

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
DIRECCIÓN DE FORMACIÓN POSTGRADUADA

Titulo: Sistema para el almacenamiento y acceso de la información asociada a los procesos de estudio y análisis de simulaciones de sistemas biológicos representados mediante sistemas de ecuaciones diferenciales.

Tesis en opción al grado de
Máster en Informática Aplicada

Autora: Lic. Nara Lidia Pérez Solá

Tutor: MsC. Maypher Román Durán

MsC. Vladir Antonio Parrado Cruz

Ciudad de La Habana, Julio de 2010

DEDICATORIA Y AGRADECIMIENTOS

SÍNTESIS

El desarrollo de la Biología ha alcanzado, en la actualidad, un nivel elevado, revolucionando consigo, además, la investigación biomédica y dentro de ésta, la Biología de Sistemas, área de la biología que hace uso de modelos matemáticos que describen el comportamiento del ente en estudio. El análisis sobre estos modelos de sistemas biológicos genera en cada estudio, una gran cantidad de datos, por lo que se hace necesario la utilización de herramientas informáticas que garanticen el almacenamiento correcto y seguro de la información generada, ya que actualmente son las bases de datos relativas a la biología las que más rápido crecimiento tienen, y en las que más tiempo de desarrollo se invierte. El uso de estas bases de datos a menudo se ve obstaculizado por el hecho de que estén diseñadas para áreas específicas y, por tanto, carecen de universalidad. Los experimentadores de esta área, si desean almacenar sus propios datos, posiblemente aún en fase de investigación, tienen dos posibilidades: pueden desarrollar su propio sistema de información para gestionar los datos que poseen, o pueden tratar de almacenarlo en los sistemas existentes, que a menudo son limitados al acceso de sólo lectura. El presente trabajo muestra el desarrollo de un sistema de gestión que permite el procesamiento dinámico de la información relacionada con los estudios y el análisis de las simulaciones de sistemas biológicos, mejorando las expectativas del cliente y brindando al investigador, una herramienta poderosa, no solo por su valioso contenido sino también por sus facilidades de uso.

Palabras Claves:

Biología de Sistemas, modelos matemáticos, simulaciones.

ÍNDICE

INTRODUCCIÓN.....	1
1 MARCO TEÓRICO	9
1.1 Introducción.....	9
1.2 Biología de Sistema.....	9
1.2.1 Simulaciones de Sistemas Biológicos.....	10
1.2.2 Técnicas de Análisis	13
1.2.2.1 Algoritmos de Clustering.....	13
1.2.2.2 Clasificación Manual.....	14
1.2.2.3 Clasificación por Weka	14
1.2.2.4 Análisis por Reglas.....	15
1.3 Datos Biológicos.....	15
1.3.1 CellML	15
1.3.2 SBML	15
1.4 Sistemas de gestión de sistemas biológicos.....	18
1.5 Almacenamiento de información de manera consistente	22
1.6 Formas de acceso a bases de datos	23
1.7 Conclusiones del capítulo	24
2 MATERIALES Y MÉTODOS	25
2.1 Introducción.....	25
2.2 Metodología de desarrollo de software	25
2.2.1 OpenUP.....	25
2.2.2 Herramientas CASE	25
2.2.2.1 Visual Paradigm	25
2.2.3 Gestores de Bases de Datos	26
2.2.4 Lenguajes utilizados	27
2.2.4.1 Lenguaje de marcado.....	27

2.2.4.2	Lenguaje Java	27
2.2.4.3	Lenguaje de Consulta Estructurado (SQL)	28
2.3	Herramientas de desarrollo.....	28
2.3.1	Netbeans	28
2.3.2	Hibernate.....	29
2.3.2.1	Análisis de optimización de consultas.....	30
2.3.3	Apache JMeter	30
2.4	Arquitectura de software	31
2.4.1	Web service (WS).....	31
2.4.2	Patrón DAO	33
2.5	Conclusiones del capítulo.....	34
3	RESULTADOS.....	35
3.1	Análisis del proceso de almacenamiento de los datos persistentes	35
3.1.1	Análisis del proceso de recuperación de la información almacenada	38
3.2	Diseño de la solución.....	38
3.3	Implementación del sistema	39
3.3.1	Implementación de la capa de acceso a datos	40
3.3.2	Implementación del Servicio Web.....	41
3.4	Conclusiones del capítulo.....	43
4	VALIDACIÓN	44
4.1	Introducción	44
4.2	Pruebas de volumen.....	44
4.3	Pruebas de carga	46
4.4	Prueba de stress	51
4.5	Conclusiones del capítulo.....	53
	CONCLUSIONES.....	54
	RECOMENDACIONES.....	55
	REFERENCIAS BIBLIOGRÁFICAS.....	56

