del usuario, modifica los datos del usuario, elimina usuario.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
<p>1. El Administrador selecciona realizar alguna de las siguientes opciones:</p> <ul style="list-style-type: none"> - Permitir aprobar la solicitud de registro de los datos del usuario. - Buscar los datos del usuario. - Modificar los datos del usuario. - Eliminar usuario. 	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> -Si el administrador selecciona la opción de aprobar la solicitud de registro, ir a Sección “Permitir aprobar la solicitud de registro de los datos del usuario”. -Si el Trabajador del proyecto selecciona la opción de buscar los datos del usuario, ir a

	<p>Sección “ Buscar los datos del usuario ”.</p> <p>-Si el Trabajador del proyecto selecciona la opción de modificar los datos de usuario, ir a Sección “Modificar los datos del usuario”.</p> <p>-Si el administrador selecciona la opción de eliminar usuario, ir a Sección “ Eliminar usuario ”.</p>
	3 Termina el caso de uso Administrar usuario.
Sección: Aprobar la solicitud de registro	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que permite aprobar la solicitud de registro de los datos del usuario.
2. El Administrador selecciona la solicitud a revisar.	3. Muestra la solicitud a revisar.
4. Aprueba la solicitud.	5. Emite un mensaje de: “ Su solicitud de registro ha sido aprobada ”.
Curso Alterno: Aprobar la solicitud de registro	
Acciones del Actor	Respuesta del Sistema
	5. Emite un mensaje de: “ Su solicitud de registro ha sido rechazada ”.
Sección: Buscar los datos del usuario	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que le permite buscar los datos del usuario deseado.
2. El Trabajador del proyecto provee el usuario.	3. Busca los datos del usuario en la base de datos.

	4. Muestra el resultado de la búsqueda.
	5. Ir a la acción 1.
Curso Alterno: Buscar los datos del usuario	
Acción del actor	Respuesta del Sistema
	3. El usuario no está registrado en la base de datos.
	4. Emite un mensaje de: " No está registrado en la base de datos "
Sección: Modificar los datos del usuario	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz correspondiente para modificar los datos del usuario.
2. El Trabajador del proyecto modifica los datos deseados.	3. Guarda los datos del usuario.
Sección: Eliminar usuario	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz correspondiente para eliminar usuario.
2. El Administrador provee el usuario.	
	3. Busca y elimina el usuario indicado de la Base de datos.
	4. Ir a la acción 1.
Prioridad	Crítico
Prototipo de interfaz	Ver Anexo 3

Tabla 2.2: Descripción del Caso de Uso Administrar usuario.

Gestionar Proyecto

Nombre del CU	Gestionar Proyecto.
Actores	Trabajador del proyecto (Inicia).
Propósito	Permitir enviar planilla de solicitud para la creación de un proyecto, aprobar la creación de un proyecto, buscar, modificar y eliminar proyecto, enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto.
Resumen	<p>El Caso de Uso se inicia cuando el usuario selecciona Gestionar proyecto, seleccionando una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Permitir enviar planilla de solicitud para la creación de un proyecto: El Líder del proyecto llena una planilla con los datos necesarios para solicitar la creación de un proyecto. • Permitir aprobar la creación de un proyecto: El administrador autoriza la creación de un proyecto una vez revisada la planilla. • Buscar Proyecto: El Trabajador del proyecto solicita buscar un proyecto y el sistema lo muestra en caso de estar registrado. • Modificar proyecto: El Trabajador del proyecto introduce los datos que se van a modificar y el sistema actualiza la base de datos. • Eliminar proyecto: El Líder del proyecto solicita el proyecto que quiere eliminar y el sistema lo elimina de la base de datos, si no está registrado el proyecto muestra un mensaje de: " El proyecto no esta registrado " • Permitir enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto: El líder del proyecto invita a otros usuarios registrados para que

	participen en el proyecto Termina el caso de uso cuando se realiza cualquiera de las opciones anteriores.
Referencias	R 5.1, R 5.2, R 5.3, R 5.4, R 5.5, R 5.6.
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	El sistema permite enviar planilla de solicitud para la creación de un proyecto, aprueba la creación de un proyecto, busca, modifica y elimina proyecto, envía mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
<p>1. El usuario selecciona realizar alguna de las opciones:</p> <ul style="list-style-type: none"> - Permitir enviar planilla de solicitud para la creación de un proyecto. - Permitir aprobar la creación de proyecto. - Buscar proyecto. - Modificar proyecto. - Eliminar proyecto. - Permitir enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto. 	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> - Si el Líder del proyecto decide crear un proyecto, ir a Sección: "Enviar planilla de solicitud para la creación de un proyecto". - Si el Administrador decide aprobar la creación de proyecto, ir a Sección: "Aprobar la creación de proyecto". - Si el Trabajador del proyecto decide buscar proyecto, ir a Sección "Buscar proyecto". - Si el Trabajador del proyecto decide modificar proyecto, ir a Sección "Modificar proyecto".

	<p>- Si el Líder del proyecto decide eliminar proyecto, ir a Sección " Eliminar proyecto".</p> <p>-Si el Líder del proyecto decide invitar a otros usuarios registrados a participar en un proyecto, ir a Sección " Enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto "</p>
	5. Termina el caso de uso Gestionar proyecto.
Sección: Enviar planilla de solicitud para la creación de un proyecto	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que le permite llenar la planilla de solicitud de creación de un proyecto.
2. El Líder del proyecto ingresa los datos necesarios para solicitar la creación de un proyecto.	
3. El Líder del proyecto envía la planilla de solicitud.	4. Verifica que el proyecto no esté registrado en la base de datos.
	5. Incorpora los datos del proyecto en su base de datos.
	6. Emite un mensaje de: "Su solicitud será valorada".
Curso Alternativo: Enviar planilla de solicitud para la creación de un proyecto	
Acciones del Actor	Respuesta del Sistema
	4. Si el proyecto está registrado en la base de datos, emite mensaje de: " Está registrado un

	proyecto con esos datos”.
Sección: Aprobar la creación de proyecto.	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que permite aprobar la creación de un proyecto.
2. El Administrador selecciona las planillas de solicitudes a revisar.	3. Aprueba la solicitud revisada.
	4. Emite un mensaje de: “ El proyecto ha sido creado satisfactoriamente ”.
	5. Ir a la acción 1.
Curso Alterno: Aprobar creación de proyecto	
Acciones del Actor	Respuesta del Sistema
	3. Rechaza la solicitud revisada.
	4. Emite un mensaje de: “ Su solicitud ha sido rechazada ”.
Sección: Buscar proyecto	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que le permite buscar un proyecto.
2. El Trabajador del proyecto provee el nombre del proyecto que desea buscar.	3. Busca el proyecto en la base de datos.
	4. Muestra el resultado de la búsqueda.
5. El Trabajador del proyecto selecciona el resultado deseado.	6. Muestra el resultado seleccionado.

	7. Ir a la acción 1.
Curso Alterno: Buscar proyecto.	
Acciones del Actor	Respuesta del Sistema
	3. Emite un mensaje de: " El proyecto no está registrado en la Base de Datos".
Sección: Modificar proyecto	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que le permite modificar un proyecto.
2. El Trabajador del proyecto selecciona la opción de búsqueda y va a la sección: Buscar proyecto.	
3. El Trabajador del proyecto modifica los datos necesarios.	4. Guarda el proyecto en la Base de Datos.
Curso Alterno: Modificar proyecto	
Acciones del Actor	Respuesta del Sistema
	4. Emite un mensaje de: " Existen errores en los datos ".
Sección: Eliminar proyecto	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que le permite eliminar un proyecto.
2. El Líder del proyecto selecciona el proyecto que desea eliminar.	3. Verifica que sea el proyecto deseado.
	4. Elimina el proyecto de la Base de Datos.

	5. Ir a la acción 1.
Sección: Enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que permite enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto.
2. El Líder del proyecto introduce los datos del usuario que quiere invitar.	3. Verifica que los datos estén correctos.
	4. Envía invitación al usuario indicado.
	5. Muestra mensaje de: " El usuario ha sido invitado a trabajar en su proyecto ".
Curso Alterno: Enviar mensaje de solicitud de invitación a un usuario registrado a participar en un proyecto.	
Acciones del Actor	Respuesta del Sistema
	3. Existen errores en los datos.
	4. Muestra un mensaje de: " Los datos están incorrectos ".
Prioridad	Crítico
Prototipo de interfaz.	Ver Anexo 4

Tabla 2.3: Descripción del Caso de Uso Gestionar Proyecto

Gestionar Sistema Biológico

Nombre del CU	Gestionar Sistemas Biológicos	
Actores	Trabajador del proyecto (inicia)	
Propósito	Permitir insertar un sistema biológico, buscar sistema biológico, modificar sistema biológico y eliminar sistema biológico.	
Resumen	<p>El Caso de Uso se inicia cuando se solicitan cualesquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> • Insertar sistema biológico: El Trabajador del proyecto inserta el sistema biológico deseado. • Buscar sistema biológico: El Trabajador del proyecto busca los Sistemas Biológicos deseados. • Modificar sistema biológico: El Trabajador del proyecto modifica el sistema biológico deseado. • Eliminar sistema biológico: El Líder del proyecto elimina los Sistemas Biológicos deseados. <p>Termina el caso de uso cuando se realiza cualquiera de las opciones anteriores.</p>	
Referencias	R 6.1, R 6.2, R 6.3, R 6.4	
Precondiciones	<p>El usuario debe estar autenticado.</p> <p>Debe existir un proyecto en la base de datos.</p>	
Poscondiciones	El sistema permite insertar un sistema biológico, busca sistema biológico, modifica sistema biológico y elimina sistema biológico.	
Curso Normal de los Eventos		
	Acciones del Actor	Respuesta del Sistema
	1. El Trabajador del proyecto selecciona realizar alguna de las siguientes opciones:	2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:

<ul style="list-style-type: none"> - Insertar sistema biológico. - Buscar sistema biológico. - Modificar sistema biológico. - Eliminar sistema biológico. 	<ul style="list-style-type: none"> - Si el Trabajador del proyecto selecciona la opción de Insertar un sistema biológico, ir a Sección: "Insertar sistema biológico". - Si el Trabajador del proyecto selecciona la opción de Buscar un sistema biológico, ir a Sección: "Buscar sistema biológico". - Si el Trabajador del proyecto selecciona la opción de Modificar sistema biológico, ir a Sección: "Modificar sistema biológico". - Si el Líder del proyecto selecciona la opción de Eliminar sistema biológico, ir a Sección: "Eliminar sistema biológico".
	<p>3 Termina el caso de uso Gestionar sistema biológico.</p>
<p>Sección: Insertar sistema biológico</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1. Muestra la interfaz que permite insertar sistema biológico.</p>
<p>2. El Trabajador del proyecto provee el nombre del sistema biológico y el SBML que desea insertar.</p>	<p>3. Verifica que el sistema no esté registrado en la base de datos.</p>
	<p>4. Incorpora el sistema en su base de datos.</p>
<p>Curso Alternativo: Insertar sistema biológico</p>	
<p>Acción del actor</p>	<p>Respuesta del Sistema</p>
	<p>3. Verificó que el sistema biológico está en la base de datos.</p>

	4. Emite un mensaje de: " El sistema biológico está en la base de datos ".
Sección: Buscar sistema biológico	
Acción del actor	Respuesta del Sistema
	1. Muestra una interfaz que le permite buscar el sistema biológico.
2. El Trabajador del proyecto provee el nombre del sistema biológico.	3. Busca el sistema biológico en la base de datos.
	4. Muestra el resultado de la búsqueda.
5. El Trabajador del proyecto selecciona el resultado deseado.	6. Muestra el resultado seleccionado.
	7. Ir a la acción 1.
Curso Alterno: Buscar sistema biológico	
Acción del actor	Respuesta del Sistema
	3. Emite un mensaje de: " El sistema biológico no está en la base de datos ".
Sección: Modificar sistema biológico	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz correspondiente para modificar un sistema biológico.
2. El Trabajador del proyecto selecciona la opción de búsqueda y va a la sección: Buscar sistema biológico.	
3. El Trabajador del proyecto modifica el sistema biológico.	4. Guarda el sistema biológico modificado.
Sección: Eliminar sistema biológico	

Acción del actor		Respuesta del Sistema
		1. Muestra la interfaz correspondiente para eliminar sistema biológico.
2. El Líder del proyecto selecciona el sistema biológico que desea eliminar.		3. Verifica que sea el sistema biológico deseado.
		4. Elimina el sistema biológico indicado.
		5. Ir a la acción 1.
Prioridad	Crítico	
Prototipo de interfaz	Ver Anexo 5	

Tabla 2.4: Descripción del Caso de Uso Gestionar Sistema Biológico

Gestionar Modelo Matemático

Nombre del CU	Gestionar Modelo Matemático
Actores	Trabajador del proyecto (inicia)
Propósito	Permitir insertar, modificar, buscar y eliminar modelo matemático.
Resumen	<p>El Caso de Uso se inicia cuando se solicitan cualesquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> • Insertar modelo matemático: El Trabajador del proyecto inserta el modelo matemático. • Buscar modelo matemático: El Trabajador del proyecto provee los parámetros de búsqueda de un modelo matemático. • Modificar modelo matemático: El Trabajador del proyecto modifica datos en el modelo matemático. • Eliminar modelo matemático: El Líder del proyecto indica el modelo matemático que quiere eliminar y el sistema lo elimina de la base de datos.

