Universidad de las Ciencias Informáticas

Facultad 6

Título: “Base de Datos del Simulador de Sistemas Biológicos: BioSyS Versión 2.0”

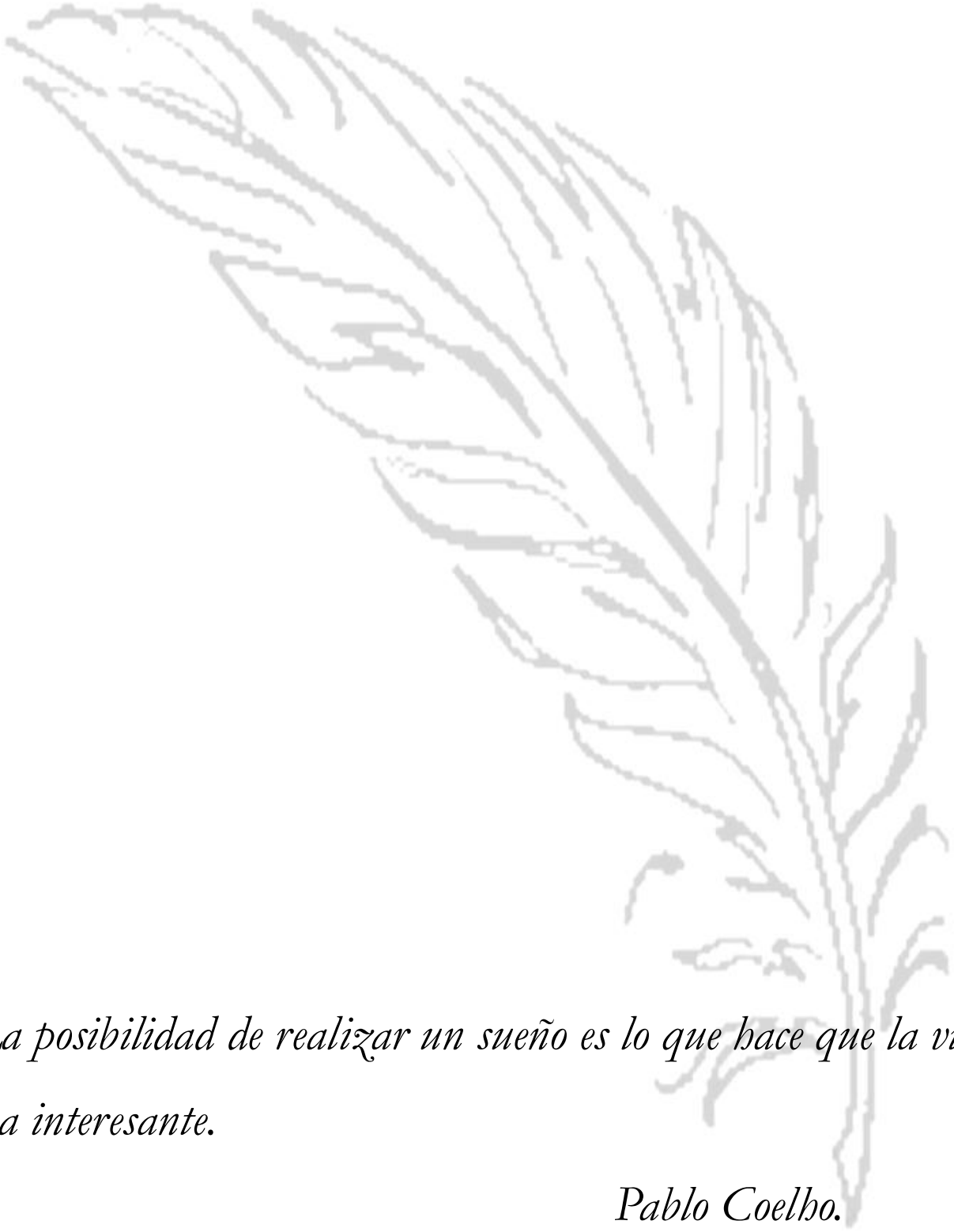
Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas.

Autores: Dayana Chaviano Brunet
Adrián Martínez Pérez

Tutores: Lic. Nara Lidia Pérez Solá
Ing. Yudelkis Abad Fuentes

Junio, 2009



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

Pablo Coelho.

DECLARACIÓN DE AUTORÍA

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

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

Dayana Chaviano Brunet

_____

Firma del Autor

Adrián Martínez Pérez

_____

Firma del Autor

Lic. Nara Lidia Pérez Solá

_____

Firma de Tutor

Ing. Yudelkis Abad Fuentes

_____

Firma de Tutor

DATOS DE CONTACTO

Tutores:

Lic. Nara Lidia Pérez Solá
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: nara@uci.cu

Ing. Yudelkis Abad Fuentes
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: yabad@uci.cu

AGRADECIMIENTOS

A la Revolución por darnos la oportunidad de cumplir nuestros sueños y permitir que nos formáramos como jóvenes del futuro.

A nuestros familiares por habernos apoyado durante toda la carrera.

A nuestras tutoras Yudelkis y Nara por su comprensión y paciencia.

A todos nuestros amigos que nos han escuchado, aconsejado y brindado su apoyo en los momentos difíciles.

A los que de manera involuntaria dejamos de mencionar, nuestro más sincero agradecimiento.

...gracias.

DEDICATORIA

De Dayana Chaviano Brunet:

A mi mamita querida por apoyarme en cada momento, por guiar mis pasos por el camino correcto, con amor, dedicación, entereza y confianza.
A mi esposo, amigo y compañero de estudios Raidel, por los momentos vividos y los que quedan por vivir, por su amor, comprensión, paciencia y apoyo incondicional.
A mis hermanos por su paciencia conmigo y porque los quiero mucho muchooo.
A mis abuelita Audelia, por ser tan buena y siempre acordarse de todos.
A mis tías Cary, Anita y Yoda y a mis tíos Barbaro y Alian, a todos los quiero mucho.
A mis suegros Dania y Luis por quererme,ayudarme y preocuparse por mi.
A mi amigo, hermano y compañero de estudios Adrián, por su amistad incondicional, paciencia y apoyo durante toda la carrera.

De Adrián Martínez Pérez:

A mis dos abuelas: Yolanda y Gloria, que son y serán siempre mis tesoros más grandes, y que a pesar de que no estén hoy aquí conmigo, viven en mí y siempre han sido fuente de inspiración, esfuerzo y amor.

A mi mamá y a mi papá, por ser los principales artífices en la consecución de mis objetivos y mi motor impulsor en la vida, por su educación, por su cariño, por su entereza y desinterés.

A mi hermano Alejandro por darme el cariño y estar ahí para que lo que necesite.

A mis abuelos por ser ejemplo de perseverancia y esfuerzo en la vida.

A mis tías Maida y Odalys, por brindarme siempre su cariño y apoyo.

A mi "segunda familia": mi barrio, por su apoyo, su confianza y aliento.

A mis tutoras que siempre cumplieron más de una función, a Nara Lidia por ser mi apoyo y guía en la realización de este trabajo y constituir esa compañera soñada para compartir tu vida y de la que no te gustaría separarte nunca; a Yudelkis por su entrega a este trabajo y por su amistad.

A Dayana Chaviano y Dayana Canova por su amistad y tolerancia.

A los amigos y compañeros de mis grupos 4 y 2, que durante estos cinco años han sabido aceptarme como soy y que aprendimos a compartir los momentos buenos y los malos en esta Universidad.

Y a todos aquellos, que por olvido involuntario, de una manera u otra, hicieron posible realizar este sueño.

RESUMEN

El Centro de Inmunología Molecular (CIM) en conjunto con la Universidad de las Ciencias Informáticas (UCI) desarrolla un software para la Simulación de Sistemas Biológicos llamado BioSyS. El mismo cuenta con una Base de Datos (BD) capaz de gestionar la información relacionada con los estudios y las simulaciones de sistemas biológicos para una primera versión del software, lo que facilita la manipulación de la avalancha de datos procedentes del análisis de los sistemas biológicos. A pesar de este actual desarrollo, aún se presentan problemas en la conservación de datos relevantes, como es el almacenamiento de la información relacionada con el proceso de análisis de las simulaciones y la información relacionada con el Editor de Ecuaciones, lo que limita la ferviente evolución en las investigaciones de esta área. En la antigua versión de BioSyS se implementaron funcionalidades básicas de análisis, en el presente trabajo se desarrolla una nueva versión del Módulo de Base de Datos, donde se le permite al investigador, clasificar las simulaciones a través de: algoritmos de Clustering, análisis por Reglas definidas por el usuario, clasificación Manual o mediante el software Weka, así como, agrupar las ecuaciones por categorías en el Editor de Ecuaciones. Este avance en el análisis de la información mejora las expectativas del cliente y brinda al investigador, una herramienta poderosa para el almacenamiento y análisis de sistemas biológicos, donde la información almacenada se modificará en tiempo de ejecución, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

PALABRAS CLAVE: Base de Datos, Simulaciones, Editor de Ecuaciones, Técnicas de Análisis.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En los últimos tiempos el desarrollo de la biología ha alcanzado un nivel elevado, donde las ciencias de la computación, las ciencias de la información, la ingeniería, las matemáticas y la estadística se han fusionado dando lugar a la disciplina hoy conocida como Bioinformática, con sus nuevas tecnologías para el tratamiento de la Información Genética (biochips, sistemas LIMS, bases de datos genómicas, sistemas de minería de datos, técnicas de cuantificación de la expresión génica) revolucionando la investigación biomédica y dentro de esta, la Biología de Sistemas.

La Biología de Sistemas es un área de investigación científica que se preocupa del estudio de procesos biológicos usando un enfoque sistémico, que 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 de los generados durante las simulaciones. [1]

La Bioinformática como ciencia proporciona los aspectos fundamentales a tener en cuenta a la hora de realizar análisis a los sistemas biológicos y a la vez está relacionada con los procesos informáticos que garantizan el almacenamiento de los datos generados en dichos sistemas. Para desarrollar estudios de estos sistemas, lo primero que se realiza es la modelación del mismo, a través de modelos matemáticos que lo representen. Estos modelos permiten predecir el comportamiento del proceso como un sistema dinámico.

En las últimas décadas, los investigadores no contaban con aplicaciones que les facilitaran el trabajo y les permitiera modelar y analizar sistemas biológicos. En la actualidad se adelanta a pasos agigantados en el análisis y diseminación de datos contando con más de cien software basados en estándares que permiten modelar, simular o realizar análisis sobre modelos de sistemas biológicos, teniendo como reto principal ofrecer una respuesta a la avalancha de datos generados a partir del análisis de estos sistemas.

A pesar de este actual desarrollo, aún se carece en el mundo de aplicaciones para el almacenamiento de estudios o simulaciones de sistemas biológicos, lo que quiere decir que aunque existan grandes repositorios de modelos, si un investigador quiere saber cómo funcionan dicho sistema, tiene que repetir el trabajo que ya otros han realizado.[2] Además, muchos de estos softwares no están disponibles para la comunidad científica, debido a que están representados por grandes intereses

comerciales.

Una de las mayores desventajas que existe hoy, es que todo está centrado en la modelación y la simulación, mientras que los análisis de los resultados son muy básicos, dejando el mayor peso en manos de los usuarios, quienes tienen que encontrar la forma para obtener información valiosa de los estudios realizados.

En Cuba, desde hace algunos años se ha venido impulsando el estudio y desarrollo de las Ciencias Biológicas. Hoy se cuenta con centros como: el Centro de Investigaciones Biológicas, 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) que se dedican a realizar simulaciones distribuidas de los sistemas biológicos modelados mediante Sistema de Ecuaciones Diferenciales (SED), para conocer la respuesta temporal del sistema, a partir de un conjunto de condiciones iniciales y una entrada de datos dada, careciendo de base de datos donde se almacene toda esta información para así poder utilizarla en estudios futuros. Es por ello que investigadores del CIM y de la UCI comenzaron a desarrollar un software para el estudio de sistemas biológicos, llamado BioSyS del que actualmente se desarrolla la versión 2.0.

En la primera versión del software BioSyS se implementó una base de datos que permitía gestionar la información generada en los estudios realizados a modelos matemáticos de sistemas biológicos, la cual presenta algunas limitaciones, que constituye una barrera para los investigadores.

Por lo que se plantea como **Problema Científico:** *¿Cómo contribuir a la gestión de la información generada en los procesos de análisis de las simulaciones de sistemas biológicos con la versión 2.0 de BioSyS?*

Se plantea como **Objeto de Estudio** el Proceso de Almacenamiento de la información generada en los procesos de análisis de las simulaciones de sistemas biológicos.

El **Campo de Acción** son las Bases de Datos para la gestión de la información generada en los procesos de análisis de las simulaciones de sistemas biológicos.

El **Objetivo General** consiste en Desarrollar nuevas funcionalidades a la BD BioSyS 1.0 permitiéndole al investigador gestionar la información referente al proceso de análisis a las simulaciones.

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

- Rediseñar la Base de Datos BioSys 1.0
- Implementar el módulo de Base de Datos
- Realizar pruebas que validen el correcto funcionamiento de la aplicación.

Para dar cumplimiento a los objetivos trazados es necesario dar cumplimiento a las siguientes **Tareas:**

1. Revisión bibliográfica sobre el Editor de Ecuaciones
2. Revisión bibliográfica sobre los Métodos y Técnicas de Análisis
3. Análisis de herramientas para diseñar la BD
4. Revisión del Levantamiento de Requisitos en el Modulo de Análisis y el Editor de Ecuaciones
5. Rediseño de la Base de Datos BioSys 1.0
6. Implementación de la capa de Acceso a Datos
7. Diseño de posibles pruebas
8. Realización de pruebas al sistema

El presente trabajo de diploma está estructurado en tres capítulos:

En el Capitulo 1 se hace una fundamentación teórica del tema a abordar, se explican conceptos fundamentales acerca de Biología de Sistemas y las simulaciones, además de realizarse un estudio de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.

En el Capítulo 2 se explican cada una las características que deberá tener la Base de Datos, mostrando los requisitos funcionales y los no funcionales. Los requisitos funcionales no son más que las tareas o actividades que el sistema será capaz de hacer y los no funcionales son las cualidades que hacen al producto atractivo, usable, rápido y confiable. Se establece la línea base de la arquitectura y se realizan todos los diagramas mostrando una visión más exacta del sistema.

En el Capítulo 3 se explican los temas relacionados con la implementación y prueba a la Base de Datos, validando así el correcto funcionamiento de la misma. Se describe de forma general las principales consultas desarrolladas que responden a las necesidades del Modulo de Análisis y el Editor de Ecuaciones.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1-Introducción

Los objetivos de este capítulo son: abordar diferentes temas que explicarán, la necesidad de almacenar la información relacionada con el editor de ecuaciones y del proceso de análisis a las simulaciones. Además, explicar conceptos fundamentales acerca de Biología de Sistemas así como, dar una breve descripción de las metodologías y herramientas de desarrollo utilizadas, que posibilitan que el presente trabajo logre sus objetivos.

1.2-Biología de Sistemas

La Biología de Sistemas es una ciencia emergente, cuyo desarrollo depende en gran medida de la aplicación de algoritmos y técnicas computacionales, que ayuden a la modelación, simulación y análisis de los problemas a los que se enfrenta. [2] La investigación en esta área, ha ido evolucionando de manera creciente. Los avances experimentales han ido en aumento, lo cual ha transformado esta rama de la biología en una disciplina rica en datos, donde el objetivo principal es comprender los procesos biológicos como un conjunto y no los sistemas como partes aisladas.