ANEXOS.....	60
Anexo 1: Informe Agregado del Plan1 para 240 usuarios concurrentemente. (3600 peticiones a la base de datos).....	60
Anexo 2: Informe Agregado del Plan2 para 240 usuarios concurrentemente. (12983 peticiones a la base de datos).....	60
Anexo 3: Monitor del sistema para el Plan1 con 240 usuarios.	61
Anexo 4: Monitor del sistema para el Plan2 con 240 usuarios.	61

INTRODUCCIÓN

El desarrollo de las técnicas para la obtención de datos a gran escala, facilita el progreso de todas las ramas de la biología, como: la bioinformática, la biología molecular, la biología de sistemas, la biología evolutiva, la genética, entre muchas otras; ya que las bases de datos referentes a datos biológicos, son las que más datos almacenan. Estas disciplinas, en conjunto con la revolución tecnológica que se vive en la actualidad, permiten la descripción detallada de los seres vivos en diferentes niveles, proporcionando además las herramientas necesarias para ordenar y dar sentido a la gran cantidad de información que se genera del estudio de sistemas biológicos.

La Biología de Sistemas es un área de investigación científica que se preocupa del estudio de procesos biológicos usando un enfoque sistémico (1). Esta área 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. 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. Esta modelación puede ser, del sistema como un todo, o de las interacciones de los componentes que lo forman.

Esta descripción matemática de un fenómeno del mundo real, que puede ser el crecimiento de las poblaciones de animales, el funcionamiento de las neuronas y la dinámica intracelular, por citar sólo algunos ejemplos de su aplicación en biología, tiene la finalidad de representar de manera comprensible, el comportamiento de dichos procesos para lograr entender en profundidad el fenómeno, probar teorías y tal vez realizar alguna predicción sobre su comportamiento futuro. Algunos de los modelos más usados son las Redes Neuronales y los Sistemas de Ecuaciones Diferenciales (SED), pero han sido

estos últimos, los de más amplio uso en la modelación de problemas biológicos, debido fundamentalmente, a las facilidades de modelación y de simulación que brindan.

Con SED se pueden describir gran cantidad de procesos biológicos y los métodos numéricos para la resolución de los mismos son muy conocidos y están disponibles tanto en la literatura como en diversos software (2).

En la actualidad, existen muchas aplicaciones basadas en estándares que permiten modelar, simular o realizar análisis sobre modelos de sistemas biológicos (3) (4) (5), descritos mediante SED u otras técnicas. Estos sistemas experimentales han permitido obtener datos biológicos a gran velocidad. Producto a esto, existen gran cantidad de bases de datos relacionadas con el tema y centros u organizaciones especializadas en tener y mantener estas bases de datos (6).

Los resultados del trabajo de cada área de la Biología de Sistema, se reflejan en los ritmos de crecimiento de las bases de datos que se van generando. Actualmente se disponen de más de 600 bases de datos biológicas gestionadas por diferentes grupos de investigación (7). Sus características desde la perspectiva bioinformática son: la dispersión geográfica, la diversidad del tipo de datos que almacenan y las diferencias en formatos y formas de acceso.

Las bases de datos más populares que existen actualmente y que son ampliamente utilizadas por especialistas e investigadores a nivel internacional están centradas mayormente en el área de la biología molecular. Son muy pocas las bases de datos, que están enfocadas en la biología de sistema como Biomodels (<http://www.ebi.ac.uk/biomodels-main>). Esta base de datos, al igual que las demás, sólo se centra en el almacenamiento de modelos matemáticos curados¹ asociados a los sistemas biológicos. Además brinda la posibilidad de realizar simulaciones sobre estos modelos predefinidos, cuyos resultados se le muestran a los investigadores pero no son almacenados en el propio sistema. En este sentido, esta base de datos carece de un

¹ Se comprueba, por un equipo de especialistas, la validez del modelo matemático.

proceso de retroalimentación donde los investigadores puedan hacer análisis de diferentes simulaciones previamente realizadas.

Además, el almacenamiento de los modelos matemáticos lo hace en un XML destinado al trabajo de sistemas biológicos. Para acceder a cualquier información relacionada con los modelos (variables o parámetros), contenida en el fichero, hay que interpretarlo previamente. Estas operaciones de acceso y búsquedas son muy costosas debido a que la información referente a los modelos matemáticos no es la única almacenada en dicho fichero.