	Termina el caso de uso cuando se realiza cualesquiera de las opciones anteriores.	
Referencias	R 7.1, R 7.2, R 7.3, R 7.4.	
Precondiciones	El usuario debe estar autenticado. Debe existir al menos un sistema biológico en la base de datos.	
Poscondiciones	El sistema permite insertar, modificar, buscar y eliminar modelo matemático.	
Curso Normal de los Eventos		
	Acciones del Actor	Respuesta del Sistema
	<p>1. El Trabajador del proyecto selecciona realizar alguna de las opciones:</p> <ul style="list-style-type: none"> - Insertar modelo matemático. - Buscar modelo matemático. - Modificar modelo matemático - Eliminar modelo matemático. 	<p>2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> - Si el Trabajador del proyecto selecciona la opción de insertar modelo matemático, ir a Sección: "Insertar Modelo matemático". - Si el Trabajador del proyecto selecciona la opción de buscar modelo matemático, ir a Sección: "Buscar Modelo matemático". - Si el Trabajador del proyecto selecciona la opción de modificar modelo matemático, ir a Sección: "Modificar Modelo matemático". - Si el Líder del proyecto selecciona la opción de eliminar modelo matemático, ir a Sección: "Eliminar Modelo matemático".
		3 Termina el caso de uso Gestionar Modelo matemático

Sección: Insertar Modelo matemático	
Acciones del Actor	Respuesta del Sistema
	1. Muestra la interfaz que permite insertar modelo matemático.
2. El Trabajador del proyecto provee: -El Sistema Biológico. -Nombre del modelo matemático - Los parámetros y las variables.	3. Verifica que el modelo matemático esté en el formato correcto
	4. Inserta el modelo matemático en la base de datos.
Curso Alterno: Insertar Modelo matemático.	
Acción del actor	Respuesta del Sistema
	3. Verificó que el modelo matemático no está en el formato correcto.
	4. Emite un mensaje de: " El modelo matemático no está en el formato correcto ".
Sección: Buscar Modelo matemático	
Acción del actor	Respuesta del Sistema
	1. Muestra una interfaz que le permite buscar el modelo matemático.
2. El Trabajador del proyecto provee el nombre del sistema biológico y el nombre del modelo matemático.	3. Busca el modelo matemático en la base de datos.
	4. Muestra el resultado de la búsqueda.

	5. Ir a la acción 1.
Curso Alterno: Buscar Modelo matemático	
Acción del actor	Respuesta del Sistema
	3. El modelo matemático no está registrado en la base de datos
	4. Emite un mensaje de: " No está registrado ningún modelo matemático "
Sección: Modificar Modelo matemático	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz correspondiente para modificar un modelo matemático.
2. El Trabajador del proyecto modifica el modelo matemático.	3. Guarda el modelo matemático modificado.
Sección: Eliminar Modelo matemático	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz para eliminar modelo matemático.
2. El Líder del proyecto selecciona el nombre del sistema biológico y el modelo matemático que desea eliminar.	3. Elimina el modelo matemático indicado.
Prioridad	Crítico
Prototipo de interfaz	Ver Anexo 6

Tabla 2.5: Descripción del Caso de Uso Gestionar Modelo Matemático

Gestionar Simulación

Nombre del CU	Gestionar simulación	
Actores	Trabajador del proyecto (inicia)	
Propósito	Permitir realizar una simulación dado un modelo matemático, mostrar un listado de las simulaciones, buscar simulación y eliminar simulación.	
Resumen	<p>El Caso de Uso se inicia cuando se solicitan cualesquiera de las siguientes opciones:</p> <ul style="list-style-type: none"> • Realizar simulación: El Trabajador del proyecto selecciona la simulación que desea realiza y la realiza dado un modelo matemático • Mostrar un listado de las simulaciones: El Trabajador del proyecto desea ver un listado de las simulaciones realizadas • Buscar simulación: El Trabajador del proyecto busca las simulaciones deseadas. • Eliminar simulación: El Líder del proyecto elimina las simulaciones deseadas. <p>Termina el caso de uso cuando se realiza cualquiera de las opciones anteriores.</p>	
Referencias	R 8.1, R 8.2, R 8.3, R 8.4	
Precondiciones	<p>El usuario debe estar autenticado.</p> <p>Debe existir al menos un modelo matemático en la base de datos.</p>	
Poscondiciones	El sistema permite realizar una simulación dado un modelo matemático, muestra un listado de las simulaciones, buscar simulación y eliminar simulación.	
Curso Normal de los Eventos		
	Acciones del Actor	Respuesta del Sistema
	1. El Trabajador del proyecto selecciona	2. El sistema, en dependencia de la

<p>realizar alguna de las opciones:</p> <ul style="list-style-type: none"> -Realizar simulación dado un modelo matemático. -Mostrar un listado de las simulaciones. -Buscar simulación. -Eliminar simulación. 	<p>operación que solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> - Si el Trabajador del proyecto selecciona la opción de Realizar simulación, ir a Sección “ Realizar simulación ”. - Si el Trabajador del proyecto selecciona la opción de Mostrar un listado de las simulaciones, ir a Sección “ Mostrar un listado de las simulaciones ”. - Si el Trabajador del proyecto selecciona la opción de Buscar simulación, ir a Sección “ Buscar simulación ”. - Si el Líder del proyecto selecciona la opción de Eliminar simulación, ir a Sección “ Eliminar simulación ”.
	<p>3 Termina el caso de uso Gestionar simulación.</p>
<p>Sección: Realizar simulación</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1. Muestra la interfaz correspondiente para seleccionar el tipo de simulación que desea realizar.</p>
<p>2. El Trabajador del proyecto selecciona el tipo de simulación que desea realizar,.</p>	<p>3. Muestra al interfaz correspondiente al tipo de simulación.</p>
<p>4. El Trabajador del proyecto inserta los datos necesarios para la realización de</p>	<p>5. Realiza la simulación.</p>

las simulaciones.	
	7. Muestra una interfaz con la simulación realizada.
Curso Alterno: Realizar simulaciones	
Acción del actor	Respuesta del Sistema
4. Si el Trabajador del proyecto inserta los datos incorrectamente.	5. Emite un mensaje de: " Existen errores en los datos "
Sección: Buscar simulación	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz que le permite buscar las simulaciones.
2. El Trabajador del proyecto provee los parámetros de búsqueda: - Sistema biológico. - Modelo Matemático. - Tipo Simulación	3. Busca la simulación en la base de datos.
	4. Muestra el resultado de la búsqueda.
5. El Trabajador del proyecto selecciona el resultado deseado.	6. Muestra el resultado seleccionado.
	7. Ir a la sección 1.
Curso Alterno: Buscar simulación	
Acción del actor	Respuesta del Sistema
	5. Emite un mensaje de: " No existe simulación con los datos introducidos "

Sección: Mostrar un listado de las simulaciones	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz que permite ver el listado de las simulaciones realizadas.
2. El Trabajador del proyecto selecciona la simulación que desea buscar.	3. Muestra las simulaciones realizadas.
Curso Alterno: Mostrar un listado de las simulaciones	
Acción del actor	Respuesta del Sistema
	3. Emite un mensaje de: " No existen simulaciones registradas con esos parámetros "
Sección: Eliminar simulación	
Acción del actor	Respuesta del Sistema
	1. Muestra una interfaz para eliminar simulación.
2. El Líder del proyecto provee los parámetros siguientes: - Sistema biológico. - Modelo Matemático. - Tipo Simulación	3. Busca la simulación en la base de datos.
	4. Elimina la simulación indicada.
	5. Ir a la acción 1.
Prioridad	Critico
Prototipo de interfaz	Ver Anexo 7

Tabla 2.6: Descripción del Caso de Uso Gestionar Simulación

Conclusiones del Capítulo

En este capítulo se ha descrito la solución propuesta utilizando el Modelo del Dominio, en el que se definieron los conceptos principales, se realizó el diagrama de casos de uso del sistema, permitiendo mostrar la relación entre éstos y los actores del sistema, además se describieron los casos de usos críticos del sistema, facilitando el diseño de las clases y la elaboración de los diagramas correspondientes, estas actividades permitirán diseñar las clases con vista a la implementación. Sólo se describen 5 de los casos de uso críticos, el resto están documentados en el expediente de proyecto.

CAPITULO 3: Análisis y diseño del sistema.

En este capítulo se describen, a través del modelo de análisis, los requisitos funcionales de forma tal que facilite una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a la estructura general del sistema. Se desarrolla el modelo de diseño creando una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos, interfaces y clases.

3.1 Descripción de la arquitectura (Vista de casos de usos).

Este artefacto es realizado por el Arquitecto, representa los casos de usos significativos para la arquitectura, debido a que describen alguna funcionalidad importante y crítica que debe priorizarse dentro del ciclo de vida del software.

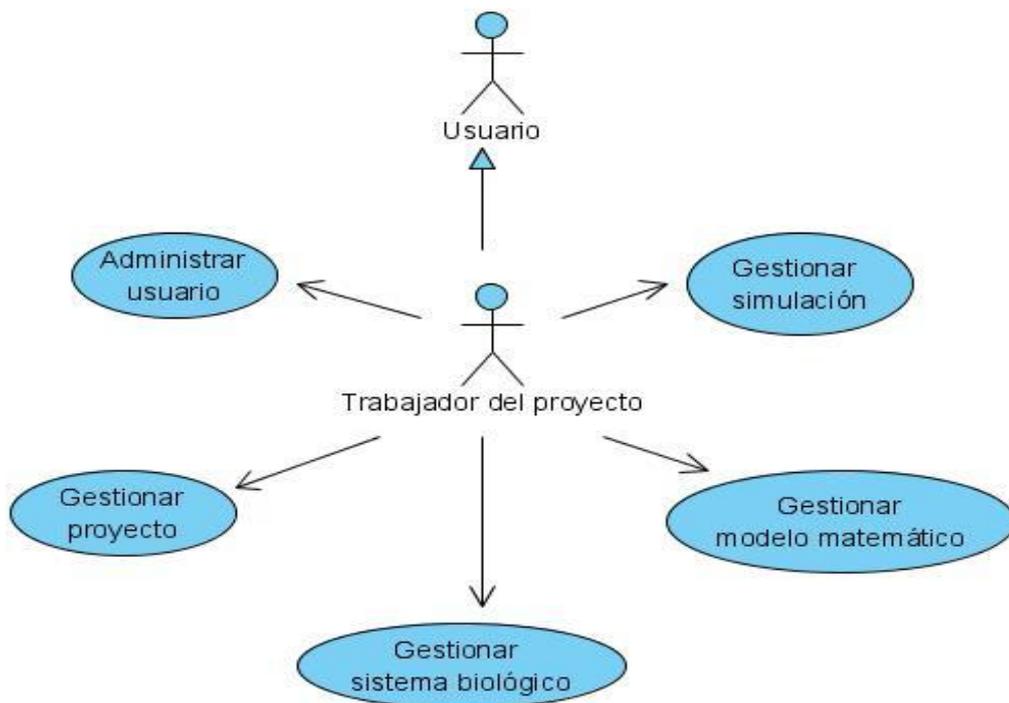


Figura 3.1: Vista de casos de usos.