1.2.1-Conceptos de Biología de Sistemas

Los orígenes de la “Biología de Sistemas”, se sitúan en el siglo XIX, en relación a los conocimientos sobre embriología y los análisis matemáticos de redes. En los últimos tiempos se exponen diferentes conceptos de Biología de Sistema usando un enfoque sistémico con sus dos componentes fundamentales, el experimental y el computacional, enfocándolo desde diferentes puntos de vista. Entre los más generales se pueden citar:

“La Biología de Sistemas contemporánea surge de la convergencia de dos líneas de investigación en biología molecular: una primera que se fundamenta en los trabajos seminales sobre la naturaleza del material genético, la caracterización estructural de las macromoléculas y los subsiguientes avances en tecnologías recombinantes y de alto poder de resolución; la segunda, que supone una base más alejada de la tradicional biología molecular pero no menos robusta para el concepto de biología de sistemas, tiene sus raíces en la teoría de la termodinámica de los procesos irreversibles (lejos del equilibrio), en la resolución de las rutas bioquímicas y el reconocimiento de los retrocontroles en los

organismos unicelulares, así como en el creciente reconocimiento de la existencia de redes en biología. ” [3]

“..La Biología de Sistemas implica el análisis de todos los componentes del sistema biológico, que incluye un análisis profundo de cómo se expresan los genes y sus interacciones complejas dentro de una célula, tejido o todo el organismo...” [4]

“La Biología de Sistemas es una disciplina académica que pretende integrar diferentes niveles de información con el fin de entender cómo funcionan los sistemas biológicos. Intenta crear modelos comprensibles de sistemas mediante el estudio de las relaciones y las interacciones entre las diferentes partes de un sistema biológico. ” [5]

“La Biología de Sistemas es un área de estudio emergente que se caracteriza por la integración de aproximaciones experimentales y computacionales en la comprensión de los sistemas biológicos. Esta disciplina permite estudiar los mecanismos que gobiernan los sistemas complejos y las interacciones existentes, entre los diferentes niveles de información biológica. ” [6]

1.2.2-Simulación de Sistemas

"La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias - dentro de los límites impuestos por un cierto criterio o un conjunto de ellos - para el funcionamiento del sistema". [7]

El objetivo de la simulación es descubrir el comportamiento de un sistema además de postular teorías o hipótesis que expliquen el comportamiento observado. La simulación por computadora se ha convertido en una parte útil del modelado de muchas situaciones de la vida real. Es una técnica numérica, para reproducir artificialmente un fenómeno o el funcionamiento de un sistema, utilizando modelos matemáticos y lógicos, que intenta encontrar soluciones analíticas a problemas que permiten la predicción del comportamiento de un sistema en el tiempo, dado un conjunto de parámetros y condiciones iniciales. La simulación es una herramienta poderosa que es ampliamente utilizada por los científicos en el estudio de sistemas complejos. Esta técnica incluye para estos estudios, tanto la construcción del modelo como su uso analítico.

El proceso de simulación de sistemas, es la experimentación con un conjunto de hipótesis de sistemas y herramientas computacionales necesarias para simular, sirviendo de base para el análisis de su comportamiento. Para realizar la simulación de un sistema, es necesario realizar primeramente un estudio previo, con el fin de determinar la interacción con otros sistemas, sus restricciones, variables e interrelaciones.

Ventajas de las simulaciones:

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

Desventajas de la simulación

1. Un buen modelo de simulación puede resultar bastante costoso; a menudo el proceso de desarrollar un modelo es largo y complicado.
2. La simulación no genera soluciones óptimas a problemas de análisis cuantitativos, en técnicas como cantidad económica de pedido, programación lineal o PERT (Técnica de Revisión y Evaluación de Programas). Por ensayo y error se producen diferentes resultados en repetidas corridas en el computador.
3. Los directivos generan todas las condiciones y restricciones para analizar las soluciones. El modelo de simulación no produce respuestas por sí mismo.
4. Cada modelo de simulación es único. Las soluciones e inferencias no son usualmente transferibles a otros problemas.

5. Siempre quedarán variables por fuera y esas variables pueden cambiar completamente los resultados en la vida real. [8]

1.3- Bases de Datos.

La informática es una herramienta de indudable valor para el desarrollo de investigación científica. Entre sus múltiples aplicaciones encontramos la gestión de bases de datos. Una Base de Datos es un conjunto de información pertenecientes a un mismo contexto, que está almacenada en forma sistemática, de manera tal, que los datos que la conforman puedan ser utilizados cuando sea necesario. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, existe una gran cantidad de herramientas destinadas a este fin, lo que ofrece un amplio rango de soluciones al problema de almacenar datos.

Según la variabilidad de los datos almacenados, se pueden encontrar bases de datos estáticas y dinámicas. Las Bases de Datos estáticas son de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones, mientras que las Bases de Datos dinámicas son aquellas donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos y operaciones fundamentales de consulta.

Para la presente investigación se parte de una Base de Datos ya existente, la cual es capaz de gestionar la información relacionada con los estudios y las simulaciones de sistemas biológicos, pero no almacena nada referente al proceso de análisis a las simulaciones y el editor de ecuaciones, lo que limita la evolución en las investigaciones de esta área. Debido a esto, se desea diseñar una nueva versión de la Base de Datos de BioSyS, que incluya estas nuevas funcionalidades facilitando así una mejor gestión de la información y ofreciendo una nueva forma de almacenar la información procedente del análisis a las simulaciones.

1.4- Versión 2.0 Base de Datos BioSyS. Nuevas Funcionalidades

No cumple objetivo alguno realizar simulaciones distribuidas si luego, los usuarios no cuentan con herramientas que posibiliten el análisis de los grandes volúmenes de datos almacenados. En la versión 1.0 de BioSyS se implementaron una serie de algoritmos y técnicas de minería de datos, que

posibilitaron los procesos de análisis. A pesar de estos avances, dichos resultado no eran almacenados, por lo que si se quería consultar las clasificaciones resultantes, era necesario repetir todo el proceso de análisis de las simulaciones, lo que evidenció la necesidad de modificar el diseño de la BD de BioSyS e incorporarle nuevas funcionalidades que garantizaran el almacenamiento de la información referente al proceso de análisis a las simulaciones, a partir de las Técnicas de Análisis.

Otra problemática era que no se almacenaba la información relacionada con el Editor de Ecuaciones, necesitando esta funcionalidad, agrupando las ecuaciones utilizadas en cada uno de los Modelos Matemáticos por Categorías.

1.4.1- Técnicas de Análisis

De nada sirve realizar simulaciones distribuidas, si después no se pone en manos de los usuarios, herramientas que ayuden al análisis de los grandes volúmenes de datos almacenados. [2]

Las técnicas utilizadas para realizar metanálisis sobre los resultados de las simulaciones son:

1.4.1.1- Clustering

Es poco probable que se pueda explorar de forma manual toda la información que de cada modelo se almacena en la BD. Es por ello que se hace necesario el uso de herramientas que permitan hacer minería de estos datos y faciliten la comprensión del sistema en estudio. Una de las técnicas de minería incorporadas a la aplicación propuesta son los algoritmos de Clustering. Estos algoritmos permiten agrupar los vectores formados por los valores finales de cada simulación de acuerdo a la distancia existente entre ellos. El usuario puede seleccionar además el algoritmo que desea utilizar, en este caso *k-Means, X-Means o CobWeb*. Una vez definida esto, el sistema se encarga de realizar el agrupamiento y mostrar el mismo, en forma de salida gráfica. El número de clusters que se obtienen se puede asociar con la cantidad de comportamientos diferentes, a los que tiende el sistema. [2]

1.4.1.2- Clasificación Manual

La clasificación manual no es más que la actividad realizada por el usuario, que consiste en explorar la Base de Datos y clasificar un grupo de simulaciones de acuerdo a clases definidas por él mismo.

1.4.1.3- Clasificación por Weka

Weka: Software que implementa múltiples algoritmos para la realización de tareas de minería de datos. Se utilizan sus algoritmos de agrupamiento (clustering) para clasificar las simulaciones realizadas de acuerdo a su comportamiento. [2]

Una de las funcionalidades del weka que se ha incorporado a la aplicación, es la clasificación. En este caso lo que se persigue es que, a partir de una serie de simulaciones que el usuario clasifique de forma manual, se pueda generar un modelo que permita la posterior clasificación de las restantes simulaciones almacenadas en la Base de Datos. Además, se quiere que de estas clasificaciones se puedan inferir reglas, las cuales serán utilizadas en otros tipos de análisis. [2]

1.4.1.4- Análisis por Reglas

El último análisis implementado dentro de BioSyS, se ha llamado análisis por reglas. Este análisis permite crear una especie de gráfica de bifurcaciones donde los estados cualitativamente diferentes se describen mediante reglas lógicas. La idea es que, una vez encontrados estos comportamientos, ya sea usando análisis de clústeres o clasificaciones, el usuario puede estudiar hacia que comportamiento tiende el sistema, cuando se varían determinados parámetros de a dos en dos. [2]

1.4.2- Editor de Ecuaciones

Los Editores de Ecuaciones constituyen aplicaciones computacionales encargadas de formatear y alinear correctamente los símbolos de las fórmulas matemáticas.

Para la versión 2.0 de la Base de Datos BioSyS, se crea una Biblioteca de Expresiones, que permite a los investigadores agrupar las mismas por categorías, además de conservar los fragmentos de fórmula que más se utilizan.

1.5- Herramientas y metodologías

Para alcanzar los objetivos trazados, se deben definir los tipos de herramientas y metodologías que se van a utilizar durante todo el proceso de desarrollo. En este epígrafe se explicará en detalle qué herramientas y metodología se usarán.

1.5.1- Metodologías de desarrollo de software

La selección de qué herramienta utilizar durante todo el ciclo de desarrollo del software constituye la principal disyuntiva a la que se enfrentan los desarrolladores de hoy. Una buena selección tanto de las herramientas a utilizar, como de la metodología que se desea emplear, favorece la calidad del software deseado, pues de ahí se deriva el papel que deben desempeñar cada miembro dentro del equipo de desarrollo, así como las actividades que tienen que cumplir cada uno de ellos, los artefactos que deben ser creados y cada detalle de la información del producto que se debe alcanzar como resultado de toda la actividad realizada.

Actualmente, existen diferentes metodologías y técnicas para el desarrollo de productos, las que son analizadas y definidas en la arquitectura del proyecto, quedando como propuesta final la metodología OpenUP, a la cual se le realizaron algunas adaptaciones debido a que no presenta el rol de desarrollador de bases de datos.

1.5.1.1- OpenUP/Basic

OpenUP/Basic es un FrameWork de procesos de desarrollo de software de código abierto. Es un proceso modelo y extensible, dirigido a la gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

Este proceso de desarrollo unificado está basado en Rational Unified Process (RUP), desarrollado por IBM. Permite un abordaje ágil al proceso de desarrollo de software, con sólo proveer un conjunto simplificado de contenidos, fundamentalmente relacionados con orientación, productos de trabajo, roles, y tareas.

OpenUP/Basic es un proceso interactivo de desarrollo de software simplificado, completo y extensible. Es un proceso para pequeños equipos de desarrollo que valoran los beneficios de la colaboración y de los involucrados con el resultado del proyecto, por encima de formalidades innecesarias. [9]

1.5.2- Lenguaje de Modelado

1.5.2.1- Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML - Unified Modeling Language), es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales, como son: procesos de negocio y funciones de sistema, además de cosas concretas como lo son: escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. [10]

Los beneficios que se consiguen al utilizar UML son varios, por un lado el uso de lenguajes visuales facilitan su asimilación y entendimiento por parte del equipo de desarrollo; el tiempo invertido en el desarrollo de la arquitectura se minimiza; la detección y resolución de errores se agiliza siempre y cuando se haga uso de herramientas adecuadas de diagnóstico y depuración; y la trazabilidad y documentación del proyecto se realiza de una forma ordenada y guiada por los casos de uso. Pero si hay una ventaja que se destaca sobre todas las demás, es la notable efectividad y productividad que se consigue en labores de diseño arquitectónico y mantenimiento, haciendo uso de UML frente a la realización de las mismas tareas, en ausencia de modelos.

1.5.3- Herramientas CASE

Debido a las exigencias y el esfuerzo adicional que requiere la elaboración de los modelos y la gran cantidad de documentación que se genera, se hace imprescindible la utilización de las herramientas CASE.

La Ingeniería de Software Asistida por Computación (CASE, por sus siglas en Inglés), permite organizar y manejar la información de un proyecto informático, mejorar la comunicación entre los participantes y hace que un sistema por muy complejo que parezca se torne más flexible y más comprensible a sus desarrolladores.

Las herramientas Case son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo del sistema de información.

Para el diseño de la Base de Datos, se hizo necesaria la utilización de dos herramientas CASE para el modelado de los artefactos a obtener:

- Visual PARADIGM.
- DBDesigner Fork para diseñar el modelo de datos.

1.5.3.1- Visual Paradigm.

Es una herramienta CASE que utiliza UML como lenguaje de modelado, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, orientados a objetos, construcción, pruebas y despliegue.

Visual Paradigm constituye una herramienta que se caracteriza por su robustez, usabilidad y portabilidad. Permite realizar ingeniería tanto directa, como inversa (proceso ingenieril en el que se obtienen modelos conceptuales a partir de los artefactos software como código fuente, ejecutables, binarios y ficheros intermedios).

Es colaborativa, ya que soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación correspondiente al proyecto automáticamente en varios formatos como Web o .Pdf, y permite llevar el control de versiones.

1.5.3.2- DBDesigner Fork

DBDesigner Fork es un potente diseñador de Bases de Datos, que integra el diseño entidad-relación y la creación de Bases de Datos. Dispone de una interfaz profesional y de detallados manuales de uso. Es multiplataforma y facilita la creación de modelos de datos.

Este programa de diseño de Bases de Datos, combina características y funciones profesionales de fácil uso, a fin de ofrecer un método efectivo para gestionar bases de datos. Permite además, administrar la Base de Datos, diseñar tablas, hacer peticiones SQL manuales y conectarse directamente a Bases de Datos ya existentes, lo cual condiciona la realización de una Ingeniería Inversa, es decir, obtener el modelo Entidad-Relación de dichas Bases de Datos.

Dada su arquitectura basada en plugins, es fácilmente extensible para trabajar con diferentes servidores de base de datos. Por defecto se proporcionan dos plugins, uno para PostgreSQL y otro para MySQL. [11]

1.5.4- Gestores de Bases de Datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una Base de Datos. Ayuda a realizar acciones como definición

de los datos, mantenimiento de la integridad de los datos dentro de la base de datos, control de la seguridad y privacidad y manipulación de los datos. [12]

1.5.4.1- MySQL

MySQL es un sistema de administración de bases de datos relacionales, licenciado bajo la GPL de la GNU.

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Este consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido y más fácil de manejar.

MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación, además de ejecutarse en la inmensa mayoría de sistemas operativos y en la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

Dentro de las principales características de MySQL se hallan:

1. Soporta gran cantidad de tipos de datos para las columnas.
2. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
3. Gran portabilidad entre sistemas.
4. Soporta hasta 32 índices por tabla.
5. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

1.5.4.2- PostgreSQL

PostgreSQL, soporta SQL92 y SQL99 y ofrece un conjunto de características modernas que lo convierten en el gestor de Base de Datos de código abierto más avanzado del mundo, tales como: consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia y arrays.

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Es un gestor altamente extensible ya que soporta operadores, métodos de acceso y tipos de datos definidos por el usuario. PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la Base de Datos.
La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++ y Pike.

1.5.5- Lenguaje de Marcado

1.5.5.1- Lenguaje de marcas extensible (XML)

No es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes. XML es un metalenguaje, que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados. Es un subconjunto de SGML especializado en la gestión de información para la Web. [13]

XML es una tecnología sencilla, que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad, ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Es un estándar de fácil proceso tanto por humanos como por cualquier software, separa la información o contenido de su presentación o formato y está diseñado para ser procesado en cualquier lenguaje o alfabeto. No pertenece a ninguna compañía en específico lo cual lo hace libre y constituye un estándar internacionalmente reconocido.

1.5.6- Lenguaje de Programación

1.5.6.1- Lenguaje Java

Java es un Lenguaje de Programación Orientado a Objetos (POO) ideado por Sun Microsystem e independiente de la plataforma. Tiene muchas cosas que lo hacen parecerse a C++ pero es mucho

más seguro y sobre todo mucho más fácil de leer. Es un lenguaje de propósito general por lo que se podría crear cualquier tipo de aplicación con él, pero su mayor éxito se produce en Internet, con los famosos Applets, las aplicaciones Cliente/Servidor o las tan de moda "Java Server Pages".

Algunas de sus características más notables son:

- Robusto
- Gestiona la memoria automáticamente
- No permite el uso de técnicas de programación inadecuadas
- Multithreading
- Cliente-servidor
- Mecanismos de seguridad incorporados (limitan el acceso a recursos de las máquinas donde se ejecuta)
- Herramientas de documentación incorporadas

1.5.6.2- Lenguaje de Consulta Estructurado (SQL)

El SQL (Structured Query Language), lenguaje de consulta estructurado, es un lenguaje surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar de lenguaje de bases de datos y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores y entre sus funciones principales está, la definición, control y gestión de la base de datos. [14]

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Más específicamente SQL está definido en torno al modelo de bases de datos relacionales, ventaja que lo impone como el sistema de mayor aceptación. Algunas de estas ventajas son: [15]

- Marco teórico sólido, fundamentado en el álgebra relacional
- Simplicidad de conceptos (modelo de base de datos: tablas=líneas x columnas)
- Definición de vínculos en la consulta, brindando gran flexibilidad
- Fácil y rápido aprendizaje.
- Arquitectura cliente-servidor.
- Integración con cualquier lenguaje de programación

- Estandarización.