A pesar de este desarrollo tecnológico surge otra limitante: estos sistemas que permiten modelar, simular o realizar análisis sobre modelos de sistemas biológicos, sólo lo realizan de manera independiente; es decir, que no existe ninguno que agrupe estas funcionalidades como un todo. La desventaja más significativa es que las bases de datos que guardan la información de los sistemas biológicos se centran, por lo general, en almacenar todo lo relacionado con los modelos, ya sean matemáticos o gráficos; pero no almacenan información a cerca las simulaciones que sobre estos se realizan, ni almacenan la información generada del estudio y análisis de estas simulaciones. Esto implica que, aunque se disponga de grandes repositorios de modelos, si se desea saber cómo funcionan dichos sistemas, se debe repetir el estudio previamente realizado.

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 de los sistemas biológicos modelados mediante SED. Estos estudios se realizan, para conocer la respuesta temporal de un sistema a partir de un conjunto de condiciones iniciales y una entrada de datos dada. La carencia de base de datos donde se almacene toda esta información o de un repositorio de estudios biológicos, frena el desarrollo de estas investigaciones e imposibilita, a los especialistas, realizar estudios más profundos sobre los sistemas biológicos.

El CIM, es un centro que tiene un reconocido prestigio tanto en la investigación básica como en la producción de medicamentos y muchos de sus trabajos se centran en el área de la biología de sistemas; en sus investigaciones, genera un gran volumen de información pero presenta problemas con el almacenamiento, acceso e interpretación de la misma, obstaculizando en gran medida el avance y desarrollo de sus estudios, de ahí que se requieran de métodos y técnicas innovadoras que permitan almacenar y recuperar la información deseada, garantizando así que se puedan realizar búsquedas de información, homologías y patrones, así como, realizar predicciones con la información obtenida.

Además, este centro no está exento a las limitaciones que internacionalmente presentan los expertos que trabajan en el área de la Biología de Sistemas. (8) Algunas de esas limitaciones son:

- La formulación de modelos matemáticos más robustos pues, los sistemas a niveles superiores muestran diferentes características cuantitativas y cualitativas que los subsistemas de niveles inferiores no poseen. Por esta razón, los métodos de modelización que son efectivos a un nivel deben ser revisados antes de ser aplicados a niveles de jerarquía superiores. Todo esto es esencial para describir el comportamiento del sistema biológico y permite desarrollar predicciones sobre el comportamiento del sistema que se está estudiando.
- La estandarización de experimentos *in vitro* e *in vivo* y sus datos es a menudo inconsistente, inaccesible, incompleta, o desestructurada. Por este motivo, los datos han de documentarse de forma homogénea en las ocasiones en las que sea posible, teniendo en cuenta los estándares de recolección de datos, almacenamiento en bases de datos, y análisis. Muchos han señalado este punto como un cuello de botella para el desarrollo de la Biología de Sistemas (8).

De la situación problemática planteada anteriormente, se identifica el **problema científico**: *¿Cómo mejorar el proceso almacenamiento y acceso de la información asociada a los procesos de estudio y análisis de simulaciones de sistemas biológicos representados mediante SED en el CIM?*

Considerando lo anterior el **Objeto de Estudio** es: *el proceso de gestión de información y el campo de acción: el proceso de almacenamiento y acceso de la información asociada a los procesos de estudio y análisis de simulaciones de sistemas biológicos representados mediante SED en el CIM.*

En correspondencia con el problema planteado, el **Objetivo General** de esta investigación consiste en: *desarrollar un sistema informático para el proceso de almacenamiento y acceso de la información asociada a los procesos de estudio y análisis de simulaciones de sistemas biológicos representados mediante SED para el CIM.*

Para guiar el trabajo de esta investigación, se identificaron las siguientes preguntas científicas:

- ¿Qué técnicas son necesarias para almacenar en un formato adecuado, la información asociada a los procesos de estudio y análisis de simulaciones de sistemas biológicos representados mediante SED?
- ¿Qué mecanismos aceleran el acceso a la información asociada a los procesos de estudio y análisis de simulaciones de sistemas biológicos representados mediante SED?

Las **Tareas de la Investigación** desarrolladas para cumplir los objetivos son:

- ✓ Revisión bibliográfica sobre simulaciones de sistemas biológicos, modelos matemáticos y técnicas de análisis a sistemas biológicos.
- ✓ Revisión bibliográfica sobre software que almacenen datos generados del proceso de simulación de sistemas biológicos.
- ✓ Revisión bibliográfica sobre software que almacenen datos generados del proceso de análisis de sistemas biológicos.
- ✓ Revisión bibliográfica sobre las base de datos estándares en la biología de sistemas.
- ✓ Revisión bibliográfica sobre formas de almacenar información biológica de manera consistente.