3.2 Modelo de Análisis

El Modelo de Análisis ayuda a conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos. Crea una entrada apropiada y un punto de partida para la implementación. Además ayuda a descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. Se puede considerar como una primera aproximación al modelo de diseño. [8]

3.2.1 Diagramas de clases del análisis.

Un diagrama de clases del análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Describe gráficamente las especificaciones de las clases de software, de las interfaces, así como sus relaciones en una aplicación. [9]

A continuación se muestran los diagramas de clases del análisis de los casos de usos críticos del sistema.

Diagrama de Clases del Caso de Uso Administrar usuario.

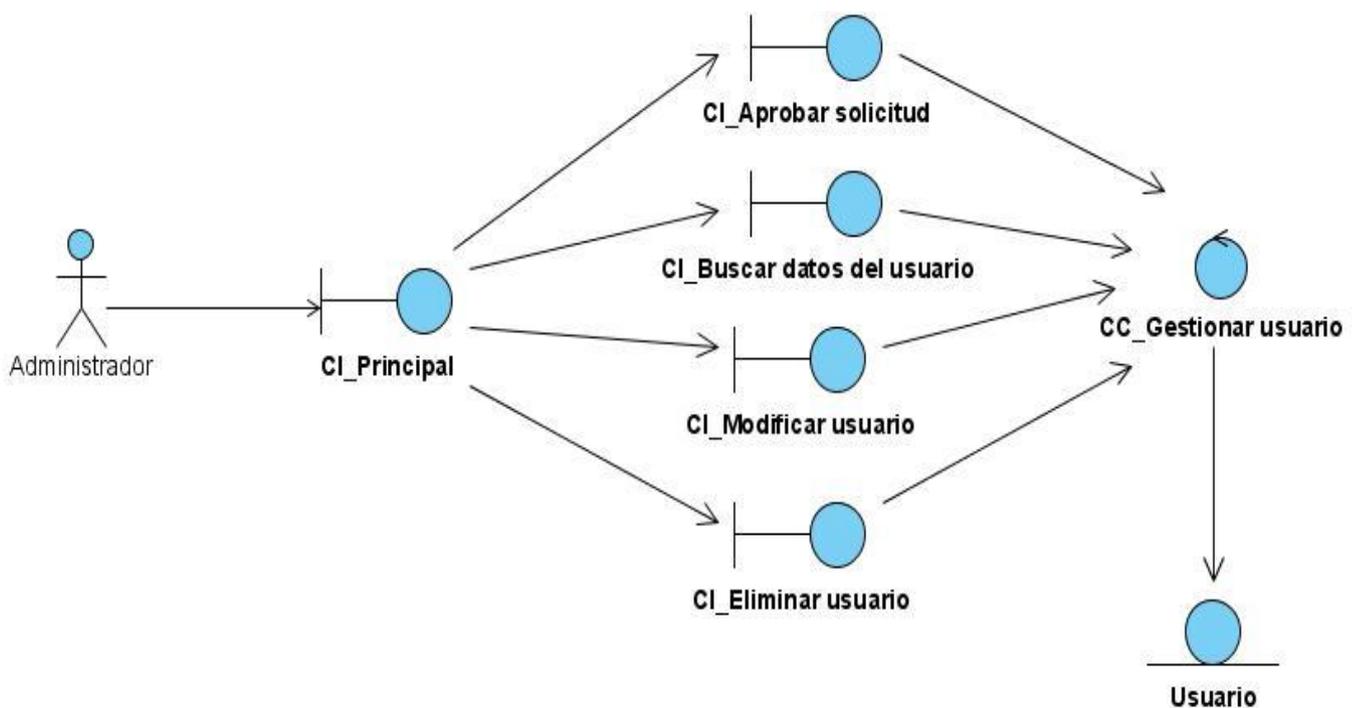


Figura 3.1: Diagrama de Clases del Caso de Uso Administrar usuario.

Diagrama de Clases del Caso de Uso Gestionar proyecto.

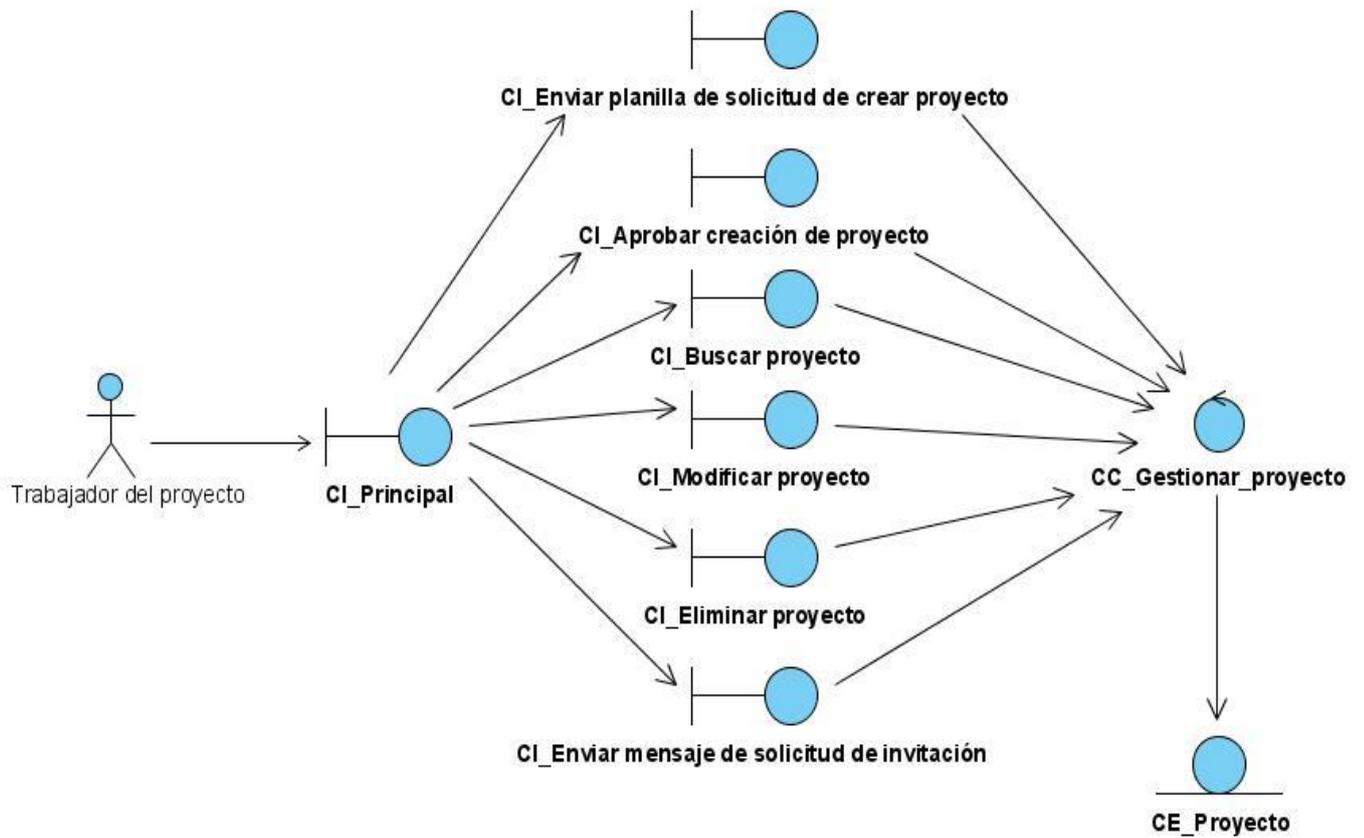


Figura 3.2: Diagrama de Clases del Caso de Uso Gestionar proyecto.

Diagrama de Clases del Caso de Uso Gestionar sistema biológico.

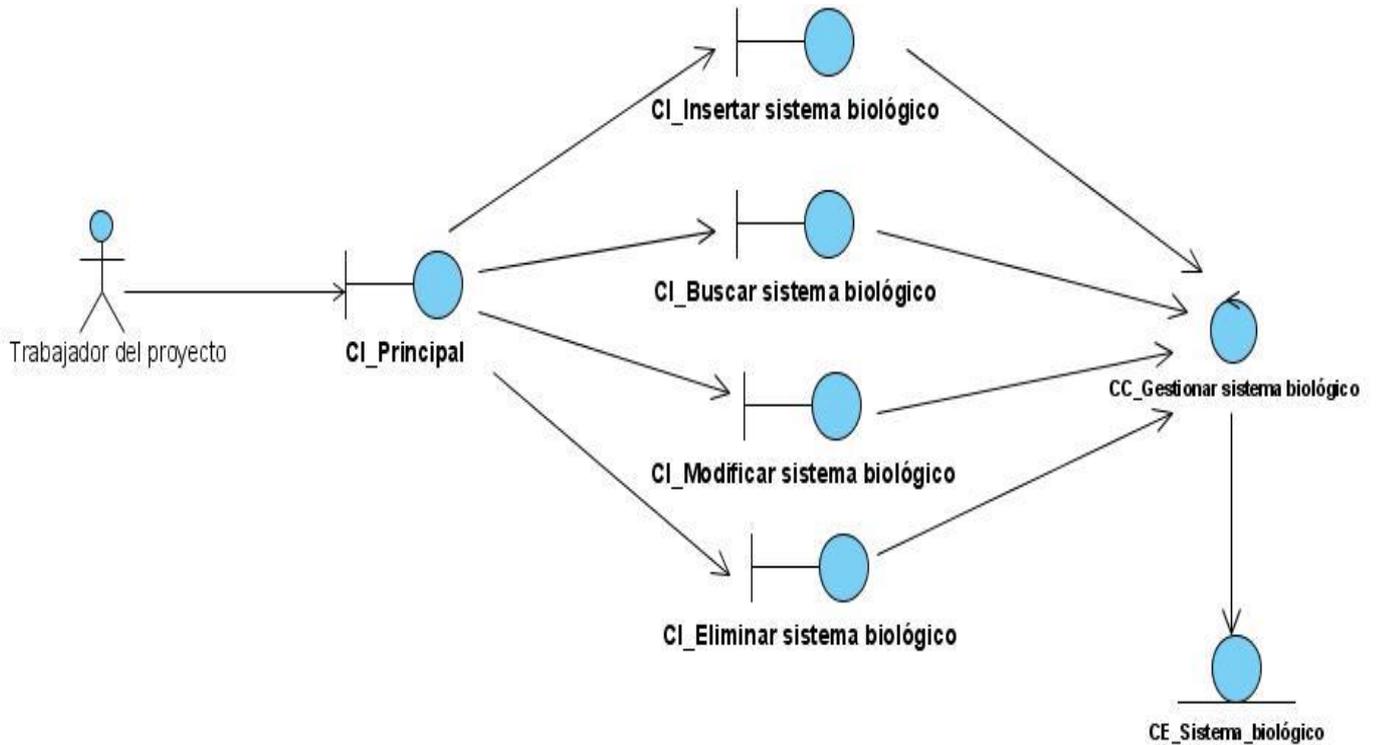


Figura 3.3: Diagrama de Clases del Caso de Uso Gestionar sistema biológico.

Diagrama de Clases del Caso de Uso Gestionar modelo matemático.