1.5.7- Herramientas de desarrollo

1.5.7.1- NetBeans IDE

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. [16] Soporta el desarrollo de todo tipo de aplicación Java: J2SE, web, EJB y aplicaciones móviles.

Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring. Todas las funciones del IDE son provistas por módulos, donde cada uno proporciona una función bien definida, tales como el soporte de Java, edición o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java.

1.5.7.2- Hibernate

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones y un gran número de tipos de datos. De una manera muy rápida y optimizada podremos generar Bases de Datos en cualquiera de los entornos soportados: Oracle, DB2, My SQL, etc.

Y lo más importante de todo, es open source, lo que supone, entre otras cosas, que no se tiene que pagar nada por adquirirlo.

Uno de los posibles procesos de desarrollo consiste en que, una vez que se tenga el diseño de datos realizado, con mapear este a ficheros XML siguiendo la DTD de mapeo de Hibernate, se puede generar el código de los objetos persistentes en clases Java y también crear Bases de Datos independientemente del entorno escogido. [17]

Hibernate no sólo se encarga de mapear clases Java a tablas de base de datos (y de tipos de datos de Java a tipos de datos SQL), sino que también provee facilidades de consulta y recuperación de datos y puede reducir significativamente el tiempo de desarrollo, que de otra forma se gasta en el manejo de los datos en SQL y JDBC.

1.5.7.3- MySQL Query Browser

MySQL Query Browser es una utilidad para trabajar con una Base de Datos MySQL. Es un editor de sentencias SQL visual, que además incorpora herramientas para optimizar las consultas. Dispone también de un editor de tablas y registros, que permite crear nuevas tablas o cambiar las existentes y la posibilidad de cambiar los registros, es decir, los datos almacenados en las tablas. [18]

Esta herramienta se compone de:

- Editor de Sentencias SQL, el cual permite crear sentencias de manera visual o manual y mostrar el historial de las que se han ido realizando.
- Visor de Resultados, posibilitando que se puedan examinar los resultados devueltos por las consultas y comparar los resultados obtenidos entre varias sentencias SQL.
- Navegador de objetos, para manejar las bases de datos, favoritos o historial, y permite visualizar un esquema de las distintas bases de datos, con sus distintas tablas y registros, que se utilizarán luego para generar las sentencias visualmente.
- Visor de información, el cual hace referencia a las base de datos MySQL, con acceso a la sintaxis y librerías MySQL.

1.5.7.4- MySQL Administrator

MySQL Administrador Tools es el nuevo software de administración de servidores de Bases de Datos de MySQL que ha creado MySQL AB. Se trata de un software multiplataforma, que por el momento se encuentra disponible para Linux y Microsoft Windows y que cuenta con un entorno gráfico de usuario muy intuitivo. [19]

Esta herramienta permita realizar un conjunto de tareas administrativas sobre servidores MySQL como por ejemplo:

- Configurar las opciones de inicio de los servidores.
- Monitorizar las conexiones al servidor.
- Administrar usuarios.
- Monitorizar el estado del servidor, incluyendo estadísticas de uso.
- Gestionar copias de seguridad y recuperaciones.

- Visualizar catálogos de datos.

Muchas de las opciones de configuración de la base de datos son muy sencillas de entender y de utilizar, al menos las opciones más básicas. En conjunto con MySQL Query Browser, se permite gestionar cualquier aspecto de una base de datos MySQL.

1.5.7.5- EMS DataGenerator 2005 para MySQL

EMS Data Generator para MySQL es una herramienta para generar datos de prueba a tablas de bases de datos MySQL. La aplicación asistente te permite definir tablas y campos para generar datos, configurar valores de rangos, generar campos de tipo carácter por máscara, cargar valores desde archivos para campos de tipo objetos binarios (BLOB), obtener listas de valores desde consultas SQL y muchas otras características más para generar datos de prueba de manera simple y directa.

Sus características están dadas por una interfaz de usuario amigable, permite la generación de datos a varias tablas desde diferentes bases de datos en un solo ordenador anfitrión (host), provee un soporte para todos los tipos de datos de MySQL, incluyendo los tipos ENUM y SET, diferentes tipos de generación por cada campo, incluyendo lista, azar (random), generación incremental de datos y otros, capacidad para usar resultados de consultas SQL como lista de valores para la generación de datos, control automático sobre integridad referencial para la generación de datos a tablas vinculadas, amplia variedad de generación de parámetros para cada tipo de campo, capacidad para configurar valores NULL (nulos) para ciertos casos, posibilidad de salvar todos los parámetros de generación, para comenzar la sesión del asistente actual, utilitario de líneas de comandos para generar datos usando el archivo que contiene la plantilla. [20]

1.5.7.6- Apache JMeter

JMeter es una herramienta de carga para llevar acabo simulaciones sobre cualquier recurso de Software. Inicialmente diseñada para pruebas de estrés en aplicaciones web, hoy en día, su arquitectura ha evolucionado no sólo para llevar a cabo pruebas en componentes habilitados en Internet (HTTP), sino además en Bases de Datos , programas en Perl , requisiciones FTP y prácticamente cualquier otro medio.

Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de requisiciones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción.

En este sentido, simula todas las funcionalidades de un Navegador ("Browser"), o de cualquier otro cliente, siendo capaz de manipular resultados en determinada requisición y reutilizarlos para ser empleados en una nueva secuencia.

El componente principal de JMeter es denominado *Plan de Prueba* o *Test Plan*, en él se definen todos los aspectos relacionados con una prueba de carga, como : parámetros empleados por requisición, tipo de reportes a generarse con los resultados obtenidos, la posible reutilización de requisiciones compuestas por usuarios, entre otros aspectos. [21]

1.6- Sistema Operativo

Ubuntu se considera una de las más importantes distribuciones de GNU/Linux en el mundo. Está basado en Debian GNU/Linux y sus principales objetivos es la facilidad y libertad de uso del mismo, la fluida instalación y los períodos de lanzamiento que tiene el producto, cada 6 meses.

Incluye una cuidadosa selección de los paquetes de Debian, para incluir solo aplicaciones importantes y de alta calidad; y mantiene un poderoso sistema de gestión de paquetes que permite instalar y desinstalar programas de una forma fácil y limpia, a diferencia de la mayoría de las distribuciones, que vienen con una enorme cantidad de software que puede o no ser de utilidad. [22]

1.7- Patrón DAO

Data Access Object (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de Datos o un archivo. Los principales objetivos de este patrón son: abstraer y encapsular los accesos, gestionar las conexiones a la fuente de datos y obtener los datos almacenados. Constituye una solución al problema del diferencial de la impedancia entre un programa de aplicación orientado a objetos y una base de datos relacional, empleando únicamente la interfaz de programación (API). [23]

El patrón DAO posibilita aislar las conexiones a la fuente de datos en una capa muy bien identificada y mantenida; baja el nivel de acoplamiento entre clases, lo cual reduce la complejidad de realizar

cambios y permite además que cualquier objeto no requiera conocimiento directo del destino final de la información que se manipula.[23]

1.8- Conclusiones

- Para el desarrollo de la aplicación se utilizó la metodología OpenUP debido a su fácil adaptación a proyectos pequeños de tres ó seis personas, además es ágil e incremental; y aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.
- El lenguaje utilizado es Java por ser multiplataforma y la herramienta más factible para el desarrollo de esta aplicación fue el NetBeans IDE, e Hibernate como framework.
- Como herramienta de modelado de artefactos se empleó el Visual Paradigm y para el diseño del modelo de datos el DBDesigner Fork.
- Para el almacenamiento y gestión de los datos que se almacenan se utilizo como gestor de base de datos MySQL y PostgreSQL, respectivamente, por ser sistemas de administración de Bases de Datos, fuertes, seguros y además multiplataforma.
- Para hacerle pruebas a la Base de Datos se utilizó el Generador de Datos EMS DataGenerator 2005 para MySQL y la herramienta Apache JMeter.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1-Introducción

En el presente capítulo se explican cada uno de los requisitos funcionales y los no funcionales que describen la Base de Datos, así como otras características con que cuenta la misma. En el caso de los requisitos funcionales se va a hacer referencia a las tareas o actividades que el sistema será capaz de hacer; y los no funcionales, a las cualidades que hacen al producto atractivo, usable, rápido y confiable. Se establece la línea base de la arquitectura y se realizan todos los diagramas para dar una visión más exacta del sistema.

2.2-Descripción de la Arquitectura

La Base de Datos se encuentra estructurada en capas para la implementación de la misma, siguiendo el patrón de arquitectura en capas. Se representan tres capas de ahí que exista una capa de presentación donde se encuentran todas las interfaces de usuario referentes al sistema en general partiendo de que la Base de Datos está integrada al mismo y no separada de este, una capa de Negocio encargada de gestionar la información de la Base de Datos y una de Datos donde residen los mismos. Las capas de negocio y la de Datos describen específicamente la arquitectura de la Base de Datos, la cual contiene trece tablas en modelo físico, que serán utilizadas por los módulos de análisis y el Editor de Ecuaciones del proyecto, haciendo uso principalmente de las tablas Manual, Reglas, Clustering y ClasificacionWeka; las cuales recogen la información de mayor peso relacionadas con el proceso de análisis de las simulaciones.

El objetivo de la arquitectura en Capas es organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes de capas inmediatamente inferiores. Las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar qué se puede reutilizar, y proporciona una estructura que nos beneficia a la hora de tomar decisiones sobre qué partes comprar y qué partes construir.

La Base de Datos se encuentra dividida en tres capas lógicas distintas, (definidas según las

características del patrón DAO):

La primera capa hace referencia al sistema BioSyS viendo a la Base de Datos unida al sistema, esta capa se denomina capa de presentación que contiene todas las clases de interfaz de usuarios que representan las pantallas de la aplicación que el usuario ve, normalmente consiste en una interfaz gráfica amigable para el usuario.

La capa intermedia o capa de negocio es la encargada de apoyar el acceso al SGBD. Se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al SGBD almacenar o recuperar datos de él. En esta capa el patrón DAO define por cada clase persistente, una clase interfaz y a su vez, una clase concreta, donde se implementan los métodos definidos, permitiendo obtener los datos mediante el uso de objetos, los que van a estar definidos en la clase factory, que es la encargada de establecer la conexión con la capa de acceso a datos.

En la capa de datos es donde residen los datos y además es la encargada de acceder a los mismos. Formada por uno o más SGBD para realizar el proceso de almacenamiento de datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio. La separación entre la capa de presentación (interfaz amigable al usuario) y la capa intermedia (patrón DAO) añade una enorme flexibilidad al diseño de la aplicación, donde pueden construirse y desplegarse múltiples interfaces de usuario, sin cambiar en absoluto la lógica de la aplicación, siempre que esté presente una interfaz claramente definida en la capa de presentación. La capa de Acceso a Datos es generada a partir del Framework Hibernate.

Estas dos últimas capas (Negocio y Datos) son las que describen específicamente la arquitectura de la Base de Datos BioSyS para la versión 2.0.

2.3-Levantamiento de Requisitos

Para el desarrollo de un software es preciso, inicialmente, comprender cada una de las necesidades que el sistema deberá cumplir, de acuerdo a las exigencias del cliente o usuario que usará el mismo. Este proceso de captura de requisitos, constituye un aspecto indispensable para que el proyecto finalice en el tiempo establecido y con la calidad requerida. [24]

2.3.1- Requisitos Funcionales

Son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales permiten expresar específicamente las responsabilidades del sistema que se propone. Permiten determinar de una manera clara lo que el sistema debe hacer y además no alteran las funcionalidades del producto, lo que quiere decir que los requerimientos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen. Luego de un estudio detallado de cada módulo que conforma a BioSyS y que aun su información no se almacenaba en la Base de Datos, se proponen como Requisitos Funcionales del módulo de base de datos, los siguientes: [24]

1. Gestionar la información de la clasificación

1.1. Clasificación por Clustering

1.1.1. Insertar clasificación por Clustering
1.1.2. Modificar clasificación por Clustering
1.1.3. Eliminar clasificación por Clustering

1.2. Clasificación por Reglas

1.2.1. Insertar clasificación por Reglas
1.2.2. Modificar clasificación por Reglas
1.2.3. Eliminar clasificación por Reglas

1.3. Clasificación Manual

1.3.1. Insertar clasificación Manuales
1.3.2. Modificar clasificación Manuales
1.3.3. Eliminar clasificación Manuales

1.4. Clasificación por Weka

1.4.1. Insertar clasificación Weka
1.4.2. Modificar clasificación Weka
1.4.3. Eliminar clasificación Weka

2. Gestionar parámetros del algoritmo de Clustering

2.1. Algoritmo CobWeb

2.1.1. Insertar parámetros

2.1.2. Eliminar parámetros

2.2. Algoritmo SimpleKMeans

2.2.1. Insertar parámetros

2.2.2. Eliminar parámetros

2.3. Algoritmo XMeans

2.3.1. Insertar parámetros

2.3.2. Eliminar parámetros

3. Gestionar biblioteca funcional

3.1. Gestionar Categoría

3.1.1. Insertar categorías

3.1.2. Eliminar categorías

3.1.3. Modificar categorías

3.2. Gestionar Elementos

3.2.1. Insertar elementos

3.2.2. Eliminar elementos

3.2.3. Modificar elementos

3.3. Gestionar Dimensión

3.3.1. Insertar dimensión

3.3.2. Eliminar dimensión

3.3.3. Modificar dimensión

4. Gestionar Fichero de los Modelo Generados

4.1. Insertar fichero del modelo generado

4.2. Eliminar fichero del modelo generado

4.3. Modificar fichero del modelo generado

2.3.2- Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe

pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a los requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán seguro, conveniente o agradable debe ser, lo que pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. [24]

Requisitos de Software

El sistema operativo a utilizar deberá ser Linux, de acuerdo a los requisitos impuestos por el cliente. La Comunicación Cliente-Servidor será a una velocidad constante de 10/100 Mbps. Como herramienta CASE el Visual Paradigm y el DBDesigner Fork. Como gestor de Bases de Datos y herramientas de desarrollo, el MySQL, PostgreSQL y el NetBeans IDE respectivamente así como el Hibernate como framework.

Requisito de Restricciones en el diseño y la implementación

El lenguaje de programación utilizado es Java.

Requisitos de Hardware

Los servidores de bases de datos a utilizar deberán ser Pentium IV, los cuales deben contar con 512 MB de memoria RAM como mínimo, aunque lo ideal sería 1 GB. La capacidad del disco duro del servidor deberá ser de 40 GB como mínimo. Las computadoras clientes tendrán como mínimo 256 MB de memoria RAM.

Requerimientos de apariencia o interfaz externa

El sistema deberá presentar una interfaz externa que modele la acción actor-sistema, la cual deberá ser sencilla y fácil de comprender.

Requerimientos de Usabilidad

La aplicación dará la posibilidad de almacenar los resultados obtenidos a partir de los diferentes procesos de análisis realizados a las simulaciones de sistemas biológicos, mediante interfaces que permitan el intercambio de datos de forma tal que sea interactiva para el usuario.