- ✓ Revisión bibliográfica sobre las posibles soluciones para cada uno de los puntos críticos identificado en la investigación.
- ✓ Evaluación de las alternativas de solución para los puntos críticos identificados en el sistema.
- ✓ Diseño de la base de datos
- ✓ Implementación de la capa de acceso a datos utilizando el concepto de Servicios Web (WS).
- ✓ Realizar pruebas que validen el correcto funcionamiento de la aplicación.

Aportes esperados del trabajo

Los procesos de simulación y análisis de sistemas biológicos tienen un alto costo en el mercado internacional. Estos permiten acelerar o retardar el experimento sobre sistemas biológicos según convenga, y en el país no se cuenta con un sistema que empaquete estos dos procedimientos. Por tanto, en la actualidad los centros del polo del país necesitan de un sistema de gestión de la información arrojada, en los procesos de simular y analizar los sistemas biológicos.

De una manera más puntual, podría decirse que los beneficios esperados del desarrollo de este trabajo serán:

- Contar con un sistema que mejora el proceso de almacenamiento, acceso e interpretación de la información asociada a las simulaciones que se realizan de sistemas biológicos, así como el estudio y los análisis que de estas se desarrollan.
- Contar con una aplicación totalmente orientada a servicios, que cumpla con los estándares internacionales de bases de datos en sistemas biológicos, para que la comunidad científica pueda interactuar con ella y marcar pautas de desarrollo intelectual.
- Reducir los tiempos de las investigaciones sobre sistemas biológicos, en el análisis y la simulación de los mismos.

Impacto socioeconómico de la investigación

La evaluación de este trabajo tiene un gran impacto tecnológico-socioeconómico en el país. El proceso de estudio y análisis de sistemas biológicos tiene un costo elevado en el mercado, por lo que la posibilidad de contar con un repositorio de información de todas las simulaciones que se realizan a los sistemas biológicos en estudio, acelera las investigaciones en los centros del polo del país.

El resultado esperado con esta aplicación, incide directamente en uno de los problemas que más le afecta a los centros cubanos actuales del polo investigativo: el tiempo. Estos estudios normalmente duran de 10 a 12 años y contar con esta herramienta, representa un paso importante en el mejoramiento de la competitividad de las empresas de software cubanas en el mercado mundial.

El análisis de las simulaciones, es también un área de mucho impacto tanto en la sociedad como en las investigaciones científicas. Algunas aplicaciones de las soluciones de Minería de Datos, sobre datos biológicos, pueden ser:

- Descubrir distintos comportamientos de una misma patología.
- Predecir las patologías que pueden aparecer como complicación de una enfermedad dada.
- Descubrir asociaciones entre patologías.

Métodos de investigación

Los métodos empleados en la investigación se dividen en teóricos, empíricos y estadísticos.

Análisis – síntesis: Se aplicó al análisis bibliográfico, definiciones y enfoques de diferentes autores sobre las formas de almacenamiento de la información biológica, dígase, modelos matemáticos de sistemas biológicos, las simulaciones de estos modelos, y el análisis e interpretación que estas simulaciones se puede obtener.

Histórico – lógico: Se utilizó para hacer un análisis retrospectivo sobre la utilización de los sistemas de gestión de datos biológicos.

Análisis de documentos: Posibilitó realizar el análisis de documentos referentes al tema en cuestión para llevar a cabo la fundamentación y el desarrollo de la investigación.

Estructura de la tesis

El desarrollo de la investigación del presente trabajo está estructurado en un volumen de 68 páginas, compuesto por varias secciones que forman tres capítulos, además de introducción, conclusiones, bibliografía, glosario de términos y anexos.

En el capítulo uno se hace una revisión bibliográfica del estado actual de la temática en estudio, con el objetivo de caracterizar los procesos tratados, así como, los avances logrados y la utilidad que tendría para la comunidad científica que se gestionaran los datos asociados a cada uno de los estudios y análisis de las simulaciones. En el capítulo dos se describen los materiales y métodos usados para poner en práctica la solución propuesta, así como, la metodología empleada para desarrollar los artefactos y la arquitectura requerida. En el capítulo tres, se abordan los resultados obtenidos en el desarrollo del trabajo y las consideraciones finales asociadas al respecto, con las cuales se da respuesta a las preguntas científicas planteadas en la investigación y en capítulo cuatro se realizan las pruebas para validar el comportamiento del sistema.