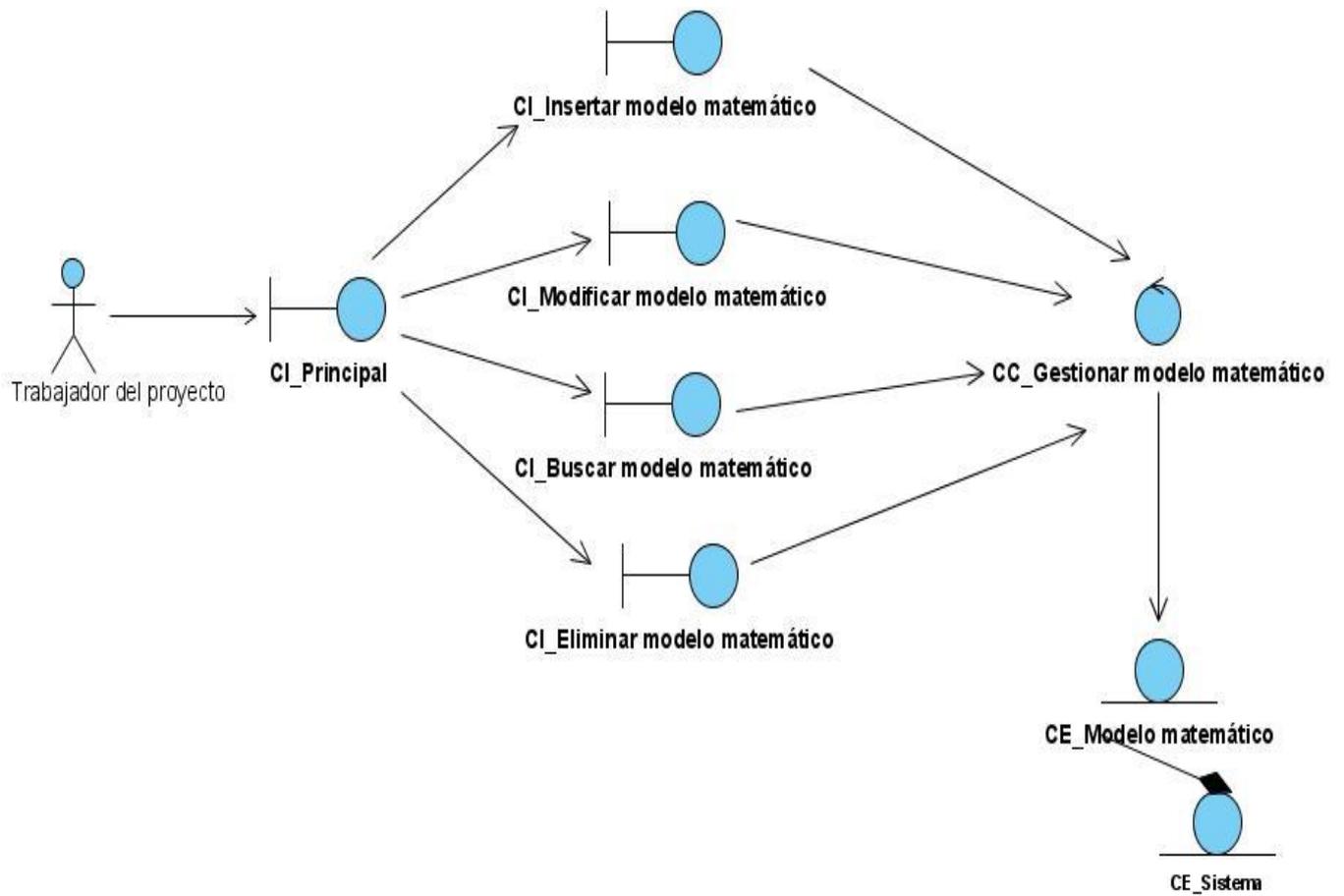


Figura 3.4: Diagrama de Clases del Caso de Uso Gestionar modelo matemático.

Diagrama de Clases del Caso de Uso Gestionar simulación.

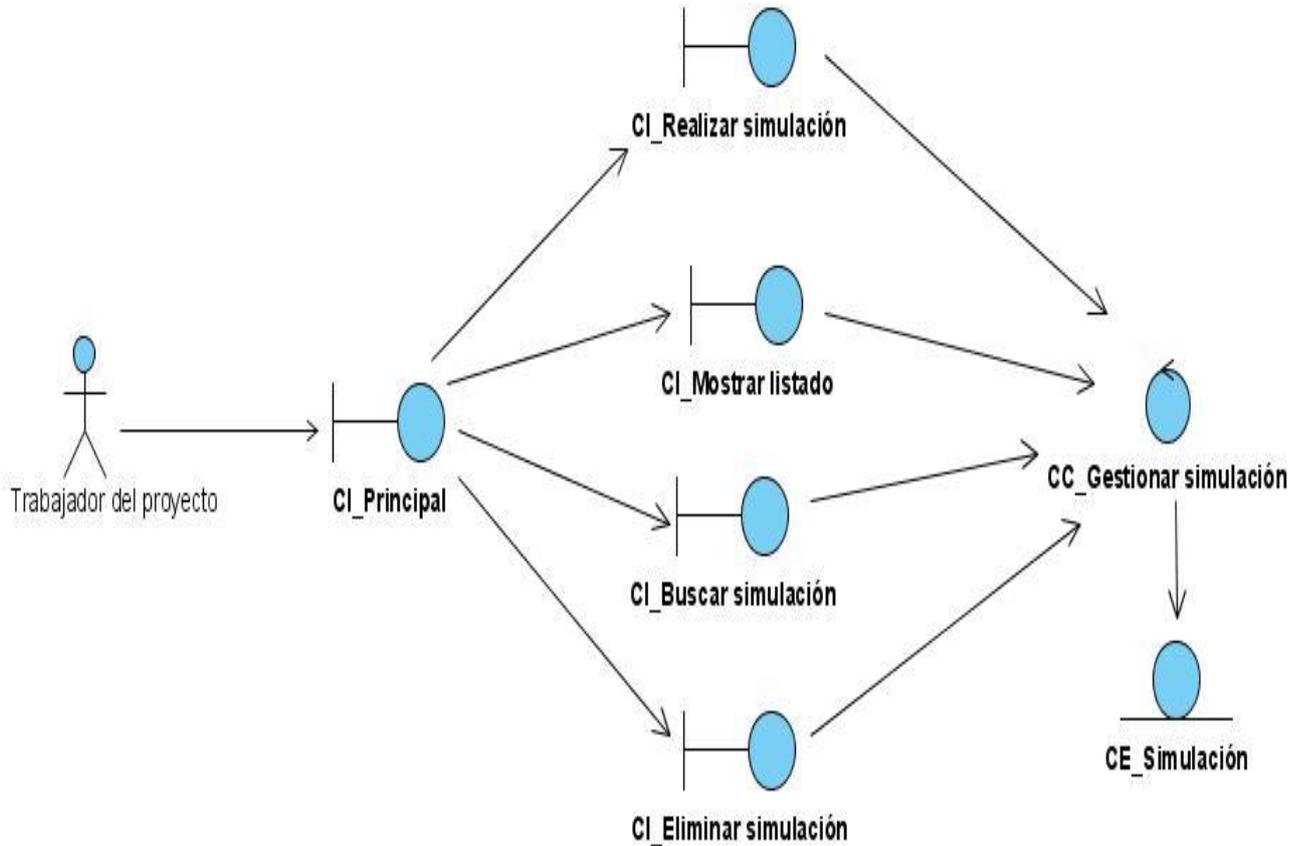


Figura 3.5: Diagrama de Clases del Caso de Uso Gestionar simulación.

3.3 Modelo de Diseño.

El Modelo de Diseño es un modelo de objeto que adquiere una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución concurrencias y, tecnologías de interfaz de usuario. Además ayuda a descomponer los trabajos de implementación en partes mas manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia. [10]

3.3.1 Patrones

¿Que es un patrón de diseño?

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular.

¿Que son los patrones GRASP?

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Patrones Básicos de GRASP

- Experto:

Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

- Creador:

Asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.

- Alta Cohesión:

Asignar una responsabilidad de modo que la cohesión siga siendo alta.

Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas.

Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo por hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande.

El patrón Alta Cohesión presenta semejanzas con el mundo real. Si alguien asume demasiadas responsabilidades, no será eficiente.

- Bajo Acoplamiento:

Asignar una responsabilidad para mantener bajo acoplamiento.

Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

- Controlador:

Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

En la investigación se aplican todos los patrones básicos de GRASP en los diagramas de clases del diseño.

3.3.2 Principios del diseño

Extensiones UML para Web.

UML posee una extensión para el modelado de aplicaciones Web, usada para el diseño de las clases, dicha extensión utiliza diferentes estereotipos que permiten definir un nuevo significado de la semántica para el elemento a modelar, los estereotipos más usados son:



<<Server Page>> Representa la página Web que tiene código que se ejecuta en el servidor.

Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.

Esta clase sólo puede tener relaciones con objetos en el servidor, La página servidor tiene una relación 1:1 con un fichero en el servidor. En las aplicaciones en PHP se corresponde con un fichero .php



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML.

Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del *script* que son accesibles para páginas cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.



<<Form>> Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields).

<<Build>>



<<Build>> Representa una asociación especial que relaciona las páginas cliente con las páginas servidor, de forma general se expresa como que las páginas que se encuentran en el servidor construyen las páginas en el cliente. Debe ser una relación direccional, donde una página servidor puede construir una o más páginas cliente.

<<Submit>>



<<Submit>> Es la relación que se crea siempre entre una página servidor y un formulario, a través de esta relación el formulario manda los valores de sus campos al servidor, para ser procesados por la página servidor.

3.3.3 Clases de la Capa de Acceso a Datos:

PHP 5 incluye el uso de la capa de abstracción de acceso a datos PDO (PHP Data Objects), que sirvió como base para el diseño de una clase (Capa_acceso_datos) con la funcionalidad necesaria para el manejo de las clases entidad. Creando una instancia de esta clase se puede acceder a la Base de Datos, utilizando todos los métodos y procedimientos de la misma.

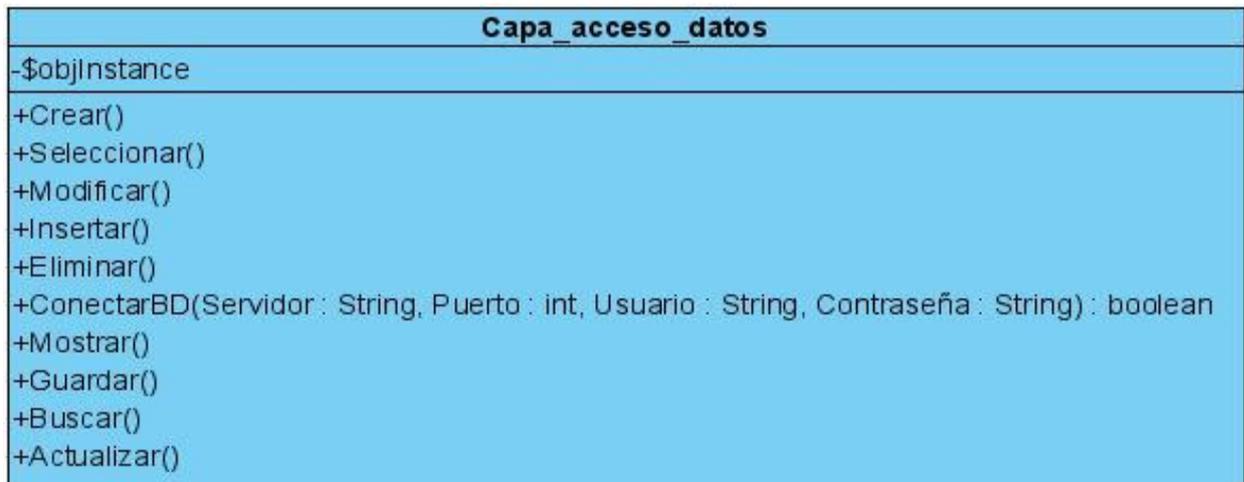


Figura 3.6: Capa de Acceso a Datos.

3.3.4 Clases de la Capa de Presentación:

La capa de presentación incluye las clases clientes con los formularios asociados, las cuales además incluyen un código JavaScript para la validación de los datos que se entran así como un método para la impresión de lo mostrado por las mismas.



Figura 3.7: Capa de Presentación.

3.3.5 Arquitectura Web de tres niveles.

La arquitectura de las aplicaciones Web suele presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).



Figura 3.8: Arquitectura Web de tres niveles.

3.3.6 Diagramas de clases de diseño.

Un Diagrama de Clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Gráficamente, un diagrama de clases es una colección de nodos y arcos. [11]

A continuación se muestran los diagramas de clase el diseño de los casos de usos críticos del sistema.