Requerimientos de Seguridad

- Confidencialidad: La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
- Disponibilidad: Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

2.4-Diagrama de Clases del Diseño

2.4.1- Arquitectura en Capas

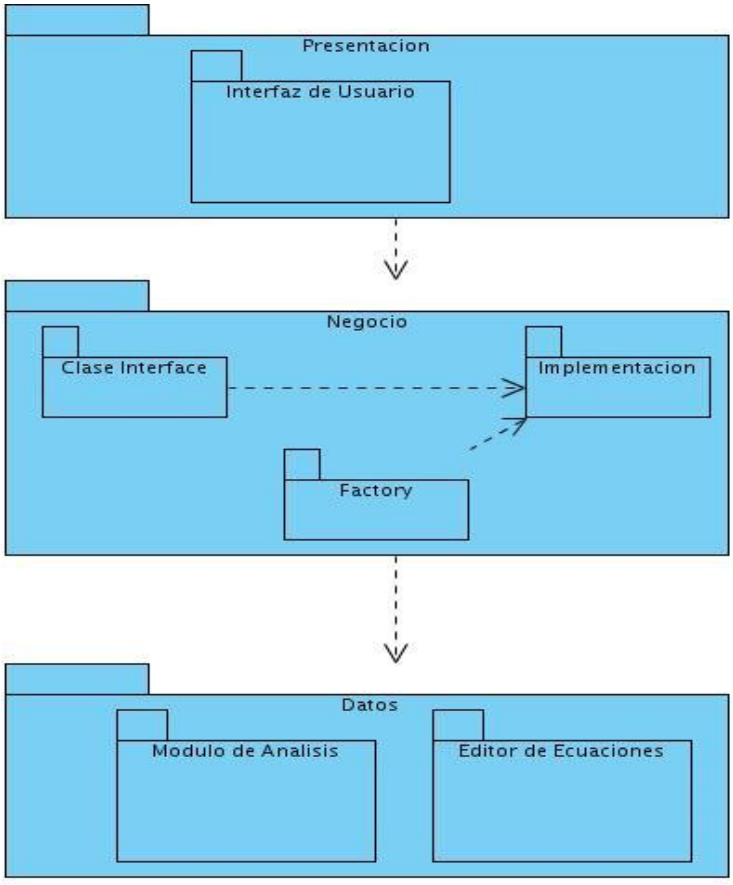


Figura 1: Arquitectura de la Base de Datos

2.4.2- Capa de Negocio

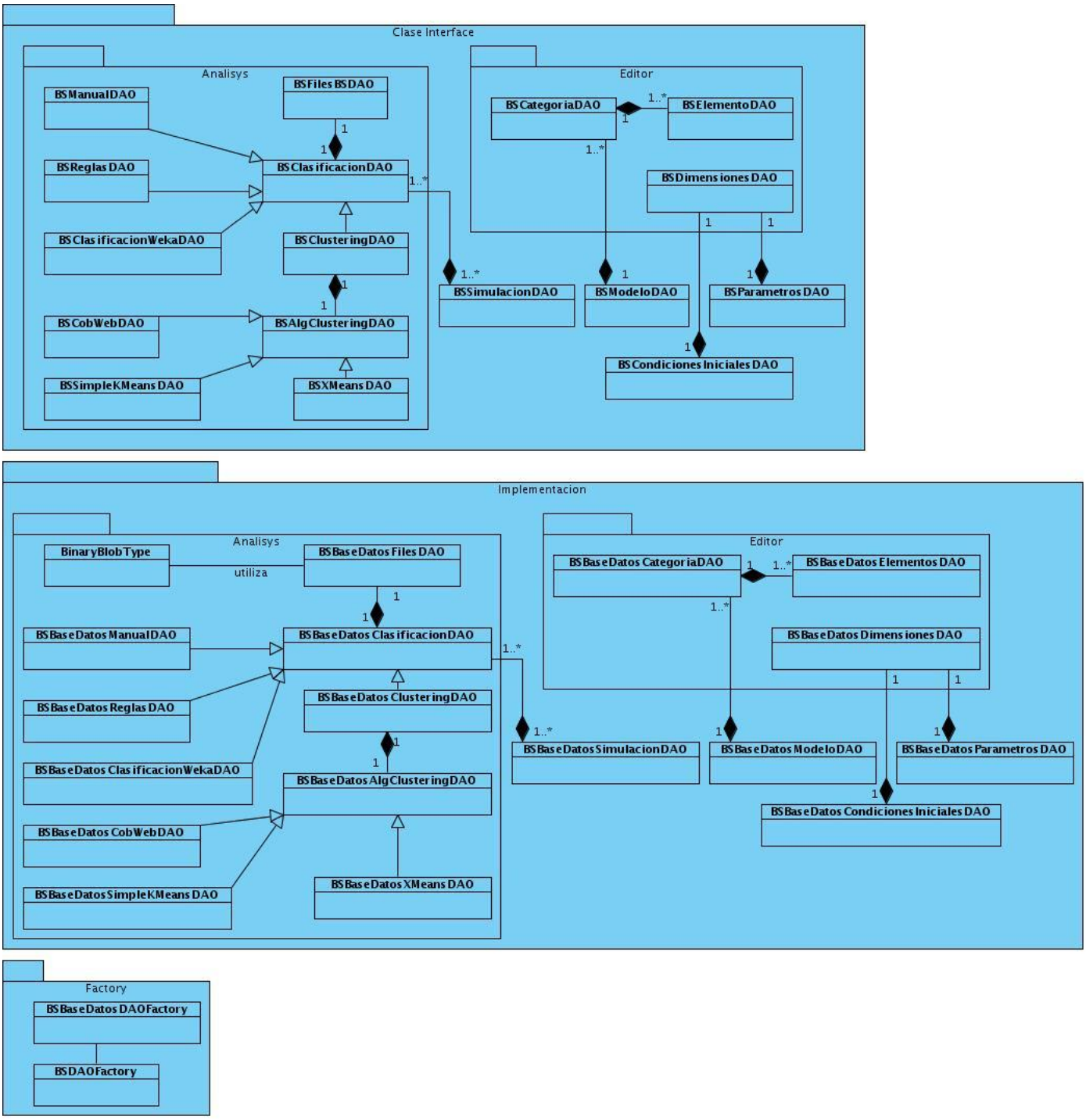


Figura 2: Capa de Negocio

2.4.3- Capa de Datos

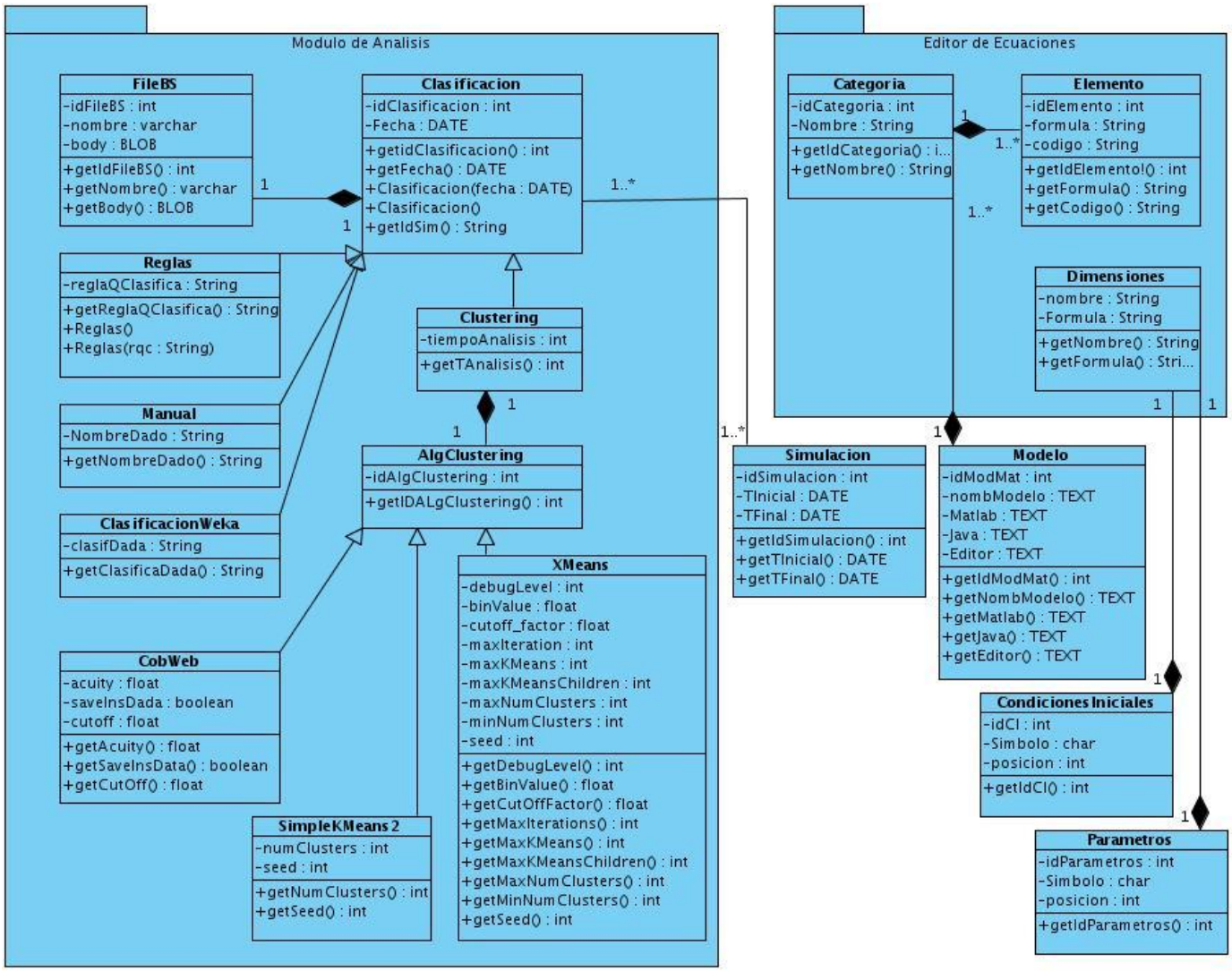


Figura 3: Capa de Datos

2.5- Diagrama de Clases Persistentes

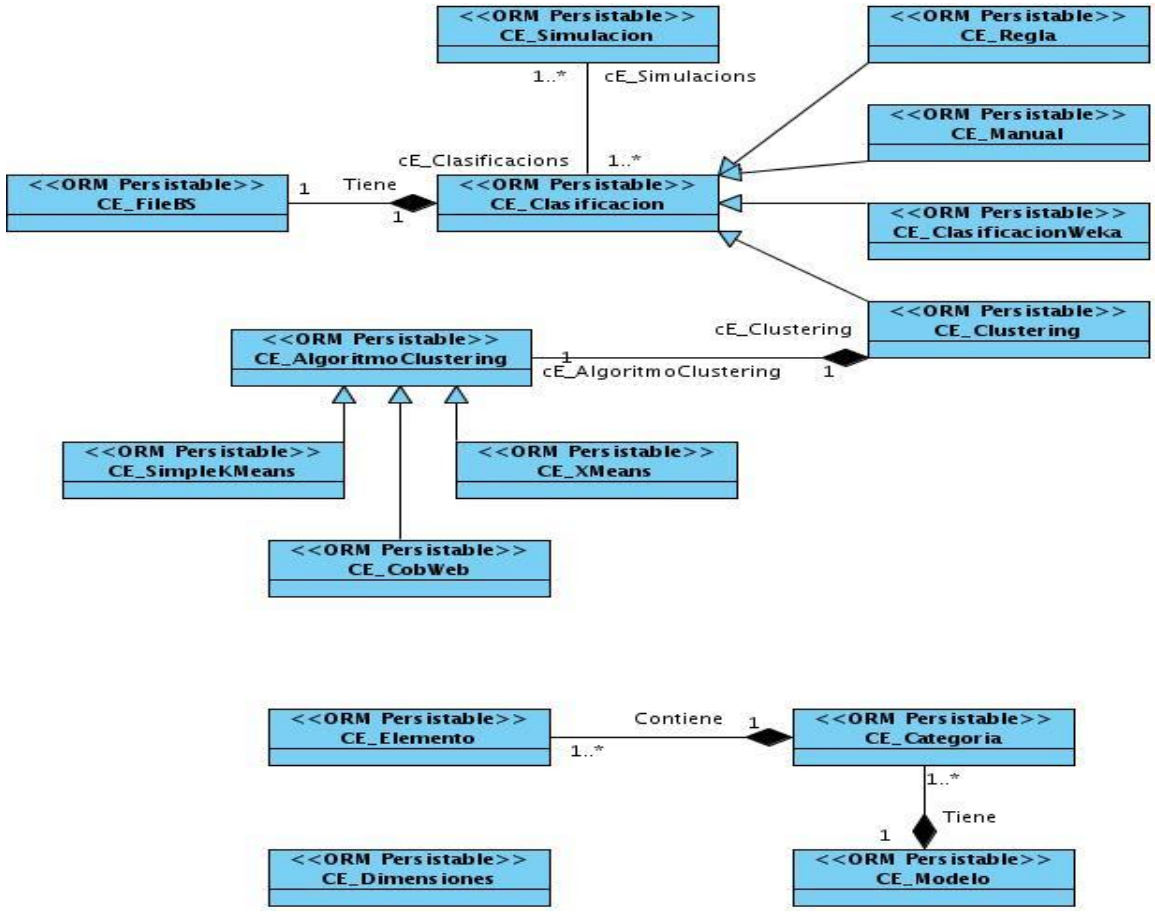


Figura 4: Diagrama de clases persistentes

2.6-Descripción de las entidades

Tabla 1: Descripción de la entidad Clasificación

Nombre: Clasificación	
Tipo de clase: entidad	
Atributo:	**Tipo:**
idClasificacion	int
Simulacion_idSimulacion	int
Fecha	DATE
Para cada responsabilidad:	

Nombre:	Gestionar la información de Clasificación por Clustering
Descripción:	Se insertan, modifican o eliminan Clasificaciones por Clustering
Nombre:	Gestionar la información de Clasificación por Reglas
Descripción:	Se inserta, modifica o elimina una Clasificación por Reglas
Nombre:	Gestionar la información de Clasificación Manual
Descripción:	Se inserta, modifica o elimina una Clasificación Manual

Tabla 2: Descripción de la entidad Clustering

Nombre: Clustering	
Tipo de clase: entidad	
Atributo:	**Tipo:**
tiempoAnalisis	int
Clasificacion_idClasificacion	int
AlgoritmoClustering_idAlgClustering	int
Para cada responsabilidad:	
Nombre:	Insertar la información de la clasificación por Clustering
Descripción:	Se llenan los campos necesarios para insertar una nueva clasificación por Clustering de las simulaciones a la base de datos
Nombre:	Modificar la información de la clasificación por Clustering
Descripción:	Para modificar una clasificación por Clustering, se recibe como parámetro el identificador de la Clasificación, con el cual se busca la clasificación por Clustering a modificar
Nombre:	Eliminar la información de la clasificación por Clustering
Descripción:	Para eliminar un Clasificación por Clustering de la base de datos, se recibe como parámetro el identificador del la Clasificación, con el cual se puede eliminar.

Tabla 3: Descripción de la entidad Manual

Nombre: Manual	
Tipo de clase: entidad	
Atributo:	**Tipo:**
Clasificacion_idClasificacion	int

NombreDado	varchar
Para cada responsabilidad:	
Nombre:	Insertar la información de la clasificación Manual
Descripción:	Se llenan los campos necesarios para insertar una nueva clasificación Manual de las simulaciones a la base de datos
Nombre:	Modificar la información de la clasificación Manual
Descripción:	Para modificar una clasificación Manual, se recibe como parámetro el identificador de la Clasificación, con el cual se busca la clasificación Manual a modificar
Nombre:	Eliminar la información de la clasificación Manual
Descripción:	Para eliminar un Clasificación Manual de la base de datos, se recibe como parámetro el identificador del la Clasificación, con el cual se puede eliminar.

Tabla 4: Descripción de la entidad ClasificacionWeka

Nombre: ClasificacionWeka	
Tipo de clase: entidad	
Atributo:	**Tipo:**
Clasificacion_idClasificacion	int
clasifDada	varchar
Para cada responsabilidad:	
Nombre:	Insertar la información de la clasificación por Weka
Descripción:	Se llenan los campos necesarios para insertar una nueva clasificación por Weka de las simulaciones a la base de datos
Nombre:	Modificar la información de la clasificación por Weka
Descripción:	Para modificar una clasificación por Weka, se recibe como parámetro el identificador de la Clasificación, con el cual se busca la clasificación por Weka a modificar
Nombre:	Eliminar la información de la clasificación por Weka
Descripción:	Para eliminar un Clasificación por Weka de la base de datos, se recibe como parámetro el identificador del la Clasificación, con el cual se puede eliminar.

Tabla 5: Descripción de la entidad Reglas

Nombre: Reglas	
Tipo de clase: entidad	
Atributo:	**Tipo:**
Clasificacion_idClasificacion	int
reglaQClasifica	varchar
Para cada responsabilidad:	
Nombre:	Insertar la información de la clasificación por Reglas
Descripción:	Se llenan los campos necesarios para insertar una nueva clasificación por Reglas de las simulaciones a la base de datos
Nombre:	Modificar la información de la clasificación por Reglas
Descripción:	Para modificar una clasificación por Reglas, se recibe como parámetro el identificador de la Clasificación, con el cual se busca la clasificación por Reglas a modificar
Nombre:	Eliminar la información de la clasificación por Reglas
Descripción:	Para eliminar un Clasificación por Reglas de la base de datos, se recibe como parámetro el identificador del la Clasificación, con el cual se puede eliminar.