1 MARCO TEÓRICO

1.1 Introducción

El objetivo principal de este capítulo es abordar los temas relacionados con la Biología de Sistema y dentro de ésta, todo lo relacionado con las simulaciones de sistemas biológicos, haciendo énfasis en los modelos matemáticos utilizados para representar las simulaciones y en las técnicas de análisis aplicadas a las simulaciones, así como, los avances logrados en el proceso de almacenamiento y recuperación de información biológica y la utilidad que tendría para la comunidad científica que se realizaran estos dos procesos antes mencionados.

1.2 Biología de Sistema

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. Comenzó a desarrollarse en los años sesenta del siglo XXI y desde entonces, hace uso de modelos matemáticos para describir el comportamiento del ente en estudio. Los modelos, permiten predecir el comportamiento del proceso como un sistema dinámico, generalmente tratado como una red compleja. 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.

La Teoría General de Sistemas (TGS) afirma que las propiedades de los sistemas no pueden ser descritas en términos de sus elementos separados, sino que su comprensión se presenta cuando se estudian globalmente (9). El análisis de la Biología de Sistemas constituye pues un nuevo paradigma de investigación en Biología. Constituye una estrategia holística, a partir de los datos obtenidos en las anteriores áreas de conocimiento, que describiría la estructura y los mecanismos de control de los fenómenos biológicos, en estado fisiológico o tras una perturbación, formulando modelos que expliquen las complejas interrelaciones moleculares (vías y rutas, redes, módulos, etc) que se producen en el interior de las células y en los organismos superiores (10).

Esta disciplina permite estudiar los mecanismos que gobiernan los sistemas complejos y las interacciones existentes entre los diferentes niveles de información biológica. Por otra parte, trata de entender los sistemas biológicos a diferentes niveles de abstracción, desde el nivel molecular hasta los ecosistemas, haciendo uso de diferentes tipos de modelos matemáticos y de técnicas computacionales.

1.2.1 Simulaciones de Sistemas Biológicos

La simulación puede ser definida como la imitación de un sistema dinámico utilizando un modelo computacional animado, con el objetivo de evaluar y mejorar la ejecución de un sistema existente o que aún no existe. El modelo imita las respuestas del sistema actual a los eventos que toman lugar en el tiempo. A través del estudio de la conducta de un modelo de simulación, se puede predecir la conducta del sistema real (11).

Son varios los criterios y las definiciones que se plantean alrededor de este término:

Según Thomas Naylor, profesor en la Universidad de Duke y uno de los fundadores de la Segunda República de Vermont, plantea que: la simulación, es una técnica numérica para conducir experimentos en una computadora digital, las cuales requieren ciertos tipos de modelos lógicos y matemáticos, que describen el comportamiento de un negocio o un sistema económico (o algún componente de ellos) en períodos extensos de tiempo real (12).

Roger Schroeder alega: La simulación es una técnica que puede utilizarse para resolver una amplia gama de modelos. Su aplicación es tan amplia que se ha dicho: "Cuando todo falle, utilice simulación" (12).

Partiendo de estos criterios, es posible determinar que el volumen de datos que genera la simulación de un sistema dado, conduce a un mejor entendimiento del mismo y proporciona sugerencias para mejorarlo. Mediante el empleo de la técnica de simulación, se adquiere experiencia que puede ser más valiosa que la simulación en sí misma.

El objetivo de la simulación es descubrir y entender el comportamiento de un sistema, en este caso biológico, para formular teorías o hipótesis que expliquen lo observado y puedan ser confrontados con los resultados experimentales. Estas teorías, son utilizadas

para predecir el comportamiento futuro del sistema biológico. Para la realización de las simulaciones, los sistemas biológicos son representados mediante modelos matemáticos, utilizando SED.

Estos modelos son una representación matemática de los mecanismos que gobiernan el comportamiento del sistema (atributos) y de su interacción con el entorno, permitiendo el estudio mediante el ordenador de la conducta de dicho sistema (13).

Por todo lo anteriormente dicho, los modelos matemáticos, son un tema central en la biología de sistemas, que permite entender, predecir, controlar o inclusive diseñar un sistema biológico real y que tiene como objetivo principal lograr nuevos conocimientos sobre el comportamiento del sistema (14).

La figura 1, muestra los pasos necesarios para la construcción del modelo matemático.