Diagrama de Clase de Diseño del Caso de Uso Administrar usuario

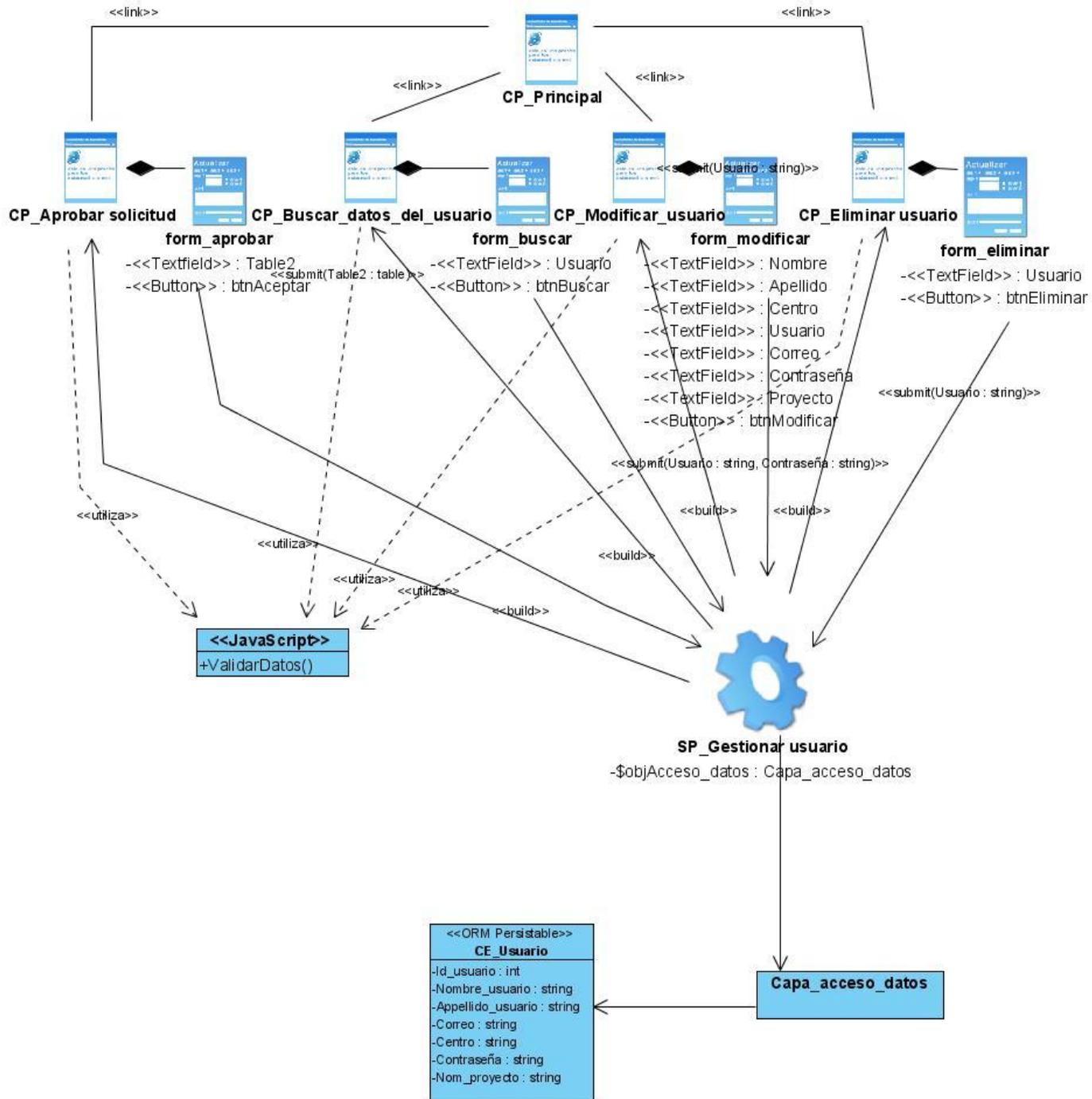


Figura 3.9: Diagrama de Clase de Diseño del Caso de Uso Administrar usuario

Diagrama de Clase de Diseño del Caso de Uso Gestionar proyecto

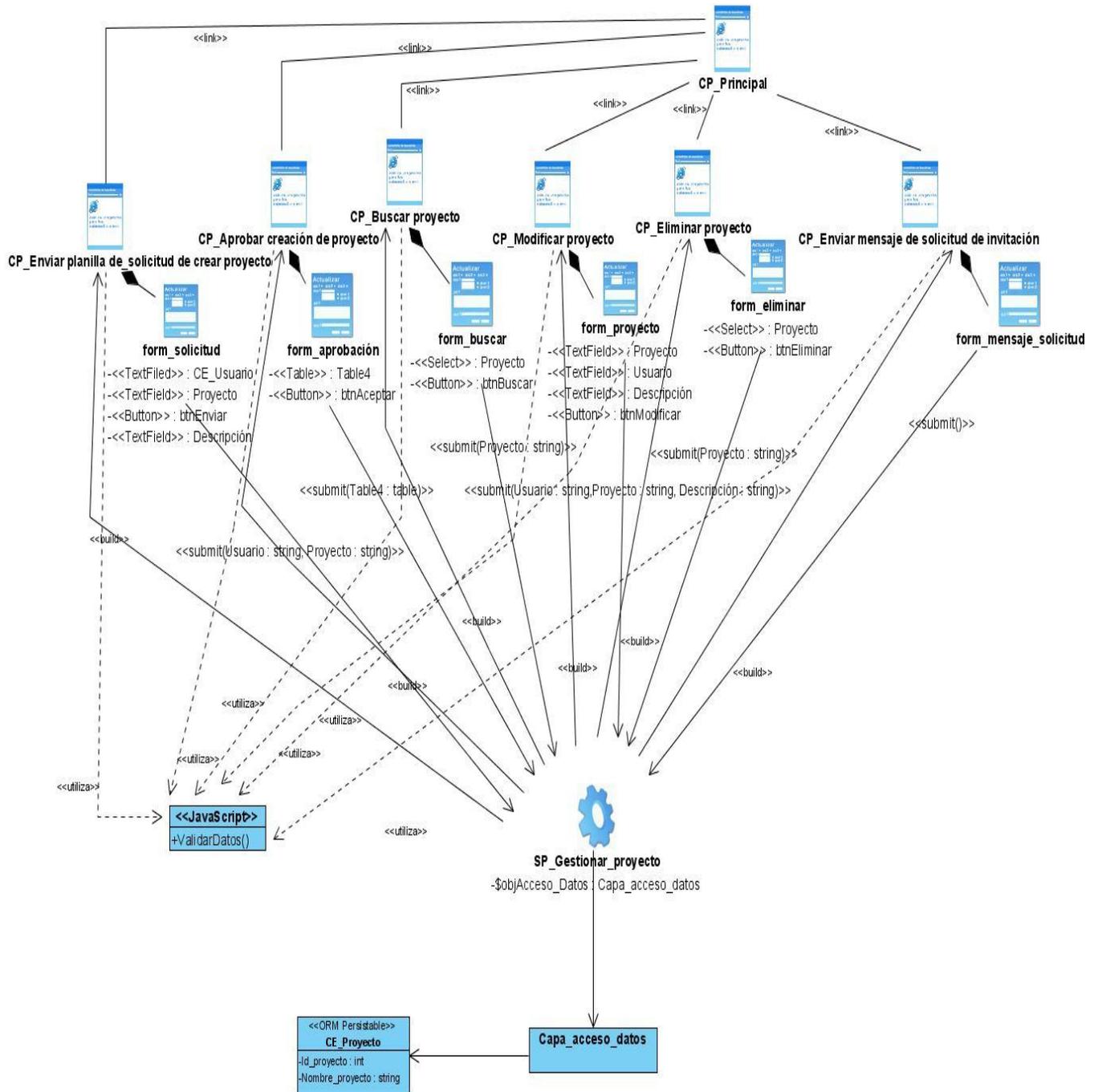


Figura 3.10: Diagrama de Clase de Diseño del Caso de Uso Gestionar proyecto

Diagrama de Clase de Diseño del Caso de Uso Gestionar sistema biológico

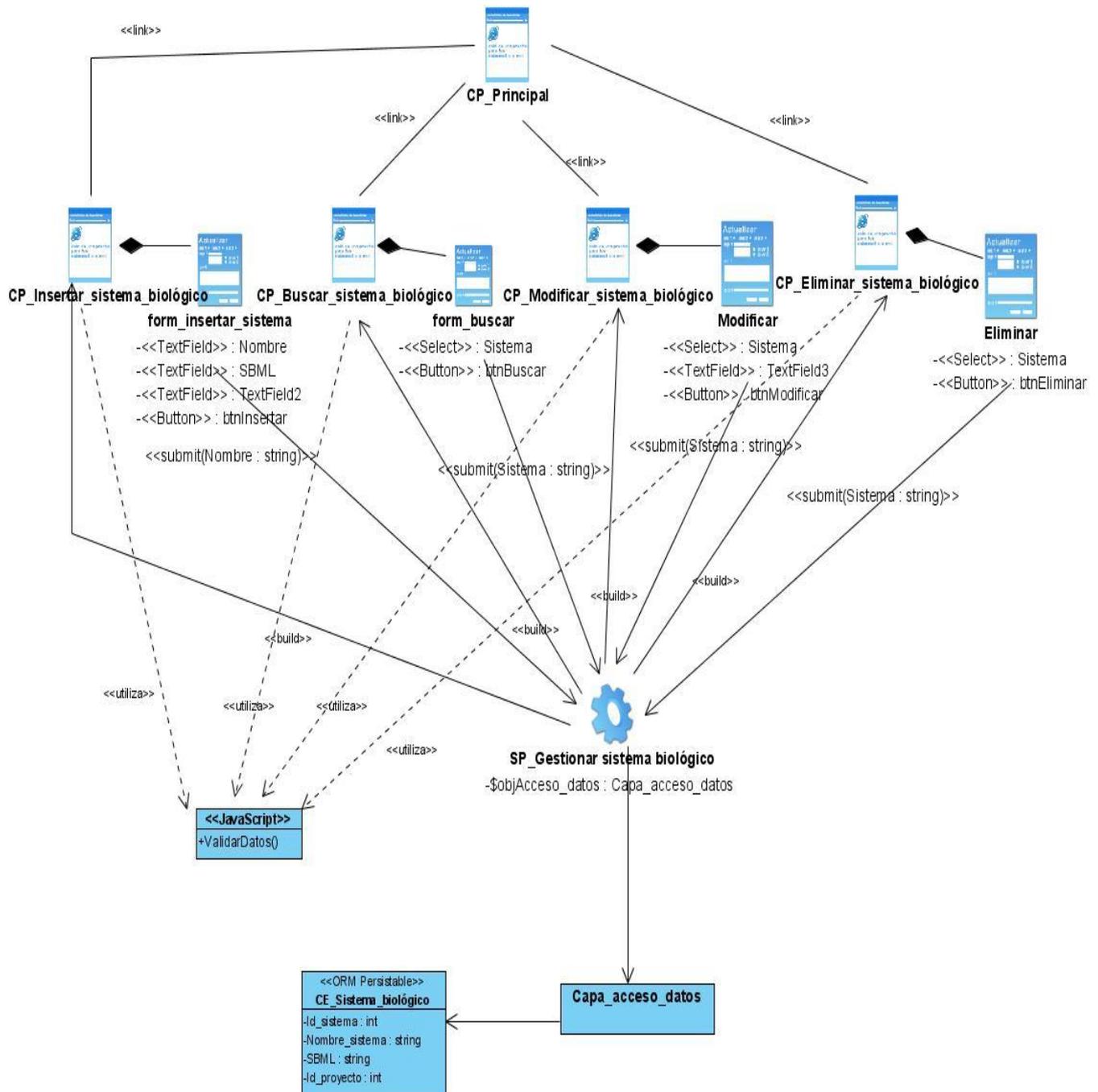


Figura 3.11: Diagrama de Clase de Diseño del Caso de Uso Gestionar sistema biológico