Tabla 6: Descripción de la entidad AlgoritmoClustering

Nombre: AlgoritmoClustering	
Tipo de clase: entidad	
Atributo:	**Tipo:**
idAlgClustering	int
Para cada responsabilidad:	
Nombre:	Gestionar parámetros del algoritmo de Clustering CobWeb
Descripción:	Se insertan o eliminan los parámetros del algoritmo de Clustering CobWeb
Nombre:	Gestionar parámetros del algoritmo de Clustering SimpleKMeans
Descripción:	Se insertan o eliminan los parámetros del algoritmo de Clustering SimpleKMeans
Nombre:	Gestionar parámetros del algoritmo de Clustering XMeans
Descripción:	Se insertan o eliminan los parámetros del algoritmo de Clustering XMeans

Tabla 7: Descripción de la entidad CobWeb

Nombre: CobWeb	
Tipo de clase: entidad	
Atributo:	**Tipo:**
AlgoritmoClustering_idAlgClustering	int
acuity	float
cutoff	float
saveInsData	bool
Para cada responsabilidad:	
Nombre:	Insertar algoritmo de Clustering CobWeb
Descripción:	Se llenan los campos necesarios para insertar los parámetros del algoritmo de Clustering CobWeb a la base de datos
Nombre:	Eliminar algoritmo de Clustering CobWeb
Descripción:	Para eliminar un Algoritmo de Clustering CobWeb de la base de datos, se recibe como parámetro el identificador del Algoritmo de Clustering CobWeb que se va a eliminar.

Tabla 8: Descripción de la entidad SimpleKMeans

Nombre: SimpleKMeans	
Tipo de clase: entidad	
Atributo:	**Tipo:**
AlgoritmoClustering_idAlgClustering	int
numClusters	int
seed	int
Para cada responsabilidad:	
Nombre:	Insertar algoritmo de Clustering SimpleKMeans
Descripción:	Se llenan los campos necesarios para insertar los parámetros del algoritmo de Clustering SimpleKMeans a la base de datos
Nombre:	Eliminar algoritmo de Clustering SimpleKMeans
Descripción:	Para eliminar los parámetros del algoritmo de Clustering SimpleKMeans de la base de datos, se recibe como parámetro el identificador del Algoritmo de Clustering SimpleKMeans que se va a eliminar.

Tabla 9: Descripción de la entidad XMeans

Nombre: XMeans	
Tipo de clase: entidad	
Atributo:	**Tipo:**
AlgoritmoClustering_idAlgClustering	int
binValue	float
cutoff_factor	float
debugLevel	int
maxIterations	int
maxKmeans	int
maxKmeansChildren	int
maxNumClusters	int
minNumClusters	int
seed	int
Para cada responsabilidad:	
Nombre:	Insertar algoritmo de Clustering XMeans
Descripción:	Se llenan los campos necesarios para insertar el algoritmo de Clustering XMeans a la base de datos
Nombre:	Eliminar algoritmo de Clustering XMeans
Descripción:	Para eliminar un algoritmo de Clustering XMeans de la base de datos, se recibe como parámetro el identificador del Algoritmo de Clustering XMeans que se va a eliminar.

Tabla 10: Descripción de la entidad FileBS

Nombre: FileBS	
Tipo de clase: entidad	
Atributo:	**Tipo:**
idFileBS	int
nombre	varchar
body	BLOB
Para cada responsabilidad:	

Nombre:	Insertar fichero del modelo generado
Descripción:	Se inserta el fichero de modelo que se genera cuando se clasifican las simulaciones.
Nombre:	Modificar fichero del modelo generado
Descripción:	Para modificar un fichero se recibe como parámetro el nuevo nombre que desea ponerle al fichero y se modifica.
Nombre:	Eliminar fichero del modelo generado
Descripción:	Para eliminar un fichero se selecciona cual se desea eliminar.

Tabla 11: Descripción de la entidad Categoría

Nombre: Categoría	
Tipo de clase: entidad	
Atributo:	**Tipo:**
idCategoria	int
Nombre	varchar
Para cada responsabilidad:	
Nombre:	Insertar Categoría
Descripción:	Se llenan los campos necesarios para insertar una nueva Categoría a la biblioteca de funciones en la base de datos
Nombre:	Modificar Categoría
Descripción:	Para modificar una Categoría, se recibe como parámetro el identificador de la Categoría, con el cual se busca la Categoría a modificar
Nombre:	Eliminar Categoría
Descripción:	Para eliminar una Categoría de la base de datos se recibe como parámetro el identificador del la Categoría, con el cual se puede eliminar.

Tabla 12: Descripción de la entidad Elementos

Nombre: Elementos	
Tipo de clase: entidad	
Atributo:	**Tipo:**
idElemento	int
Categoria_idCategoria	int

formula	varchar
codigo	varchar
Para cada responsabilidad:	
Nombre:	Insertar Elemento
Descripción:	Se llenan los campos necesarios para insertar una nuevo Elemento a la biblioteca de funciones en la base de datos
Nombre:	Modificar Elemento
Descripción:	Para modificar un Elemento, se recibe como parámetro el identificador del Elemento, con el cual se busca el Elemento a modificar
Nombre:	Eliminar Elemento
Descripción:	Para eliminar un Elemento de la base de datos se recibe como parámetro el identificador del Elemento, con el cual se puede eliminar.

Tabla 13: Descripción de la entidad Dimensiones

Nombre: Dimensiones	
Tipo de clase: entidad	
Atributo:	**Tipo:**
Nombre	varchar
Formula	varchar
Para cada responsabilidad:	
Nombre:	Insertar Dimensiones
Descripción:	Se llenan los campos necesarios para insertar Dimensiones a la biblioteca de funciones en la base de datos
Nombre:	Modificar Dimensiones
Descripción:	Para modificar una Categoría se recibe como parámetro el identificador de las Dimensiones, con el cual se buscan las Dimensiones a modificar
Nombre:	Eliminar Dimensiones
Descripción:	Para eliminar Dimensiones de la base de datos se recibe como parámetro el identificador del las Dimensiones, con el cual se puede eliminar.

2.7-Modelo Físico de Datos

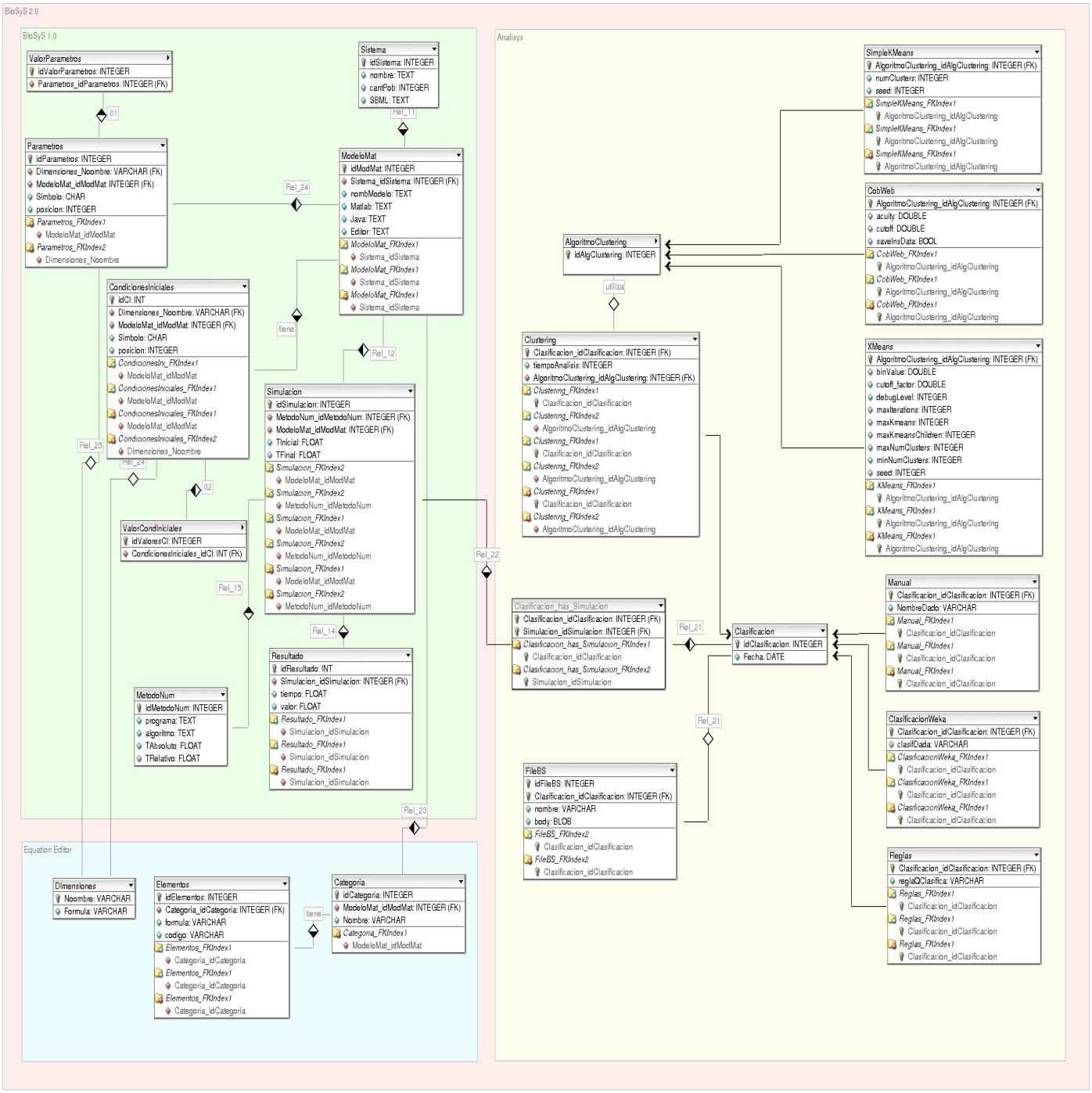


Figura 5: Modelo Físico de Datos

2.8-Modelo Entidad Relación

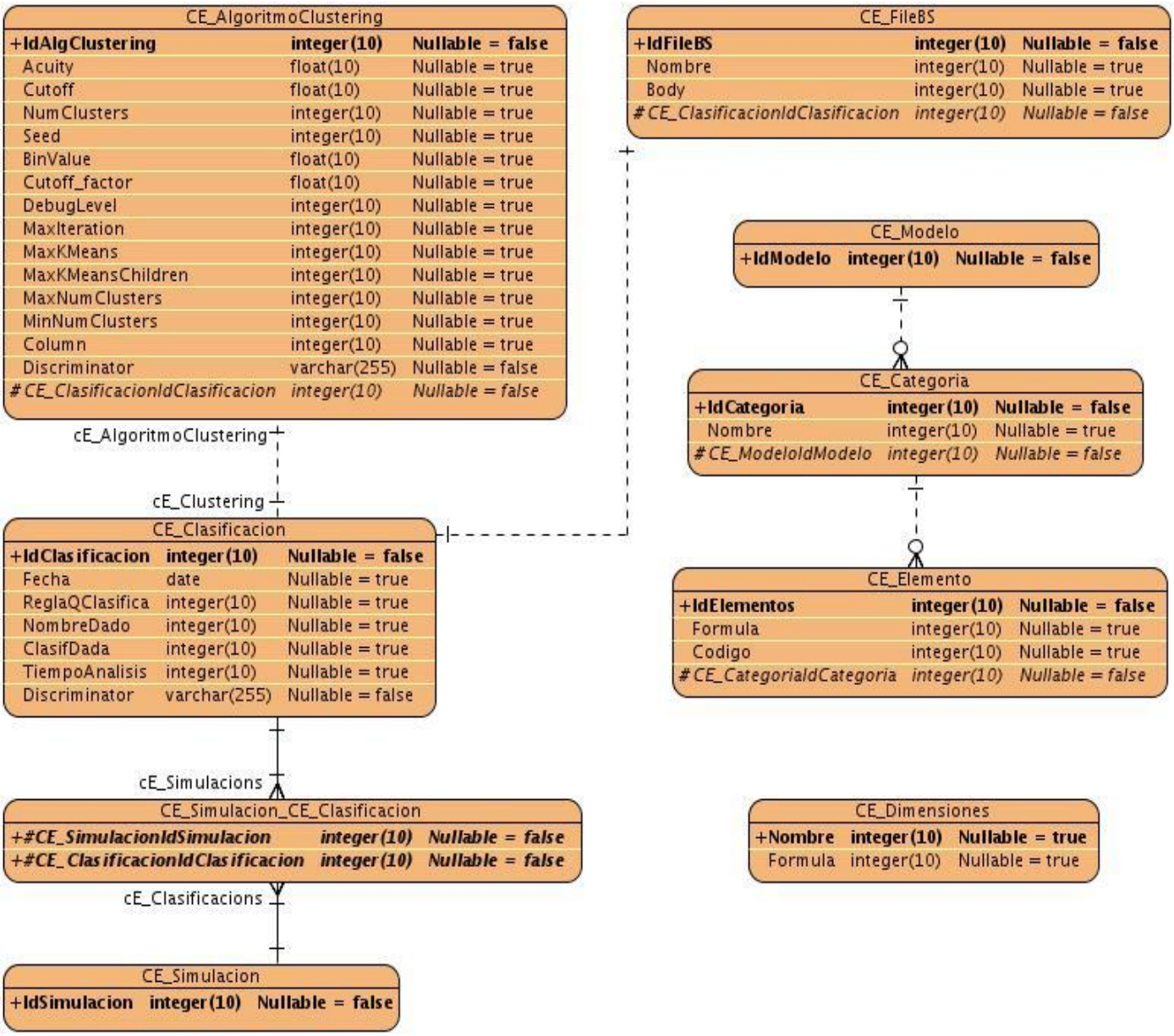


Figura 6: Modelo Entidad Relación

2.9- Descripción de las tablas

La tabla *Clasificación* almacena los datos generales de la clasificación de una simulación. Como llave primaria de la tabla, se encuentra el identificador de la clasificación, el que se encarga de identificar

cada clasificación de una simulación, además del identificador de la simulación como llave foránea y la fecha como un dato más de la clasificación. Cada simulación puede ser clasificada de diferentes maneras: *Manual*, mediante *Reglas*, utilizando la herramienta *Weka* o mediante *Clustering*.

La tabla *Manual* contiene los datos específicos de una clasificación de ese tipo. La clasificación manual está compuesta por el identificador de la clasificación el cual constituye su llave primaria y por un nombre que se le da a dicha clasificación. La tabla *ClasificacionWeka* está compuesta por el identificador de la clasificación, el cual constituye su llave primaria además de la clasificación que le da el programa Weka a las simulaciones. La tabla *Reglas* contiene los datos específicos de una clasificación de ese tipo. La clasificación por reglas está compuesta por el identificador de la clasificación el cual constituye su llave primaria y por las reglas que clasifican a las simulaciones. En la *tabla Clustering* existe un identificador de la clasificación que es su llave primaria, un tiempo de análisis y el algoritmo empleado para la clasificación a las simulaciones. Para realizar una clasificación por Clustering es necesario un algoritmo de *Clustering*, que puede ser *CobWeb, SimpleKMeans o XMeans*. En la tabla *AlgoritmoClustering* tiene un identificador del Algoritmo de Clustering.

La tabla *FileBS* contiene como llave principal el identificador del fichero del modelo generado, a partir de la clasificación a las simulaciones. Además tiene un nombre y un body de tipo BLOB, donde se almacena toda la información del modelo generado.

Las *tablas CobWeb, SimpleKMeans y XMeans* contienen como llave principal el identificador del algoritmo de *Clustering,* además de los datos específicos que identifican al algoritmo.
La tabla *Categoría* tiene como llave primaria el identificador de la Categoría, además del nombre. La categoría contiene *Elementos*, esta tabla tiene un identificador de elemento, una fórmula y un código. La tabla *Dimensiones* contiene un nombre como identificador y una fórmula.

2.10-Conclusiones

En este capítulo se realizó un análisis de la situación problémica, demostrándose la necesidad de incorporar nuevas funcionalidades a la Base de Datos ya existente, para lograr una mejor gestión de la información, relacionada con el proceso de análisis a las simulaciones y el Editor de Ecuaciones, de los que se realizó una extracción de los requisitos del sistema a implementar. Además de una descripción de las entidades según el diagrama de clases persistentes, que trajo como resultado el diseño del modelo entidad-relación y el modelo físicos de la Base de Datos.

CAPITULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1- Introducción

En el presente capítulo se explicarán los temas relacionados con la implementación y prueba a la Base de Datos para validar su correcto funcionamiento. Se describe de forma general las principales consultas desarrolladas que responden a las necesidades del Modulo de Análisis y el Editor de Ecuaciones.

3.2- Diagrama de Despliegue

El diagrama de despliegue es un diagrama de objetos que describe la distribución física del sistema (BioSys), en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Este diagrama se utiliza como entrada fundamental en las actividades de diseño e implementación, debido a la influencia que tiene en la distribución del sistema y se define a través de nodos y sus interrelaciones.

Los nodos son objetos físicos en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Estos son usados para representar los servidores y PC clientes.

Las interrelaciones están generalmente asociadas al protocolo de comunicación existente entre los nodos.

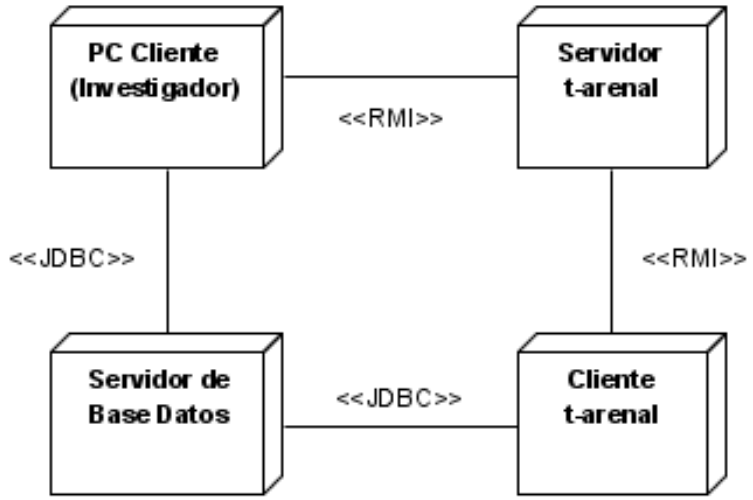


Figura 7: Vista de Despliegue del sistema BioSyS

Nodo PC Cliente: Representa la máquina cliente asignada al investigador, que en este caso es el

cliente del sistema, permite gestionar las peticiones del cliente, estará conectada a un servidor de base de datos por medio de JDBC y al T-arenal por medio de RMI.

Nodo T-arenal Server: Representa un servidor de objetos o Servidor T-arenal, recibe las peticiones del cliente, las divide en estaciones de trabajo y se las envía a sus máquinas clientes por RMI. Por ser una plataforma de cálculo distribuido permite una mayor agilidad en los resultados.

Nodo T-arenal Cliente: Representa las máquinas que están a disposición del T-arenal (servidor de objetos), se representó con solo una PC pero en realidad no se sabe la cantidad de PC Clientes que se van a utilizar, estas son las encargadas de realizar el trabajo enviado por él y para ello necesitan tener instalada de manera opcional la herramienta de cálculo MatLab, en caso de no estar el MatLab, se usa Java. Luego de realizar el trabajo lo envía particionado al servidor de base de datos por JDBC.

Nodo Servidor de BD: Representa un servidor de base de datos utilizado para el almacenamiento de los datos de la aplicación y para lograr la conexión del sistema con la base de datos se utiliza JDBC como protocolo de comunicación asociado al ORM Hibernate. Se encarga también de organizar la información enviada por la PC clientes del T-arenal.

Nodo PC Cliente: Representa la máquina cliente asignada al investigador, que en este caso es el cliente del sistema, permite gestionar las peticiones del cliente, estará conectada a un servidor de base de datos por medio de JDBC y al T-arenal por medio de RMI.

Protocolo de comunicación JDBC: Es una especificación de un conjunto de clases y métodos de operación que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. La aplicación de Java debe tener acceso a un driver JDBC adecuado. Este driver es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real.

Protocolo de comunicación RMI: Esta es la tecnología asociada al lenguaje de programación de Java. Es un enfoque Java puro en el que solo los programas escritos en ese lenguaje se pueden comunicar con un objeto RMI distribuido. [25]

3.3-Diagrama de Componentes

Un diagrama de componentes se utiliza para modelar la vista estática de un sistema. Los elementos de

modelado dentro de un diagrama de componentes, serán componentes y paquetes, donde se muestran las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables. Los componentes de software tienen tipos, que indica si son útiles en tiempo de compilación, enlace o ejecución.

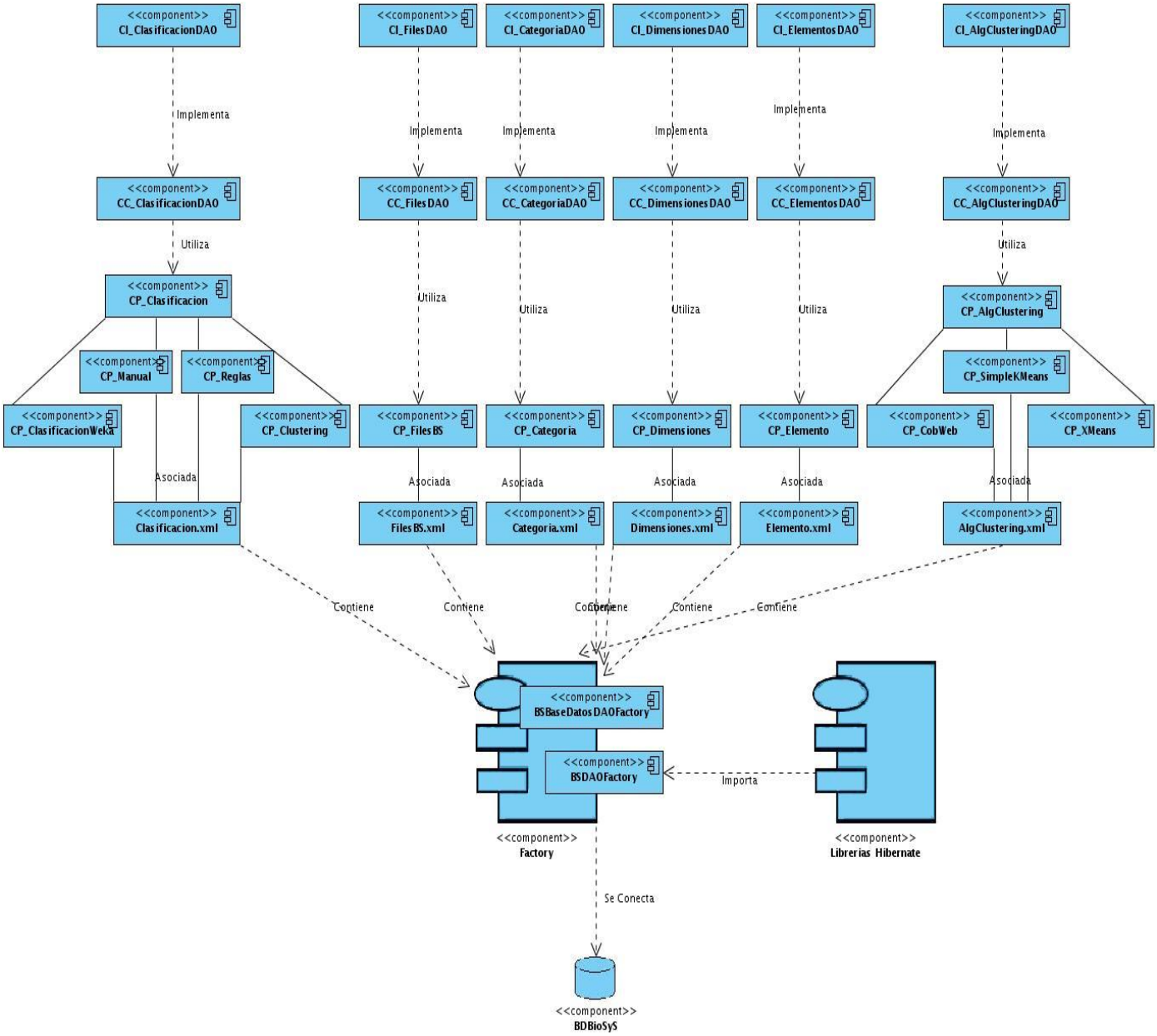


Figura 8: Diagrama de Componentes

3.4- Validación teórica del diseño

La validación de un modelo se puede definir como la demostración de su exactitud (ausencia de error sistemático y aleatorio) para una aplicación concreta.

Durante el proceso de desarrollo de una Base de Datos, es importante determinar si los datos están actualizados, completos y suficientes. Es frecuente, descubrir inconsistencias, falta de información estratégica, datos obsoletos o incompletos, duplicidad, bajo rendimiento y falta de normalización.

Las mediciones que se realizan sobre la calidad pueden tener distintos objetivos, dependiendo de la situación en la que se encuentre el proyecto. Los modelos de análisis y diseño no se pueden probar en el sentido convencional, ya que no pueden ejecutarse. Sin embargo, se pueden utilizar revisiones técnicas formales y otras prácticas, para examinarlos.

3.4.1- Integridad de datos

Integridad de datos se refiere al estado de corrección y completitud de los datos ingresados en una Base de Datos. Cuando el contenido de una Base de Datos se modifica mediante sentencias INSERT, DELETE o UPDATE, la integridad de los datos puede perderse de muchas maneras. Pueden añadirse datos no válidos a la base de datos, pueden modificarse datos existentes tomando un valor incorrecto, los cambios en la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de energía, los cambios pueden ser aplicados parcialmente. Una de las funciones importantes de un sistema manejador de bases de datos (DBMS) relacional es preservar la integridad de sus datos almacenados en la mayor medida posible.

La integridad de los datos se encarga de analizar los siguientes elementos:

Integridad de dominio: La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones CHECK, o el intervalo de valores posibles mediante restricciones FOREIGN KEY, restricciones CHECK, definiciones DEFAULT, definiciones NOT NULL y reglas.

Integridad de entidad: La integridad de entidad define una fila como entidad única para una tabla

determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY. Todos los atributos primarios de las tablas nunca serán nulos.

Integridad referencial: La integridad referencial protege las relaciones definidas entre las tablas, cuando se crean o se eliminan filas. La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la Base de Datos.

3.4.2-Normalización de la Base de Datos

El proceso de normalización de Bases de Datos es un proceso de descomposición que aplica una serie de reglas a las relaciones obtenidas, tras el paso del modelo entidad-relación al modelo relacional. La normalización de Bases de Datos relacionales toma un esquema relacional y le aplica un conjunto de técnicas para producir un nuevo esquema, que representa la misma información pero contiene menos redundancias y evita posibles anomalías en las inserciones, actualizaciones y borrados.

El proceso de normalización en esencia consiste en la comprobación del esquema original, garantizando si pertenece o no a una cierta forma normal, analizando las dependencias funcionales en cada paso. La Primera Forma Normal (1FN) define que los dominios de los atributos deben incluir solo valores atómicos, es decir que no pueden ser multivaluados, ni compuestos. Un esquema está en 2FN si está en 1FN y lo atributos no primos dependen totalmente de la clave o llave del esquema; un atributo primo es aquel que forma parte de alguna clave del esquema. Una relación está en 3FN si y sólo si, está en 2FN y además, cada atributo que no está incluido en la clave primaria no depende transitivamente de ésta.

La Base de Datos para BioSyS 2.0 se encuentra en 3FN, primeramente porque todas sus tablas están en 2NF y no existen dependencias transitivas entre los atributos de un mismo esquema.

A pesar de los innumerables beneficios que brinda la normalización, la realización de consultas en ocasiones puede requerir la combinación de varias tablas para unir la información, lo que provoca que mientras mayor sea el número de tabas a combinar, el tiempo de ejecución de la consulta aumenta considerablemente. Por este motivo, el uso de una Base de Datos normalizada, no es siempre la mejor alternativa. En este caso no fue necesario llegar a un grado de desnormalización aceptable de la Base de Datos, ya que la información que se debe brindar no requiere de la combinación de muchas relaciones.

3.4.3- Análisis de la Redundancia de la Información

Un modelo de datos completamente normalizado no contiene redundancia. Si algo de redundancia se introduce, entonces existe la posibilidad de inconsistencias en los datos. Para la nueva versión de BioSyS se obtuvo un diseño normalizado y optimizado de la Base de Datos, con el fin de lograr una redundancia nula, además de impedir al máximo las dependencias de los datos para que el proceso de actualización de la base de datos sea fácil y eficiente.

3.4.4- Análisis de la seguridad de la Base de Datos

MySQL Administrador es un programa muy útil para administrar visualmente y de manera sencilla, servidores de Bases de Datos MySQL. Permite realizar tareas administrativas sobre servidores tales como: la configuración de las opciones de inicio de los servidores, monitorización de conexiones al servidor, administración de usuarios, entre otros privilegios necesarios para la seguridad de la Base de Datos.

Luego de la instalación se gestionan los privilegios y permisos de usuarios mediante la opción de “Administración de usuario”.

Las propiedades de usuarios se dividen en tres partes:

Información de usuario: con los datos de entrada (nombre de usuario y contraseña de acceso) y otros datos personales del usuario.

Privilegios de Esquema: Se seleccionan los permisos de este usuario para cada Base de Datos de nuestro sistema. Mediante esta opción se puede seleccionar una Base de Datos y acceder a una lista con todos los privilegios posibles, para permitir o denegar el acceso. Inicialmente, para un usuario nuevo, todos los permisos están denegados, así que tendremos que seleccionar los que deseamos otorgar. Para ello simplemente seleccionamos entre los permisos disponibles y los pasamos a

permisos asignados.

Recursos: Son los recursos disponibles para ese usuario.

Para una mayor seguridad se utilizaron los comandos GRANT y REVOKE que permiten a los administradores de sistemas crear cuentas de usuario MySQL y restringir así el acceso a la información almacenada.

Los permisos pueden darse en varios niveles:
Nivel global: Los permisos globales se aplican a todas las Bases de Datos de un servidor dado. Estos permisos se almacenan en la tabla mysql.user. GRANT ALL ON *.* y REVOKE ALL ON *.* otorgan y quitan sólo permisos globales.

Nivel de Base de Datos: Los permisos de Base de Datos se aplican a todos los objetos en una Base de Datos dada. Estos permisos se almacenan en las tablas mysql.db y mysql.host. GRANT ALL ON *db_name*.* y REVOKE ALL ON *db_name*.* otorgan y quitan sólo permisos de Bases de Datos específicas.

Nivel de tabla: Los permisos de tabla se aplican a todas las columnas en una tabla dada. Estos permisos se almacenan en la tabla mysql.tables_priv. GRANT ALL ON *db_name.tbl_name* y REVOKE ALL ON *db_name.tbl_name* otorgan y quian permisos sólo de tabla.

Nivel de columna: Los permisos de columna se aplican a columnas en una tabla dada. Estos permisos se almacenan en la tabla mysql.columns_priv. Usando REVOKE, debe especificar las mismas columnas que se otorgaron los permisos.

Nivel de rutina: Los permisos CREATE ROUTINE, ALTER ROUTINE, EXECUTE, y GRANT se aplican a rutinas almacenadas. Pueden darse a nivele global y de Base de Datos. Además, excepto para CREATE ROUTINE, estos permisos pueden darse en nivel de rutinas para rutinas individuales y se almacenan en la tabla mysql.procs_priv.

Se crearon diferentes cuentas de usuario MySQL para restringir el acceso a la información almacenada. Las cuentas creadas son Módulo de Estimación, Módulo de Simulación, Módulo Editor de

Ecuaciones, Módulo Modelación y Módulo Análisis, donde a cada una se le asignaron o denegaron privilegios mediante los comandos GRANT y REVOKE en dependencia de la información con la que tenga que interactuar, en el desarrollo de su trabajo. Los privilegios asignados o denegados fueron a nivel de Base de Datos y a nivel de tablas.

3.4.5-Trazabilidad de las acciones

La trazabilidad es la cualidad que permite que todas las acciones realizadas sobre un sistema de tecnología de la información sean asociadas a un individuo o entidad.

Para llevar un control de las acciones realizadas por los usuarios que interactúan con la Base de Datos se creó una tabla log _Table, en la cual se almacenará el historial de las actividades de los usuarios al conectarse a la Base de Datos, es decir, recogerá los datos necesarios para registrar las conexiones (identificador de la conexión, usuario, la tabla a la que accede, operación que realiza y la fecha), la operación que se realizan pueden ser de consulta, inserción, actualización o eliminación.