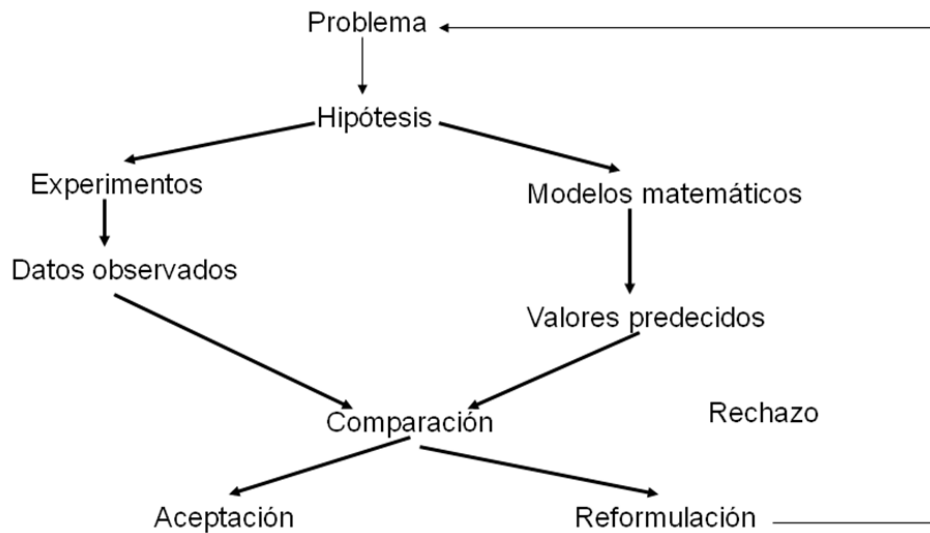


Figura 1. Diagrama de flujos en el proceso de modelación

La mayoría de los SED que se obtienen como resultado de la modelación de problemas reales son altamente no lineales y no tienen solución analítica, por lo que se hace necesario recurrir al uso de métodos numéricos. (2) En los últimos años, han sido

desarrollados muchos métodos numéricos para este fin, buscando siempre las mejores aproximaciones, algunos ejemplos de ellos lo son:

- Runge-Kutta (15) (16)
- Adams (17)
- Rosenbrock (18) (19)
- Euler (17) (20)

Finalmente se puede concluir que a través de la integración del conocimiento obtenido en cada subsistema, la biología de sistema, puede estudiar el sistema global modelando las relaciones entre sus partes. Estas relaciones se representan mediante sistemas de ecuaciones diferenciales que se simulan, utilizando los resultados, a su vez, para refinar el modelo inicial y al final compararlo con los casos reales. La figura 2, muestra el proceso descrito anteriormente.