Diagrama de Clase de Diseño del Caso de Uso Gestionar modelo matemático

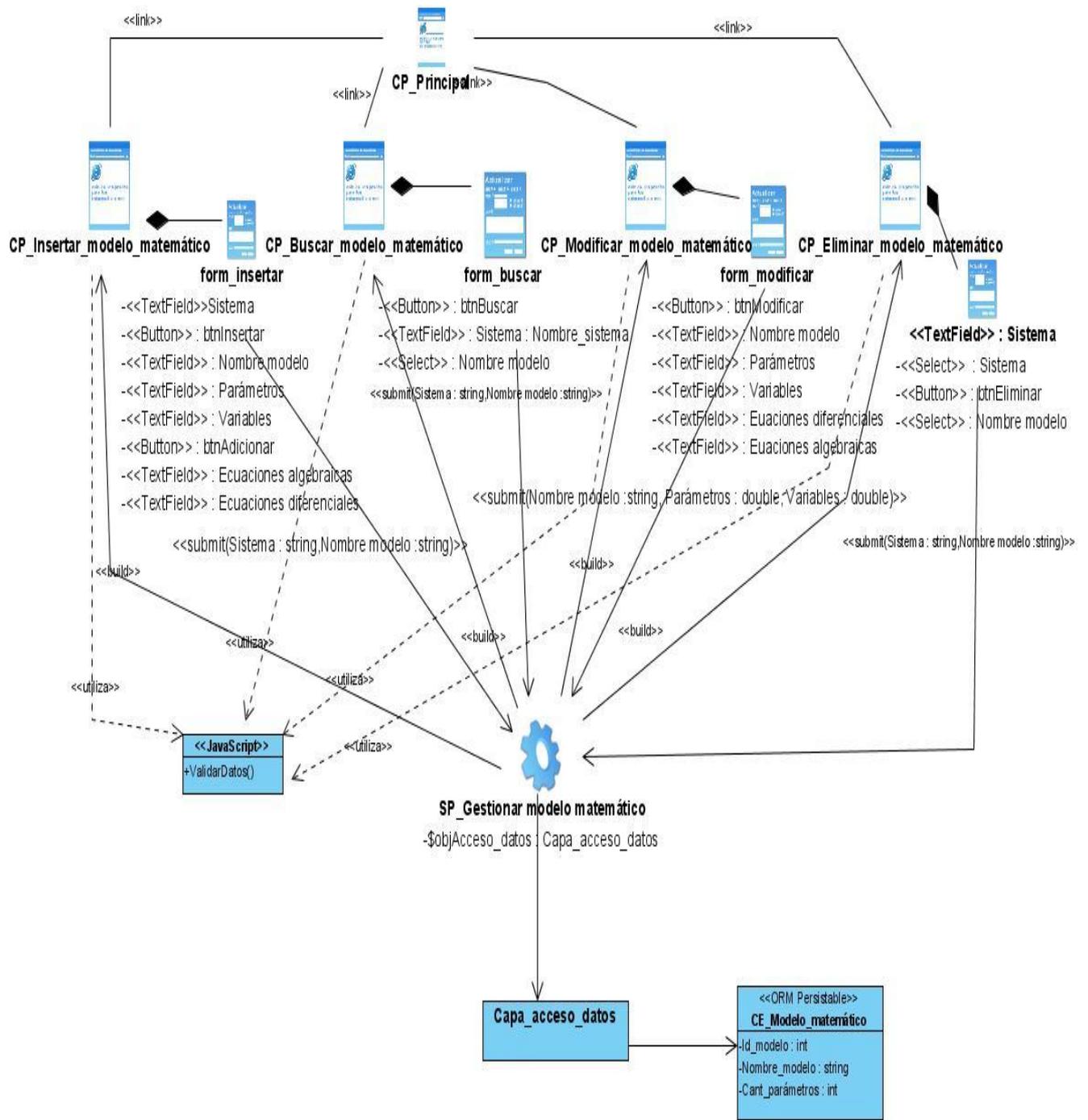


Figura 3.12: Diagrama de Clase de Diseño del Caso de Uso Gestionar modelo matemático

Diagrama de Clase de Diseño del Caso de Uso Gestionar simulación

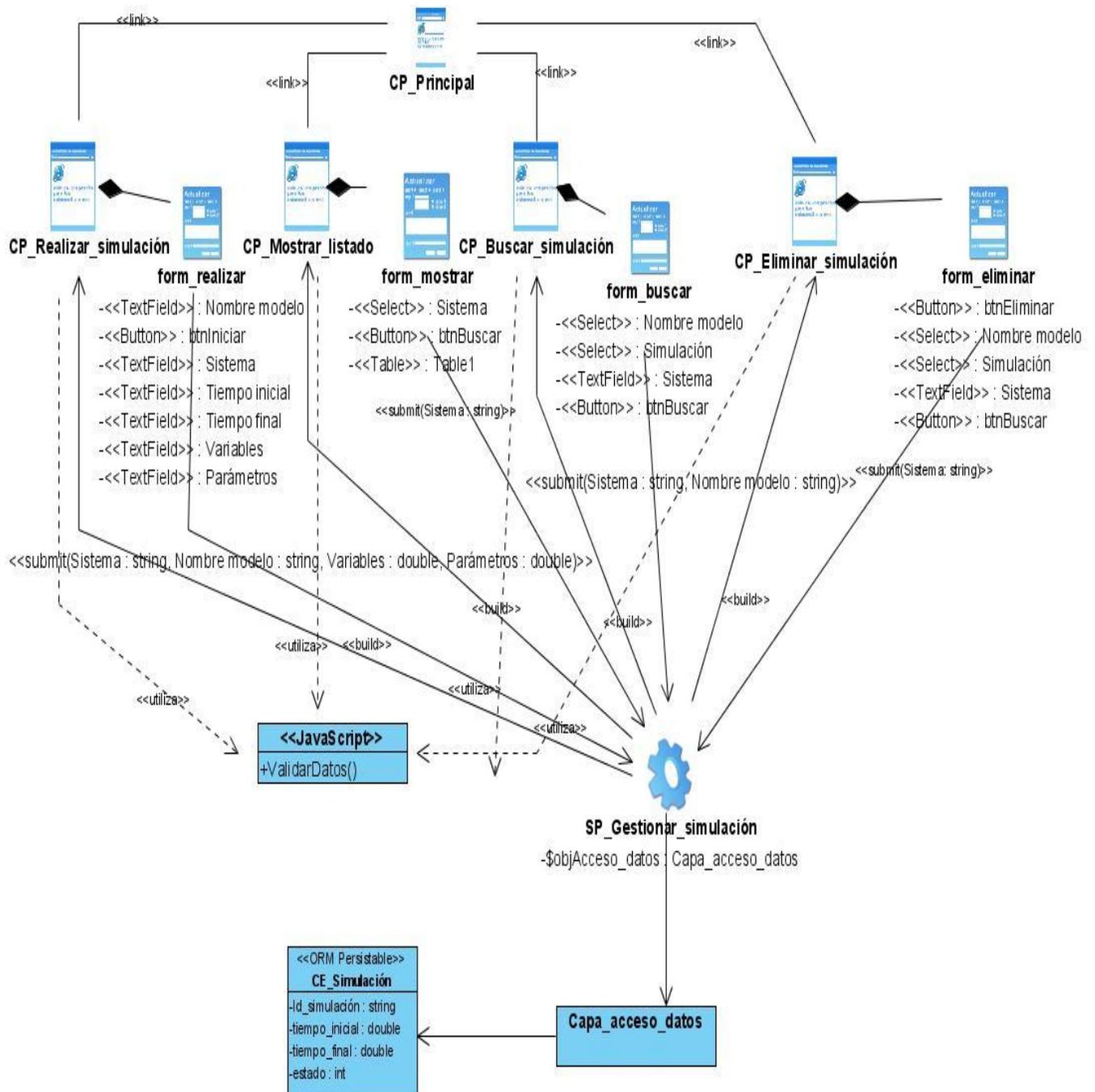


Figura 3.13: Diagrama de Clase de Diseño del Caso de Uso Gestionar simulación.

3.4 Base de Datos.

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. La estructura tiene que ser desarrollada para la necesidad de las aplicaciones que la usarán.

La puesta en práctica de la base de datos es el paso final en el desarrollo de aplicaciones de soporte del negocio. Se conforman con los requisitos del proceso del negocio, que son la primera abstracción de la vista de la base de datos.

Una clase persistente es una clase entidad que tiene la capacidad de mantener su valor en el espacio y en el tiempo.

En el diagrama de clases persistentes se muestran dichas clases y las relaciones entre ellas (asociación, agregación/composición).

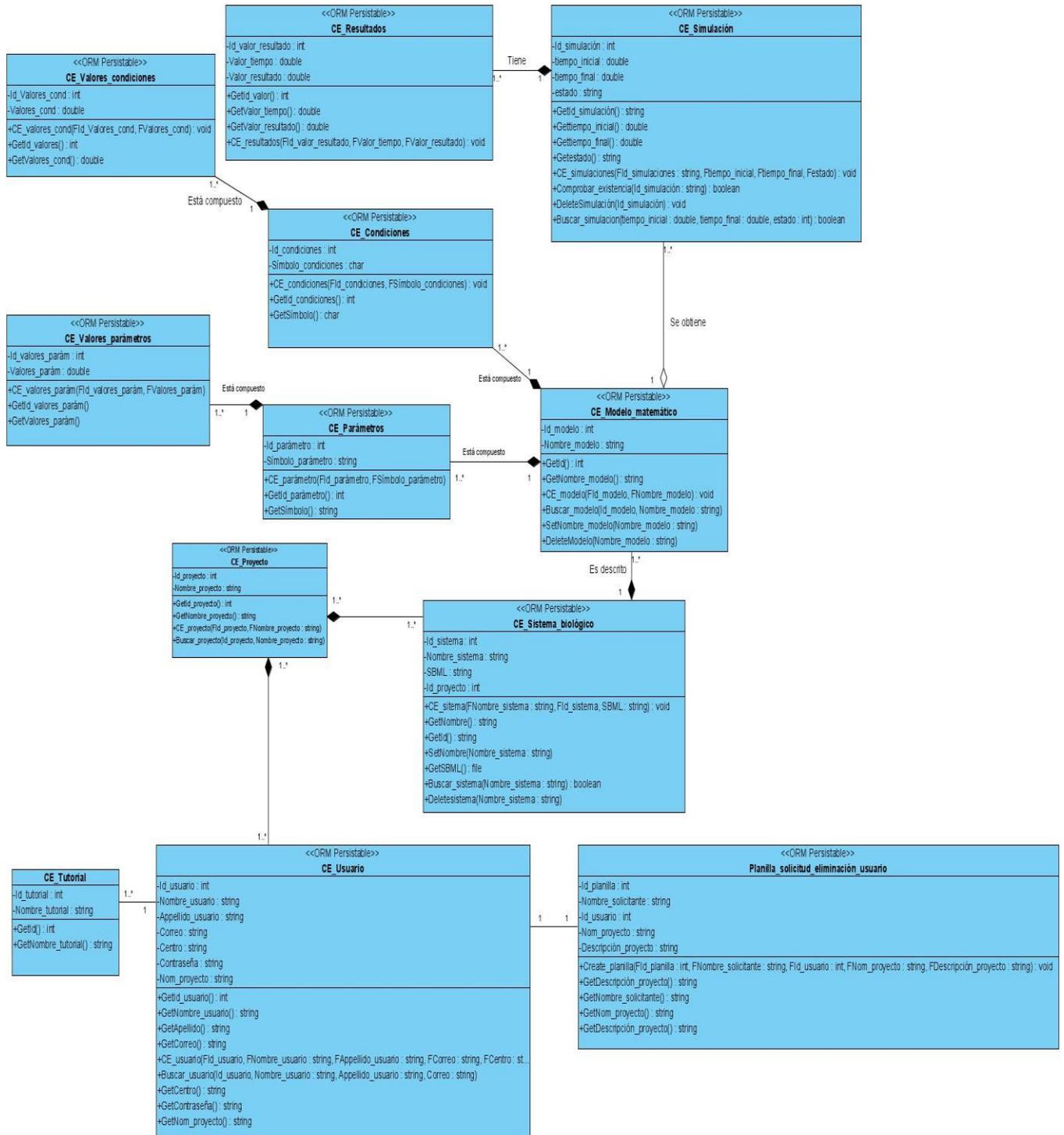


Figura 3.14: Diagrama de Clases Persistentes

Modelo físico de datos

En el diagrama del Modelo físico de datos, se muestran las tablas y sus relaciones.