3.5-Validación Funcional

La validación funcional es la verificación del modelo frente a observaciones obtenidas de manera independiente. La evaluación ideal consiste en obtener los datos pertinentes del mundo real y realizar una comparación estadística de los resultados simulados y las observaciones.

3.5.1-Validación y Prueba

Las pruebas que validan el rendimiento de una Base de Datos son las que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo, demostrándose si el sistema cumple o no los requerimientos deseados.

3.5.1.1-Pruebas de Volumen

Pruebas que se hacen para analizar el comportamiento de la Base de Datos, con volúmenes de datos almacenados similares a los esperados en la explotación real del sistema, poblándose cada tabla con cantidades de registros de entre un 110% y un 150% de los volúmenes esperados. Estas pruebas pueden detectar errores tanto del diseño como de la implementación del sistema como tal; errores tales como: falta de espacio en disco, conflictos provocados por las llaves autogeneradas, falta de optimización de las consultas, mal diseño de las mismas, entre otros.

Para el llenado de la Base de Datos se utilizó la herramienta EMS Data Generador 2005 para MySQL,

la cual permite la generación de datos para una o varias tablas a la vez, donde se gestionan por cada tabla la generación de sus datos de forma independiente.

Con el uso de la herramienta EMS Data Generador 2005 para MySQL se generaron para las pruebas a la Base de Datos, volúmenes de 100 valores random, generando 2 300 valores para las 23 tablas en 0:01:17 segundos sin errores. Luego se insertaron en cada tabla 10 000 valores random, generando 230 000 valores en 0:10:49 segundos. Por último se insertaron 99 204 valores random en 2:03:45 segundos con 796 errores.

Tabla 14: Información de generación de datos.

Datos /Tablas	Datos Insertados	Tiempo	Errores
100	2 300	0:01:17	
10 000	230 000	0:10:49	
100 000	99 204	2:03:45	796

Las gráficas muestran los valores de las pruebas de volumen, para una mejor visión del impacto de la simultaneidad de datos soportados. La primera ilustración muestra como se comportaron las métricas de datos insertados respecto al tiempo; y la segunda ilustra el margen de errores obtenidos en cada carga de datos, el que puede ser gradual en dependencia de la cantidad de datos a procesar.

3.5.1.2-Pruebas de Carga

Las pruebas de carga realizadas a una Base de Datos no es más que someter a la misma a un régimen de carga de trabajo (habitualmente por simulación de concurrencia) similar al esperado en su explotación real.

El objetivo primordial de las pruebas de carga es validar que el comportamiento y los tiempos de respuesta experimentados, bajo ciertas condiciones (Número de usuarios, número de transacciones por unidad de tiempo) satisfacen adecuadamente los requerimientos especificados.

Mediante la utilización de la herramienta Apache-JMeter se valida el comportamiento y se verifican los tiempos de respuesta de la Base de Datos de BioSyS para la versión 2.0, tomando como punto de partida un requisito crítico: Insertar una clasificación por Clustering. Insertando una clasificación por

Clustering primeramente se debe insertar el Algoritmo de Clustering que realiza dicha clasificación, posteriormente se busca el grupo de simulaciones que se quieren clasificar y se inserta la clasificación tanto en la tabla Clasificación como en la específica que almacena los tipos de clasificaciones por Clustering.

Las pruebas de carga para la Base de Datos se van a realizar mediante la simulación de concurrencia, conectándose un número determinado de usuarios a la Base de Datos. Esto es posible gracias a la facilidad que brinda la herramienta JMeter para realizar peticiones sobre un servidor de Base de Datos y después repetirlas, simulando la conexión un número de usuarios tantas veces como se desee.

Se realizarán tres pruebas con el objetivo de verificar el correcto funcionamiento de la Base de Datos.
La prueba va a comenzar con un nivel de concurrencia de 10 usuarios, con un contador de bucle igual a 3. Para la ejecución de la consulta a probar se necesitan 4 consultas ó peticiones dada la cantidad de operaciones que implica una inserción de este tipo. Estos datos arrojan que 10 usuarios, por 4 peticiones, por 3 repeticiones dan un resultado del 120 solicitudes a la Base de Datos. Se realiza la misma operación con 15 y 20 usuarios.

Consultas a realizar para el desarrollo de la prueba son:
1. Seleccionar Simulación a clasificar.
2. Insertar Algoritmo Clustering.
3. Insertar Clasificación.
4. Insertar Clasificación por Clustering.

Se realizará la medición para obtener el tiempo promedio que demora la Base de Datos en dar respuesta a cada query.
La prueba arroja los siguientes resultados:

Tabla 15: Prueba de Carga.

Cantidad Usuarios	1	2	3	4	Tiempo Promedio(ms)
10	38	44	56	50	47
15	41	58	65	58	55.5
20	45	62	70	80	64.25

Se observa que al simular la conexión de 20 usuarios concurrentes a la Base de Datos el uso del CPU se mantuvo constante a un 98%, lo que demuestra que la Base de Datos y los recursos de hardware del servidor soportan esa cantidad de usuarios conectados concurrentemente.

El servidor de Base de Datos llegará al límite cuando el consumo de alguno de los recursos siguientes llegue al máximo:

- CPU.
- Memoria RAM.
- Red.

La gráfica muestra el tiempo promedio de realización de cada query según la cantidad de usuarios que realizan las peticiones concurrentemente.

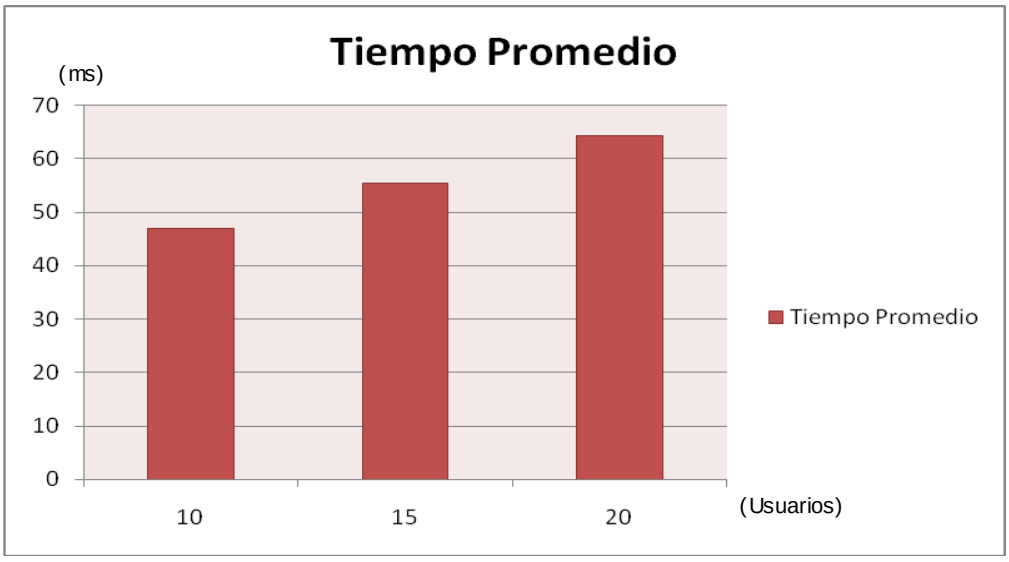


Figura 9: Tiempo promedio de la prueba.

Con esta prueba, ejecutando una de las operaciones más críticas que se realizan sobre la Base de Datos, 120, 180 y 240 veces concurrentemente según la cantidad de usuarios, podemos ver que el tiempo promedio de respuesta aumenta debido al incremento de las peticiones que debe de atender el servidor de Base de Datos.

3.5.1.3-Pruebas de Stress

Las pruebas de Stress realizadas a una Base de Datos no es más que someter a la misma a un régimen de carga de trabajo (habitualmente por simulación de concurrencia) muy superior al esperado en su explotación real. Pudiera verse como un caso particular de prueba de carga.

Tanto las pruebas de carga como las de stress nos sirven para identificar errores que atentan contra el buen desempeño del sistema en general.

Errores como:

- Hardware Insuficiente (espacio en disco, consumo de RAM)
- Necesidad de ajuste del servidor de BD
- Respuesta lenta
- “Time Outs” en diferentes lugares

Mediante la herramienta Apache JMeter por simulación de concurrencia se realiza la prueba de Stress a la Base de Datos. Se observa que al simular la conexión de 30 usuarios concurrentes a la Base de Datos, el uso del CPU realizaba picos hasta 100%, lo que quiere decir que al servidor no poder dar respuesta a diferentes solicitudes por el consumo tan elevado que esto lleva, la solicitud se queda sin respuesta, por lo que esa solicitud no consume recursos y realiza los picos en el CPU, además demuestra que los recursos de hardware del servidor no soportan esa cantidad de usuarios conectados concurrentemente.

3.5.2-Principales consultas

Con el objetivo de probar el funcionamiento correcto de la Base de Datos, se realizaron consultas, teniendo en cuenta los resultados y el tiempo de ejecución de las mismas.

Las consultas generadas para la realización de pruebas, se escogen de acuerdo a las operaciones que se esperan sean las más utilizadas una vez implantada la Base de Datos. Algunas de estas consultas que responden a determinados requisitos funcionales son:

1. ”Obtener las clasificaciones realizadas a una simulación dado su identificador (identificador=1)”

SELECT * FROM clasificacion INNER JOIN sim_clasif ON
(clasificacion.IDClasificacion= sim_clasif.IDClasificacion) INNER JOIN simulacion ON
(sim_clasif.id_Simulacion= simulacion.id_Simulacion) WHERE simulacion.Id_Simulacion = 1;
Esta consulta se ejecuta en un tiempo de 867 ms.

2. “Obtener las clasificaciones por clustering realizadas con el algoritmo Xmeans.”

SELECT * FROM clasificacion INNER JOIN clustering ON
(clasificacion.IDClasificacion=clustering.IDClasificacion) INNER JOIN alg_xmeans ON
(clustering.idAlgClustering=alg_xmeans.idAlgClustering);
Esta consulta se ejecuta en un tiempo de 384 ms.

3. “Obtener las categorias utilizadas en un modelo matemático (identificador=1).”

SELECT * FROM categoria INNER JOIN modelo_matematico ON
(categoria.Id_Modelo=modelo_matematico.Id_Modelo) WHERE modelo_matematico.Id_Modelo=1;
Esta consulta se ejecuta en un tiempo de 513 ms.

4. “Modificar el nombre de la categoría cuyo identificador del modelo sea 1”

UPDATE categoria SET categoria.Nombre='Dayana' WHERE categoria.Id_Categoria=1;
Esta consulta se ejecuta en un tiempo de 185 ms.

5. “Obtener los elementos de un modelo matemático cuyo identificador del dicho modelo sea 1.”

SELECT * FROM elemento INNER JOIN categoria ON
(elemento.categoria_Id_Categoria=categoria.Id_Categoria) INNER JOIN modelo_matematico
ON(categoria.Id_Modelo=modelo_matematico.Id_Modelo) WHERE modelo_matematico.Id_Modelo=1;
Esta consulta se ejecuta en un tiempo de 271 ms.

6.”Optener un listado de File perteneciente a clasificaciones realizadas manualmente”

SELECT * FROM files INNER JOIN manual ON (files.IDClasificacion = manual.IDClasificacion);

Esta consulta se ejecuta en un tiempo de 96 ms.

7.“Obtener las clasificaciones realizadas en la fecha del 28 de mayo del 2009”

SELECT * FROM clasificacion where clasificacion.Fecha = '2009-05-28';
Esta consulta se ejecuta en un tiempo de 749 ms.

3.5.3- Análisis de optimización de consultas

Las consultas son solicitudes que se realizan a la Base de Datos para almacenar, visualizar y actualizar los datos. Una consulta puede expresarse de varias maneras, donde cada una propone una forma diferente de hallar el resultado, de ahí que unas sean más óptimas que otras.

El acceso a los datos en la Base de Datos de BioSyS para la versión 2.0 se implementa mediante objetos, Hibernate le permite a la aplicación manipular los datos mediante estos objetos, con todas las características de la programación orientada a objeto. Esta una herramienta de mapeo objeto/relacional para ambientes Java, no solo se encarga del mapeo de clases Java a tablas de la base de datos (y de regreso), sino que también maneja los queries y recuperación de datos, lo que puede reducir de forma significativa el tiempo de desarrollo que de otra forma gastaríamos manejando los datos de forma manual con SQL, encargándose de esta forma de alrededor del 95% de las tareas comunes relacionadas con la persistencia de datos, manejando todos los problemas relativos con la base de datos particular con la que se está trabajando. Entonces, si cambiamos el manejador de Base de Datos no será necesario que modifiquemos todo el SQL que ya que se tenía para adaptarse al SQL que maneja la nueva Base de Datos. Solo será necesario modificar una línea en un archivo de configuración de Hibernate, y este se encargará del resto. Hibernate convertirá los datos entre los tipos utilizados por Java y los y definidos por SQL y generará las sentencias SQL posibilitando así el manejo de los datos que resultan de la ejecución de dichas sentencias, manteniendo así la portabilidad entre todos los SGBD.

La implementación de las consultas básicas de la Base de Datos son implementadas en Hibernate mediante métodos desarrollados para realizar cada una de esas operaciones (INSERT, DELETE, UPDATE). Se propone realizar un análisis de las vías más óptimas de acceso a los datos para obtener

mejores tiempos de respuestas a las solicitudes.

Un ejemplo claro es cuando se quiere obtener los Elementos de una Categoría, se puede utilizar el método getElementos(int idCategoria) implementado en Hibernate. En este caso se realiza una consulta simple sin utilizar SQL, para conseguir la abstracción de la base de datos. Para las selecciones más complejas de objetos en la Base de Datos, se necesitan equivalentes a las sentencias WHERE, ORDER BY, GROUP BY y demás sentencias SQL, las que serán ejecutadas a partir de métodos contenidos en Hibernate para realizar esta operación.

3.6-Conclusiones

En este capítulo se representa el diagrama de despliegue referente al sistema BioSyS, el cual muestra la distribución física del sistema como punto de partida para la implementación y además se diseñó el diagrama de componentes, que describe la arquitectura lógica de la Base de Datos. Con el uso de las herramientas NetBeans e Hibernate se implementó la capa de acceso a datos, permitiendo de esta forma el acceso a la información almacenada. Se desarrollaron diferentes pruebas para validar el correcto funcionamiento de la Base de Datos para BioSyS 2.0.

CONCLUSIONES

El presente trabajo a lo largo de sus 3 capítulos mostró la importancia del desarrollo de la versión 2.0 de la Base de Datos de BioSyS para el avance en las investigaciones de sistemas biológicos, ya que posibilita la gestión de la información generada en los procesos de análisis de las simulaciones de sistemas biológicos.

- Se creó la nueva versión de la Base de Datos que incluye lo referente al módulo de análisis.
- Se implementó la capa de acceso a datos.
- Se realizó un grupo de pruebas que validaron el correcto funcionamiento de la Base de Datos.

RECOMENDACIONES

Aplicarle técnicas de inteligencia artificial a la Base de Datos para obtener patrones de comportamiento que garanticen mejores estudios.

REFERENCIAS BIBLIOGRÁFICAS

1. **B.Tidor., B.Tadmor and.** Interdisciplinary research and education at the biologic engineering computer science interface: a perspective (reprinted article).Biosilico, Vol 10(No.17). September 2005.

2. **Lemus, Lic. Noel Moreno.** BioSyS: Software para la simulación y análisis de sistemas. La Habana: s.n., 2007.

3. **Un nuevo giro. La biología molecular recupera la biología de sistemas** [En línea] [Citado el: 21 de noviembre de 2008.] http://www.institutoroche.es/editorial2.php?op=biotecnologia&id=29.

4. **Biología de sistemas.** [En línea] 2004. [Citado el: 21 de noviembre de 2008.] http://bvs.isciii.es/bib-gen/Actividades/curso_virtual/Ftes_informacion/Biologiasistemas.htm.

5. **Un diagrama derivado de la biología de sistemas.** [En línea] [Citado el: 21 de noviembre de 2008.] http://www.proz.com/kudoz/english_to_spanish/biology_tech_chemmicro_/2130827-a_systems_biology_derived_picture.html.