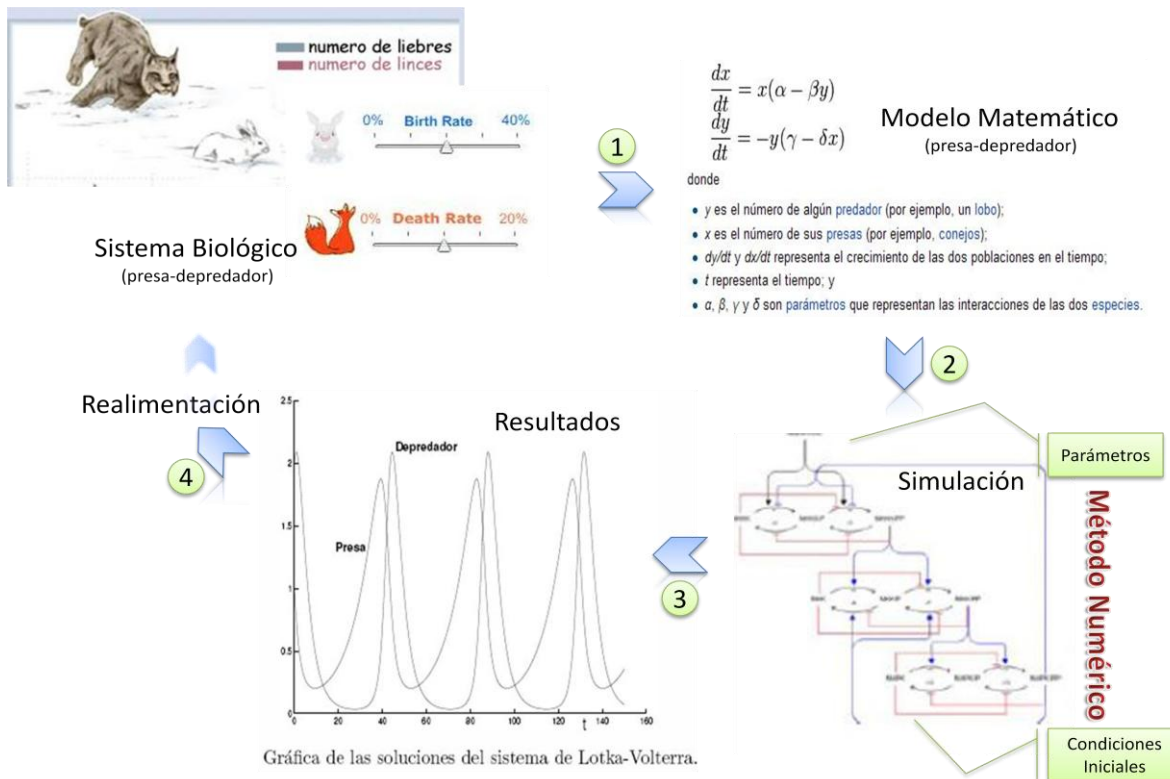


Figura 2. Proceso de simulación de sistemas biológicos. Elaboración propia

La mayoría de los sistemas de gestión de la información asociados a los sistemas biológicos, almacenan los datos referentes, nada más, a los modelos matemáticos que

representan los sistemas en estudio, los que posteriormente se pueden simular y analizar, si se quiere obtener más conocimiento.

1.2.2 Técnicas de Análisis

Los modelos matemáticos almacenados en la base de datos, carecen de utilidad sino se hace uso de herramientas que permitan hacer minería de estos datos y faciliten la comprensión de los mismos, ya que es poco probable que se pueda explorar de forma manual toda la información que de cada modelo se obtiene, con el objetivo de extraer el conocimiento implícito que estos datos poseen.

Un aspecto a tener en cuenta, y donde la Biología de Sistema y la Bioinformática de manera general tienen las mayores expectativas de crecimiento y de ingresos, lo es sin duda, la minería de datos; técnica que constituye hoy, una herramienta muy sofisticada para predicciones de todo tipo y toma de decisiones con todo rigor y fiabilidad.

Por lo que se puede afirmar que la minería de datos es fundamental en la investigación científica y técnica, como herramienta de análisis y descubrimiento de conocimiento a partir de datos de observación o de resultados de experimentos (21). La información que genera al aplicar estas técnicas debe ser correctamente gestionada pues contribuye a la toma de decisiones efectivas.

Los datos a gestionar en el sistema, proceden de la aplicación de las siguientes técnicas de análisis:

1.2.2.1 Algoritmos de Clustering

El Clustering es un tipo de algoritmo de aprendizaje inductivo en el que los datos sobre los que se trabaja no están normalmente etiquetados. Consiste en agrupar los datos según algún criterio, normalmente para minimizar algún objetivo, de acuerdo con una medida de similitud. Se puede emplear como un paso exploratorio para una posterior clasificación, para una reducción de datos, para simplificar un proceso auxiliar, etc (22).

Estos algoritmos permiten clasificar un conjunto de elementos de muestra en un determinado número de grupos, basándose en las semejanzas y diferencias existentes entre los componentes de la muestra.

Un algoritmo de agrupamiento (en inglés, clustering) es un procedimiento de agrupación de una serie de vectores según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes (23). Clustering, es una técnica, en la que el aprendizaje es no supervisado, jugando un papel muy importante en aplicaciones de minería de datos, tales como: exploración de datos científicos, diagnóstico médico, análisis de ADN y muchas otras.

Una gran variedad de algoritmos de clustering han surgido en los últimos años, cada uno con sus características específicas, en esta investigación se trabaja con la información generada al aplicar tres de ellos: CobWeb (24), XMeans (25) y SimpleKMeans (23).

1.2.2.2 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.

1.2.2.3 Clasificación por Weka

WEKA, es una herramienta que permite realizar minería de datos con una interfaz gráfica lo que facilita su utilización. Además, permite una comparación con los distintos métodos que se utilizan para el pre-procesamiento, clasificación de información, clustering y meta-aprendizaje. WEKA proporciona una plataforma para evaluar un problema con distintas combinaciones de algoritmos y poder extraer conocimiento interesante (22).

Una de las funcionalidades del weka que se ha incorporado al sistema, 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. Ambos conjuntos de datos son gestionados en la base de datos.

1.2.2.4 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).

1.3 Datos Biológicos

El crecimiento exponencial de los datos biológicos, ha impulsado al uso, insustituible, de herramientas que permitan a los biólogos almacenar, organizar, buscar, manipular y recuperar dichos datos.

Todas las bases de datos biológicas contienen gran cantidad de información, pero en diferentes códigos y formatos como: SBML (del inglés Systems Biology Markup Language) y CellML (del inglés Cell Markup Language).