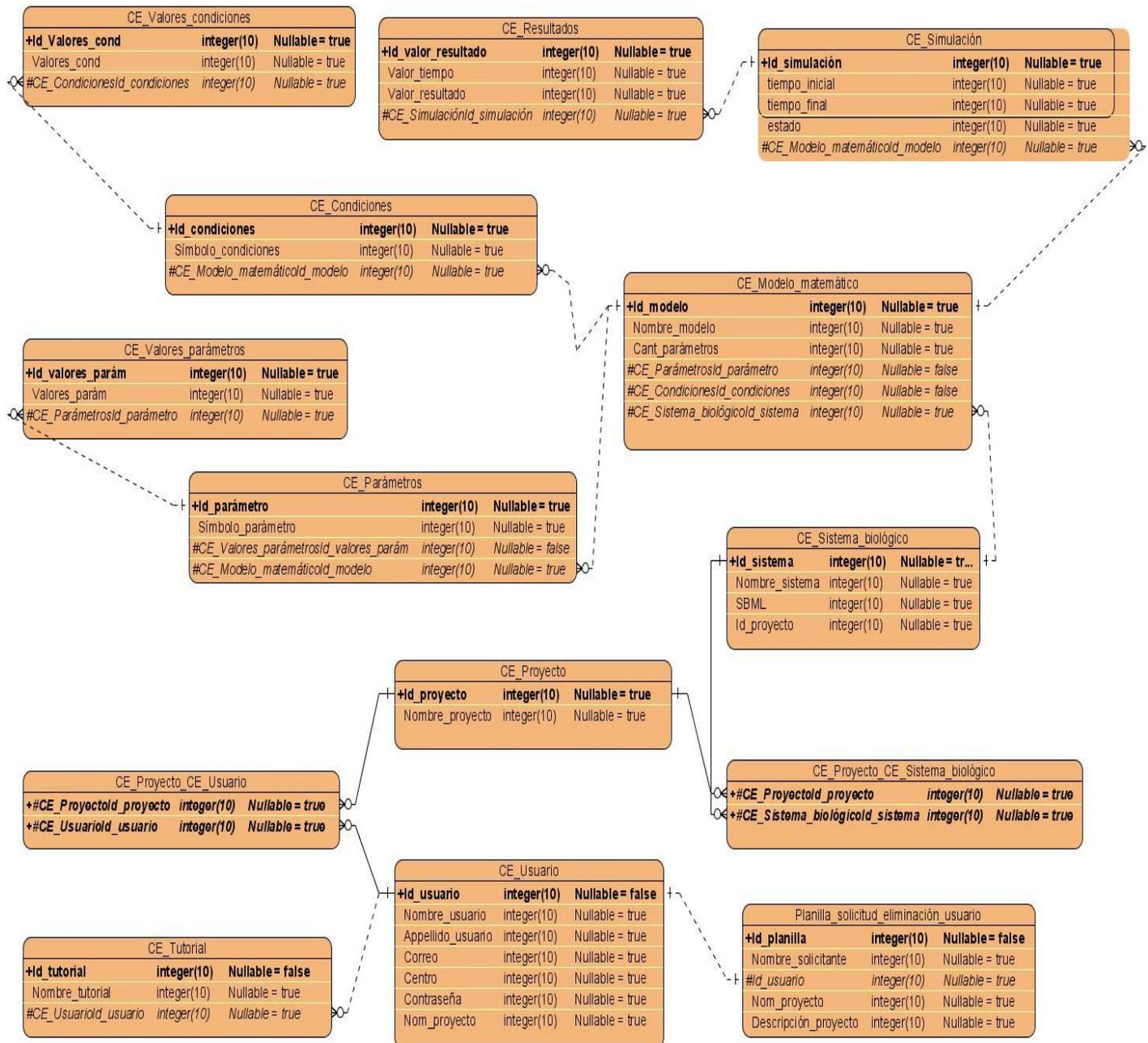


Figura 3.15: Modelo físico de datos.

Concepción de la ayuda.

El sistema consta de una ayuda que permite a los usuarios conocer aspectos de su funcionamiento, el vínculo para acceder a la ayuda se encuentra en la plantilla del sistema por lo que se podrá consultar desde cualquier interfaz, en ella se explica de forma clara y amena todas las funcionalidades del sistema por niveles de permiso.

Conclusiones del Capítulo

Como resultados de este capítulo, se representan los Diagramas de Clases del Análisis y los diagramas de Clases del Diseño de los 5 casos de usos críticos. Además se presentaron los principios del diseño seguidos en el sistema propuesto, además del diagrama de base de datos y el diagrama físico.

CONCLUSIONES

- Se ha realizado el análisis y diseño de una aplicación Web que mejorará la eficacia en el estudio de los Sistemas Biológicos a través de la integración de un grupo de algoritmos utilizados en la simulación de los mismos.
- Se elaboró un modelo conceptual para contribuir a la comprensión del contexto del sistema. Además se determinaron y elaboraron los requisitos funcionales y no funcionales para desarrollar las características del sistema y determinar las responsabilidades que se propone el mismo; se describieron los casos de uso del sistema para una mejor comprensión de cada uno, se diseñaron los prototipos de interfaz para tener una visión de cómo iría quedando el sistema en un principio y se llevó a cabo el diseño de las clases a implementar con el objetivo de ayudar a la programación del sistema.
- Esta aplicación Web permitirá gestionar la información de las simulaciones realizadas sobre los Sistemas Biológicos, facilitándole a las instituciones que no la posean, el potencial de cálculo necesario en los análisis que así lo requieran y así obtener resultados más eficaces en sus investigaciones.

RECOMENDACIONES

- Implementar la aplicación Web.
- Agregar funcionalidades a la aplicación Web.

REFERENCIAS BIBLIOGRÁFICAS

1. [Online] 1 6, 2007.
http://www.laverdad.es/murcia/pg060602/prensa/noticias/Region_Murcia/200606/02/MUR-REG-011.html.
2. [Online] 1 11, 2007. <http://www.imbiomed.com/Innsz/Nnv46n4/espanol/Wnn44-07.html>.
3. [Online] 2 3, 2007. <http://php.ciberaula.com/articulo/PHPoASP/>.
4. [Online] 2 10, 2007. <http://www.addlink.es/productos.asp?pid=1>.
5. [Online] 2 12, 2007. <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art51.asp>.
6. [Online] 3 4, 2007.
<http://www.grupoassa.com/Espanol/general.asp?N1=3&N2=4&N3=1&N4=0&cat=mi&reIID=284>.
7. Ivar Jacobson, Grady Booch, James Rumbaugh. El proceso unificado de desarrollo. s.l. : Félix Varela, 2004. (4/04/2007)
8. Ivar Jacobson, Grady Booch, James Rumbaugh. *El proceso unificado de desarrollo*. s.l. : Félix Varela, 2004. (6/04/2007)
9. Ivar Jacobson, Grady Booch, James Rumbaugh. *El proceso unificado de desarrollo*. s.l. : Félix Varela, 2004. (10/04/2007)
10. Ivar Jacobson, Grady Booch, James Rumbaugh. *El proceso unificado de desarrollo*. s.l. : Félix Varela, 2004. (11/04/2007)
11. Ivar Jacobson, Grady Booch, James Rumbaugh. *El proceso unificado de desarrollo*. s.l. : Félix Varela, 2004. (2/05/2007)

BIBLIOGRAFÍA

1. [Online] 2 10, 2007. <http://www.addlink.es/productos.asp?pid=1>.
2. [Online] 1 11, 2007.
http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures.html
3. [Online] 3 11, 2007. <http://www.cellware.org>
4. [Online] 18 3, 2007. <http://www.csi.map.es/csi/silice/Global71.html>
5. [Online] 20 3, 2007. <http://es.tldp.org/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.html>
6. [Online] 19 3, 2007. <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>
7. [Online] 3 4, 2007.
<http://www.grupoassa.com/Espanol/general.asp?N1=3&N2=4&N3=1&N4=0&cat=mi&reIID=284>.
8. [Online] 1 11, 2007. <http://www.imbiomed.com/Innsz/Nnv46n4/espanol/Wnn44-07.html>.
9. [Online] 1 11, 2007. <http://www.immunogrid.org>.
10. Ivar Jacobson, Grady Booch, James Rumbaugh. *El proceso unificado de desarrollo*. s.l. : Félix Varela, 2004. (4/07/2007)
11. [Online] 1 6, 2007.
http://www.laverdad.es/murcia/pg060602/prensa/noticias/Region_Murcia/200606/02/MUR-REG-011.html.
12. [Online] 2 12, 2007.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art51.asp>.
13. [Online] 8 1, 2007. <http://www.maestrosdelWeb.com/editorial/zendstudio/>
14. Övergaard, Gunnar y Palmkvist, Karin: *Use Cases Patterns and Blueprints*, Addison Wesley Professional, 2004. (6/07/2007)
15. [Online] 2 3, 2007. <http://php.ciberaula.com/articulo/PHPoASP/>.
16. REF Larman, Craig: *UML y patrones, introducción al análisis y diseño orientado a objetos*, Félix Varela, 2004. (6/07/2007)
17. [Online] 1 11, 2007. <http://www.vcell.org>

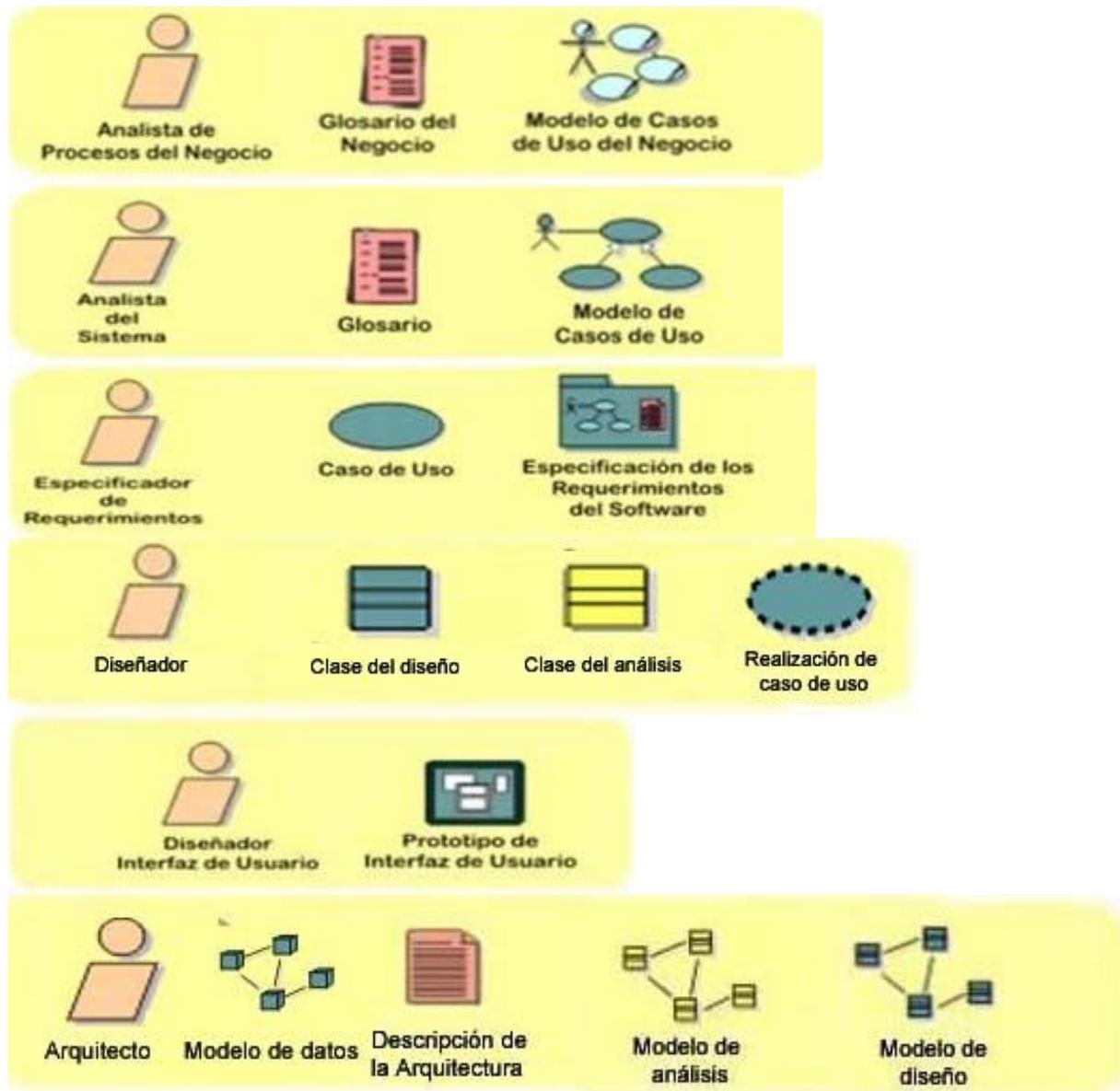
18. [Online] 3 11, 2007. <http://www.virtualcell.org>

19. [Online] 15 3, 2007. <http://www.visual-paradigm.com/product/vpuml/>

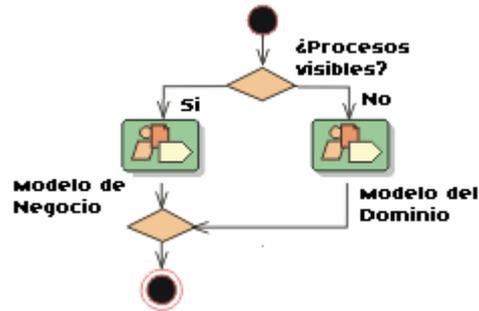
20. [Online] 1 11, 2007. <http://www.webcell.org>

ANEXOS

Anexo 1



Anexo 2



Anexo 3 Prototipos de interfaz no funcionales del Caso de Uso Administrar usuario.