6. **Máster en Biología de Sistemas.** [En línea] [Citado el: 21 de noviembre de 2008.] http://www.uvic.cat/eps/estudis/biologia_sistemes/es/inici.html

7. **Simulación.** [En línea] 2007 [Citado el: 21 de noviembre de 2008.] http://www.simulart.cl/simulacion_web.htm

8. **Ventajas / Desventajas de la Simulación.** [En línea] [Citado el: 21 de noviembre de 2008.] http://docencia.50webs.com/simula03.htm

9. **Balduino, Ricardo.** [En línea] [Citado el: 21 de noviembre de 2008.] http://www.eclipse.org/epf/general/OpenUP.pdf

10. **Salinas Caro, Patricio y Histchfeld K, Nancy.** Tutorial de UML [http://www.dcc.uchile.cl/~psalinas/uml/]

11. **Ing. Irais Mogena Tamayo.** Base de Datos del Simulador de Sistemas Biológicos. BioSyS. La Habana: s.n., 2008.

12. **Sistemas gestores de bases de datos** [En línea] 31/7/2007[Citado el: 22 de noviembre de 2008.] http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html

13. **María Isabel García Arenas**. Curso XML 1ª Edición [En línea] [Citado el: 22 de noviembre de 2008.] http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml#12

14. **AulaClic.** Curso de SQL. [En línea] [Citado el: 22 de noviembre de 2008.] http://www.aulaclic.es/sql/f_sql.htm

15. **Investigación sobre MySQL.** [En línea] [Citado el: 21 de noviembre 2008] http://www.salnet.com.ar/inv_mysql/pag01_intro.htm

16. **¿Que es NetBeans?.** [En línea] [Citado el: 22 de noviembre 2008] http://www.netbeans.org/index_es.html

17. **Suárez González, Héctor.** Manual de Hibernate. [En línea] 2003 [Citado el: 22 de noviembre 2008] http://www.javahispano.org/contenidos/es/manual_hibernate/

18. **Caracteristicas del MySQL Query Browser.** [En línea] Febrero 2005 [Citado el: 22 de noviembre 2008] http://www.desarrolloweb.com/articulos/1832.php

19. **Caracteristicas del MySQL Administrator.** [En línea] Enero 2005 [Citado el: 22 de noviembre 2008] http://www.desarrolloweb.com/articulos/1798.php

20. **Generador de Datos EMS 2005 para MySQL** (EMS Data Generator 2005 for MySQL). . [En línea] Enero 2005 [Citado el: 15 de marzo 2009] http://www.freedownloadmanager.org/es/downloads/Generador_de_Datos_de_SME_2005_para_MySQL_37201_p/

21. **JMeter** [En línea] Octubre 2005 [Citado el: 20 de abril 2009] http://www.osmosislatina.com/jmeter/basico.htm

22. **Características de Ubuntu.** [En línea] [Citado el: 24 de noviembre 2008] http://www.guadalinex.org/guadapedia/index.php/Ubuntu

23. **Jose Luis Mesa, Andres Ricardo Torres, Claudia Patricia Oviedo, Jennifer Andrea Tenorio.** [En línea] [Citado el: 24 de noviembre 2008] http://eisc.univalle.edu.co/materias_Soft/Material_Desarrollo ware/expoDAO.pdf

24. **Expediente de Proyecto**. Documento de especificación de Requisitos: Modulo de Análisis y Editor de Ecuaciones.

25. **Expediente de Proyecto.** Documento de Arquitectura: Vista de despliegue.

BIBLIOGRAFÍA

B.Tidor. B.Tadmor and Interdisciplinary research and education at the biologic engineering computer science interface: a perspective (reprinted article).Biosilico, Vol 10(No.17). September 2005.

Lemus, Lic. Noel Moreno. BioSyS: Software para la simulación y análisis de sistemas. La Habana: s.n., 2007.

Un nuevo giro. La biología molecular recupera la biología de sistemas [En línea] [Citado el: 21 de noviembre de 2008.] http://www.institutoroche.es/editorial2.php?op=biotecnologia&id=29.

Biología de sistemas. [En línea] 2004. [Citado el: 21 de noviembre de 2008.] http://bvs.isciii.es/bib-gen/Actividades/curso_virtual/Ftes_informacion/Biologiasistemas.htm.

Un diagrama derivado de la biología de sistemas. [En línea] [Citado el: 21 de noviembre de 2008.] http://www.proz.com/kudoz/english_to_spanish/biology_tech_chemmicro_/2130827-a_systems_biology_derived_picture.html.

Máster en Biología de Sistemas. [En línea] [Citado el: 21 de noviembre de 2008.] http://www.uvic.cat/eps/estudis/biologia_sistemes/es/inici.html

Simulación. [En línea] 2007 [Citado el: 21 de noviembre de 2008.] http://www.simulart.cl/simulacion_web.htm

Ventajas / Desventajas de la Simulación. [En línea] [Citado el: 21 de noviembre de 2008.] http://docencia.50webs.com/simula03.htm

Balduino, Ricardo. [En línea] [Citado el: 21 de noviembre de 2008.] http://www.eclipse.org/epf/general/OpenUP.pdf

Salinas Caro, Patricio y Histchfeld K, Nancy. Tutorial de UML [http://www.dcc.uchile.cl/~psalinas/uml/]

Ing. Irais Mogena Tamayo. Base de Datos del Simulador de Sistemas Biológicos. BioSyS. La Habana: s.n., 2008.

Sistemas gestores de bases de datos [En línea] 31/7/2007[Citado el: 22 de noviembre de 2008.] http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html

María Isabel García Arenas. Curso XML 1ªEdición [En línea] [Citado el: 22 de noviembre de 2008.] http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml#12

AulaClic. Curso de SQL. [En línea] [Citado el: 22 de noviembre de 2008.] http://www.aulaclic.es/sql/f_sql.htm

Investigación sobre MySQL. [En línea] [Citado el: 21 de noviembre 2008] http://www.salnet.com.ar/inv_mysql/pag01_intro.htm

¿Que es NetBeans?. [En línea] [Citado el: 22 de noviembre 2008] http://www.netbeans.org/index_es.html

Suárez González, Héctor. Manual de Hibernate. [En línea] 2003 [Citado el: 22 de noviembre 2008] http://www.javahispano.org/contenidos/es/manual_hibernate/

Caracteristicas del MySQL Query Browser. [En línea] Febrero 2005 [Citado el: 22 de noviembre 2008] http://www.desarrolloweb.com/articulos/1832.php

Caracteristicas del MySQL Administrator. [En línea] Enero 2005 [Citado el: 22 de noviembre 2008] http://www.desarrolloweb.com/articulos/1798.php

Generador de Datos EMS 2005 para MySQL (EMS Data Generator 2005 for MySQL). . [En línea] Enero 2005 [Citado el: 15 de marzo 2009] http://www.freedownloadmanager.org/es/downloads/Generador_de_Datos_de_SME_2005_para_MySQL_37201_p/

JMeter [En línea] Octubre 2005 [Citado el: 20 de abril 2009] http://www.osmosislatina.com/jmeter/basico.htm

Características de Ubuntu. [En línea] [Citado el: 24 de noviembre 2008] http://www.guadalinex.org/guadapedia/index.php/Ubuntu

Jose Luis Mesa, Andres Ricardo Torres, Claudia Patricia Oviedo, Jennifer Andrea Tenorio. [En línea] [Citado el: 24 de noviembre 2008] http://eisc.univalle.edu.co/materias_Soft/Material_Desarrollo ware/expoDAO.pdf

Expediente de Proyecto. Documento de especificación de Requisitos: Modulo de Análisis y Editor de Ecuaciones.

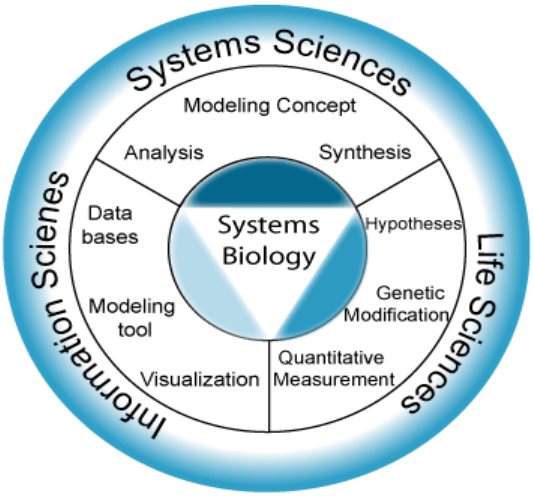
Expediente de Proyecto. Documento de Arquitectura: Vista de despliegue.

Ian Gilfillan. La Biblia MySQL. 1ª edición (30 de Junio del 2003).

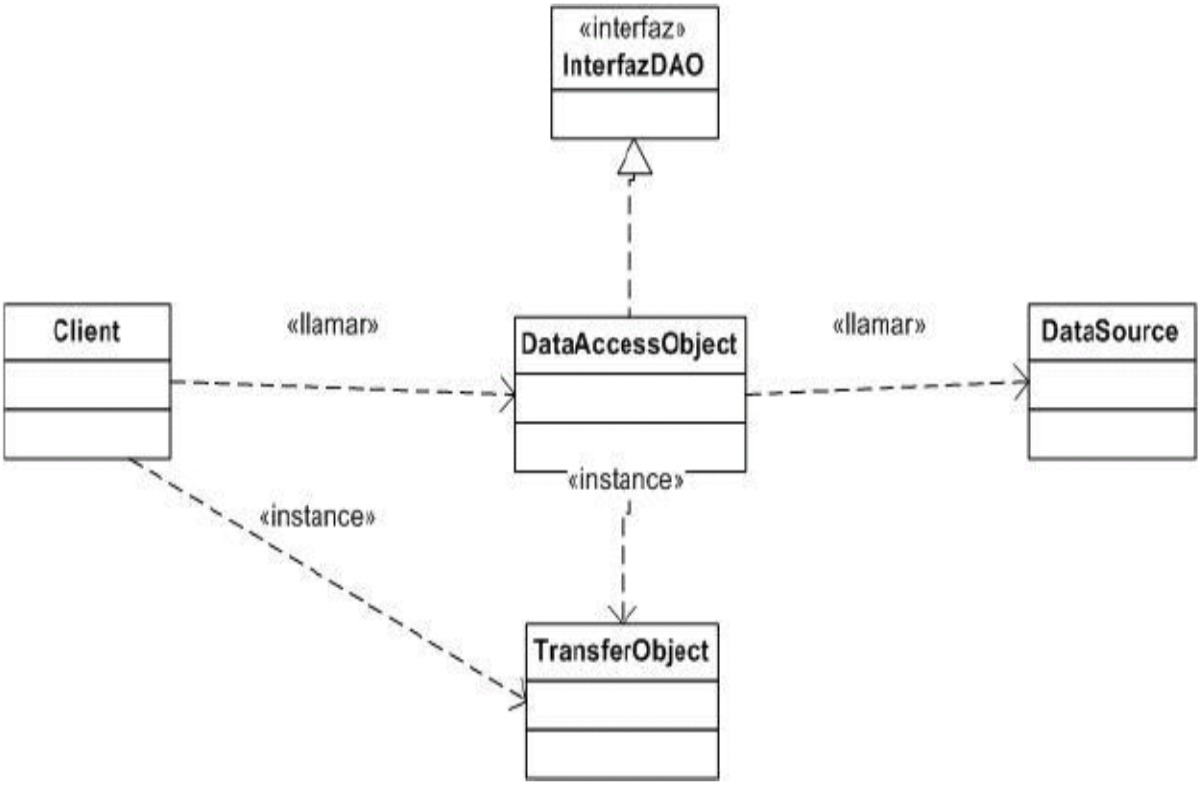
Cátedra de Base de Datos. Oracle-Considerando la introducción de redundancia (Desnormalización) [En línea] [Citado el: 15 de enero de 2008.] http://bdatos.wordpress.com/base-de-conocimiento/oracle-considerando-la-introduccion-de-redundancia-desnormalizacion/

PostGreSQL vs. MySQL. ¿Qué es MySQL?. [En línea] [Citado el: 15 de enero de 2008.].http://www.netpecos.org/docs/mysql_postgres/x57.html

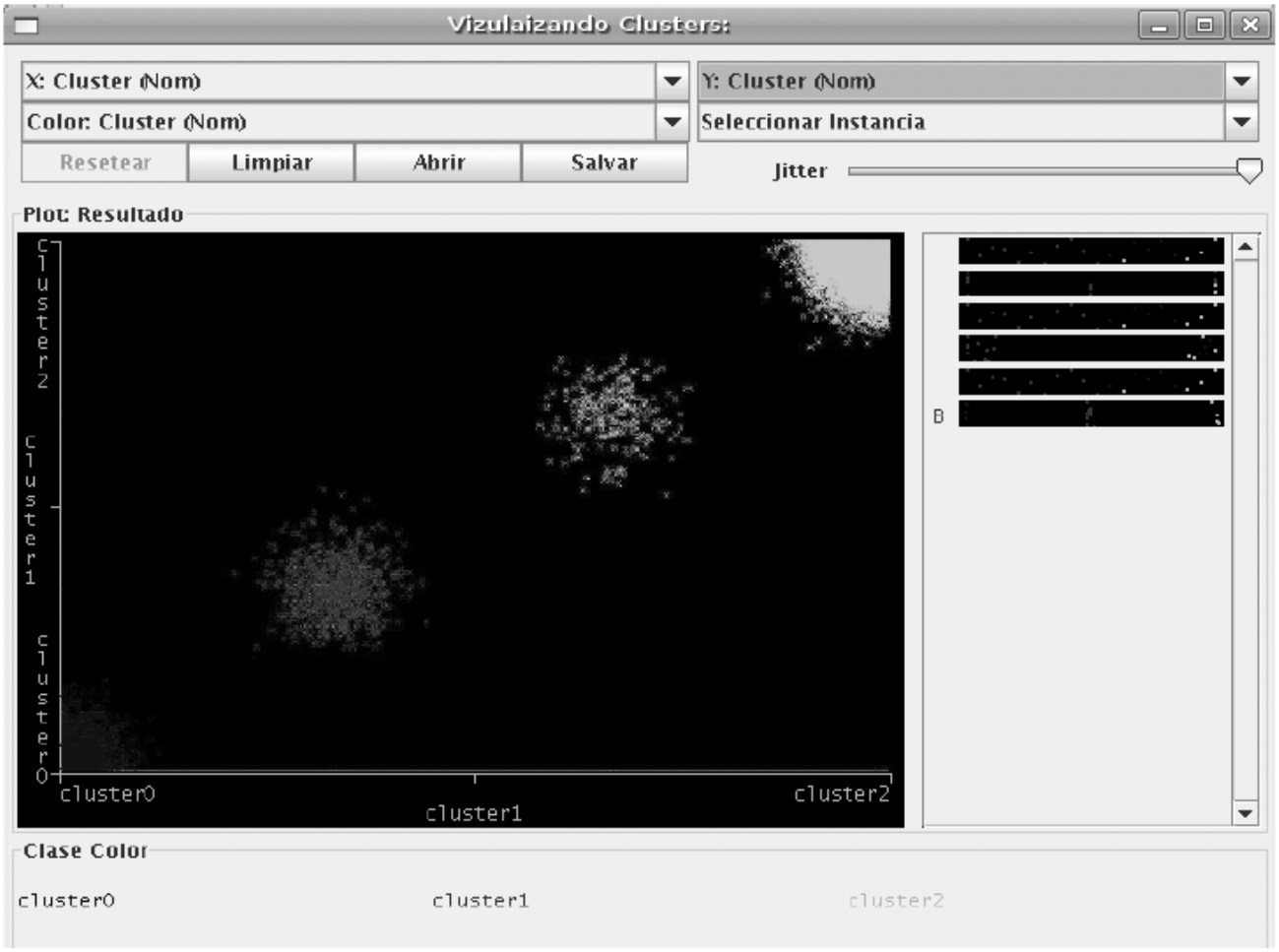
ANEXOS



Anexo 1: Biología de Sistema



Anexo 2: Patrón DAO



Anexo 3: Clustering

GLOSARIO DE TÉRMINOS

BioSyS: Biological System Simulator (Software para la Simulación de Sistemas Biológicos).

Base de Datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

SGBD: Sistema Gestor de Base de Datos.

DAO: Data Access Object (Objeto de Acceso a Datos).

Análisis: Estudio, mediante técnicas informáticas, de los límites, características y posibles soluciones de un problema al que se aplica un tratamiento por ordenador.

Clasificación: Es la acción o el efecto de ordenar o disponer por clases.

Clusters: Grupo de objetos que comparten características semejantes.

Clustering: Es la clasificación de los objetos en diferentes grupos.

Algoritmo: Es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.

Simulación: Es un intento de modelar situaciones de la vida real por medio de un programa de computadora.

Editor de Ecuaciones: Herramienta para alinear los símbolos matemáticos correctamente.

FN: Forma Normal.