Sección: Permitir aprobar la solicitud de registro de los datos del usuario.

Usuario	Centro de Trabajo	Proyecto	Datos
xxxxxxx	xxxxxxx	xxxxxxx	ver

Sección: Buscar los datos del usuario.

Usuario:

Sección: Modificar los datos del usuario.

Nombre:

Apellidos:

Centro de trabajo:

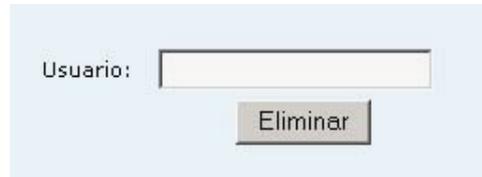
Correo electrónico:

Usuario:

Contraseña:

Proyecto:

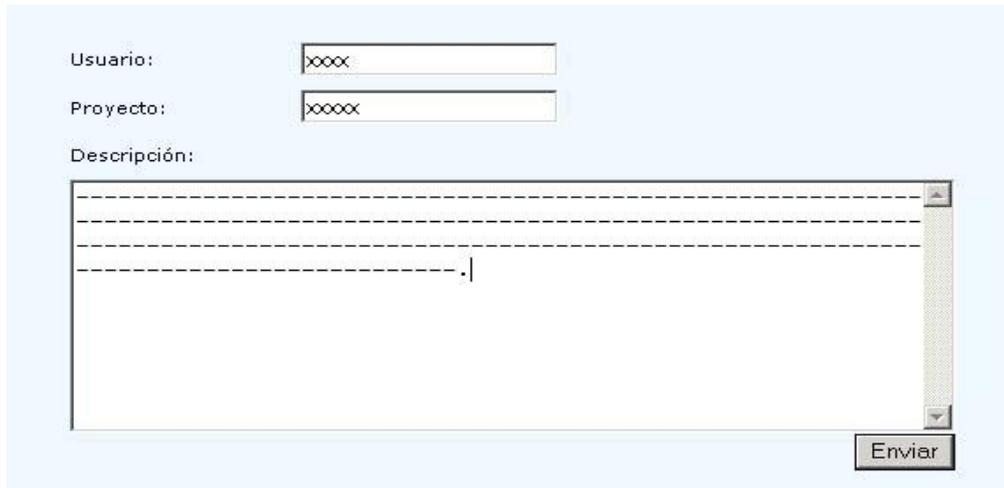
Sección: Eliminar usuario.



A light blue rectangular form containing a text input field labeled 'Usuario:' and a button labeled 'Eliminar' positioned below the input field.

Anexo 4 Prototipos de interfaz no funcionales del Caso de Uso Gestionar proyecto.

Sección: Permitir enviar planilla de solicitud para la creación de un proyecto.



A light blue rectangular form containing three input fields: 'Usuario:' with 'xxxx', 'Proyecto:' with 'xxxxx', and 'Descripción:' with a large text area containing a dashed line and a cursor. A button labeled 'Enviar' is located at the bottom right of the form.

Sección: Permitir aprobar la creación de un proyecto.

Usuario	Proyecto	Planilla
xxxxxxx	xxxxxxx	ver

Sección: Buscar proyecto.



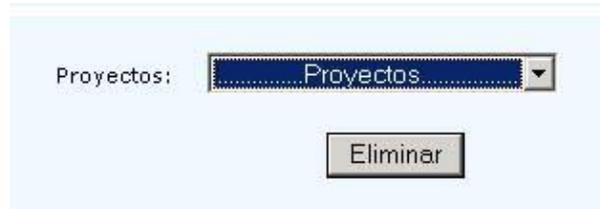
Proyectos:

Sección: Modificar proyecto



Usuario:
Proyecto:
Descripción:

Sección: Eliminar proyecto.



Proyectos:

Anexo 5 Prototipos de interfaz no funcionales del Caso de Uso Gestionar sistema biológico.

Sección: Insertar sistema biológico.



Nombre:

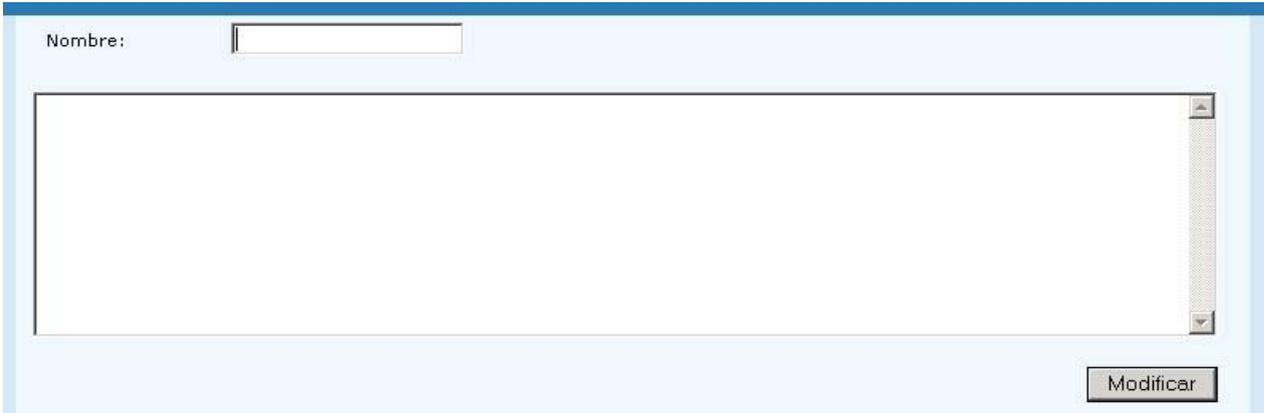
SBML:

Sección: Buscar sistema biológico.



Sistema

Sección: Modificar sistema biológico.



A screenshot of a web form with a light blue background. At the top left, there is a label "Nombre:" followed by an empty text input field. Below this is a large, empty rectangular area with a vertical scrollbar on the right side, likely for a description or notes. In the bottom right corner of the form, there is a button labeled "Modificar".

Sección: Eliminar sistema biológico.



A screenshot of a web form with a light blue background. It features a label "Sistema" to the left of a dropdown menu. The dropdown menu is open, showing a list with the word "Sistemas" selected. Below the dropdown menu is a button labeled "Eliminar".

Anexo 6 Prototipos de interfaz no funcionales del Caso de Uso Gestionar modelo matemático.

Sección: Insertar modelo matemático.

Sistema: Nombre Modelo:

Parametros

Variables

Formato Matematico:

Ecuaciones algebraicas:

Ecuaciones Diferenciales:

Sección: Buscar modelo matemático y Sección: Eliminar modelo matemático.

Sistema: Nombre Modelo:

Sección: Modificar modelo matemático.

ID Sistema: Nombre Modelo:

p	Parametros
	<input type="text"/>

y	Variables
	<input type="text"/>

Formato Matematico:

Ecuaciones algebraicas:

Ecuaciones Diferenciales:

Anexo 7 Prototipos de interfaz no funcionales del Caso de Uso Gestionar simulaciones.

Sección: Realizar simulación dado un modelo matemático.

Tipo de simulación fija:

Sistema: Nombre Modelo:

Variables	Valores	Parametros	Valores
	0		0
	0		0
	0		0
	0		0
	0		0

Tiempo Inicial : Tiempo Final:

Método matemático:

Tipo de simulación múltiple:

Sistema: Nombre Modelo:

Variables	V. Inicial	Varía	V. Final	Num. Ptos	Variación	Parametros	V. Inicial	Varía	V. Final	Num. Ptos	Variación
	0	<input checked="" type="checkbox"/>	0	0	Lineal		0	<input checked="" type="checkbox"/>	0	0	Lineal
	0	<input type="checkbox"/>	0	0	Lineal		0	<input type="checkbox"/>	0	0	Lineal
	0	<input type="checkbox"/>	0	0	Lineal		0	<input type="checkbox"/>	0	0	Lineal
	0	<input checked="" type="checkbox"/>	0	0	Lineal		0	<input type="checkbox"/>	0	0	Lineal
	0	<input type="checkbox"/>	0	0	Lineal		0	<input checked="" type="checkbox"/>	0	0	Lineal

Tiempo Inicial : Tiempo Final:

Método matemático:

Tipo de simulación regla:

Sistema: Nombre Modelo:

p	Parametros
1	a
2	a
3	a
4	a
5	a

Parametros	V. Inicial	Varía	V. Final	Num. Ptos	Variación
	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Lineal <input type="button" value="v"/>
	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Lineal <input type="button" value="v"/>
	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Lineal <input type="button" value="v"/>
	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Lineal <input type="button" value="v"/>
	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Lineal <input type="button" value="v"/>

Tiempo Inicial: Tiempo Final:

Editar Reglas:

Método matemático:

Error:

Sección: Buscar simulación y Sección: Eliminar simulación.

Sistema:

Modelos:

Simulación:

- Fija
- Múltiple
- Regla

Sección: Mostrar un listado de las simulaciones.

Sistema

Sistema	Modelo	Simulación	Valor de tiempo	Valor de resultado
xxx	xxx	xxx	1111	11111

GLOSARIO DE TÉRMINOS

Artefacto: Es un fragmento de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los artefactos son los resultados tangibles del proyecto que se van creando y usando hasta obtener el producto final.

Base de datos: Conjunto no redundante de información almacenada en memoria organizada independientemente de su utilización y su implementación en máquinas accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

Bioinformática: Disciplina científica y tecnológica en la que interactúan en armonía los planteamientos investigadores de la Biología Genética y Molecular con los enfoques metodológicos y tecnológicos de la Ciencia de la Computación y la Ingeniería Informática. (Coltell et al., 2002).

Caso de uso: Especificación de las secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Cookies: Realmente las cookies no son mas que cadenas de texto que son enviadas desde el servidor al cliente (navegador) y almacenadas en este.

DirectX: Aplicación que mejora el rendimiento gráfico ya sea bidimensional, o tridimensional, texturas 3d, renderizaciones, sitios Web, juegos, etc.

Especificador de casos de uso: [...] asumen las responsabilidades de las descripciones de uno o más casos de uso.

Flujo de trabajo: es una relación de actividades que producen resultados observables dado por una secuencia de actividades realizadas por los diferentes roles.

Glosario: Define términos comunes importantes que los analistas (y otros desarrolladores) utilizan al describir el sistema.

Hardware: Componentes físicos que constituyen las Computadoras y demás dispositivos periféricos.

Hyper Text Transfer Protocol (HTTP): El protocolo de transferencia de hipertexto es el protocolo usado en cada transacción del WEB (WWW).

In silico: Investigaciones donde se aplican los implementos computacionales.

Prototipo de interfaz de usuario: Nos ayudan a comprender y especificar las interacciones entre actores humanos y el sistema durante la captura de requisitos. No sólo nos ayuda a desarrollar una interfaz gráfica mejor, sino también a comprender mejor los casos de uso.

Requerimiento: Condición o capacidad que debe cumplir un sistema.

SBML: (Systems Biology Markup Language) es un lenguaje de marco en beneficio de la biología de sistemas esta basado en XML y creado para describir simulaciones en biología de sistemas.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

UML: (Lenguaje Unificado de Modelado), es un lenguaje estándar para el modelado de software. Lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software.

WWW: World Wide Web, la Web o WWW, es un sistema de hipertexto que funciona sobre Internet.