1.3.1 CellML

El lenguaje CellML es un estándar basado en el lenguaje de marcas XML, cuyo propósito es almacenar y facilitar el intercambio de modelos matemáticos relevantes al campo de la biología. CellML incluye la descripción de la organización de un modelo en componentes (partes), y cómo estas partes están conectadas entre sí para crear un modelo completo (26).

Básicamente es un lenguaje basado en XML que permite el intercambio de modelos que describen los procesos celulares y subcelulares. Utiliza MathML (27) para describir la matemática de los modelos de celulares y es muy bueno para el modelado celular, pero no está adaptado para el modelado multi-celular. Entre sus características principales, además de incluir el trabajo con metadatos, se destaca que CellML puede convertirse fácilmente en otros formatos, incluyendo SBML

1.3.2 SBML

SBML es un formato legible por máquina para la representación de modelos. Es orientado a describir los sistemas en que las entidades biológicas están involucrados, y modificado por los procesos que ocurren en el tiempo. Un ejemplo de esto es una red de reacciones

bioquímicas. SBML es utilizado para la representación de los modelos que se encuentran comúnmente en la investigación sobre una serie de temas, incluyendo vías de señalización celular, las vías metabólicas, las reacciones bioquímicas, la regulación de genes, y muchos otros (28).

Se puede concluir que un SBML, no es más que un lenguaje de marcado para sistemas biológicos, basado en XML; un fichero estándar que almacena toda la información asociada a un modelo tanto gráfico como matemático de un sistema biológico. En muchas ocasiones éste estándar es utilizado para el intercambio de información entre herramientas.

Sin embargo, el SBML no representa un intento de definir un lenguaje universal para la representación de modelos cuantitativos biológicos (28). Una alternativa más realista es reconocer la diversidad de enfoques y métodos que se exploran en la biología de sistemas y que pueden ser utilizados para manipular dicha información, teniendo en cuenta, además, las limitaciones del formato SBML, como por ejemplo: no apoya la definición de rangos de valores de los parámetros. En la siguiente tabla, se realiza una comparación entre CellML y SBML:

Criterios de comparación	CellML	SBML
Estructura	<p>Un modelo CellML se construye como una red de componentes. Un componente puede contener variables, expresiones matemáticas, etc.</p> <p>La información biológica es, almacenado en los metadatos en lugar de los elementos del lenguaje.</p>	<p>Se basa en las sucesivas y jerárquicas declaraciones de los componentes del modelo</p>
Aplicación	<p>Se utiliza para el modelado multiescala biológica, biología de sistemas, biología sintética y la</p>	<p>Se especifica en modelos biológicos, en particular la biología de sistemas</p>

	variedad de modelización de problemas de ingeniería.	(metabolismo, de señalización celular y las redes de reacciones bioquímicas).
Modularidad y modelo de composición	Los modelos se basan en una estructura modular, que permiten la creación de nuevos y/o modelo de gran tamaño utilizando la reutilización de los modelos existentes. CellML expresar la modularidad utilizando un elemento de importación que permite la reutilización anidada modelo.	Actualmente carece de esta característica.
Tiempo de retardo	Carece de esta característica.	Soporta las variables retrasadas usando función de retardo que es útil para la representación de los procesos biológicos que tengan una respuesta retardada.
Metadato y anotación	Los modelos matemáticos se describe utilizando el lenguaje CellML, todo lo demás se describe a través de los metadatos empleando Dublin Core, BQS and modelos de datos vCard	La información biológica y los modelos están embebidos en los elementos del lenguaje. Se refiere a términos de vocabulario controlado y los identificadores de base de datos que definen y describen las entidades biológicas y bioquímicas
Ontología	SBO, BioPAX y otras utilizando su framework de metadatos	SBO
Modelado basado en reglas	Ambos no soportan el modelado basado en reglas.	
SBML se considera más aceptado que CellML, puesto que este último recibe menos ayuda de terceras personas (8) además de ser un año más joven.		

1.4 Sistemas de gestión de sistemas biológicos

Muchos son los que han reconocido la importancia, que los métodos informáticos de procesamiento de información tienen, para la realización vertiginosa de las investigaciones en el área de la biología y sus ramas. La biología es una ciencia que genera, años tras años, un enorme volumen de complejos datos, por lo que existen centros e instituciones cuya misión es la de administrar esta avalancha de datos en diversas bases de datos.

Hasta ahora, el almacenamiento de datos biológicos se ha llevado a cabo en tres núcleos principales del mundo: en Japón; en Estados Unidos, a través del NIH (del inglés National Institutes of Health); y en Europa, a través del EMBL (del inglés European Molecular Biology Laboratory), de Heidelberg (Alemania), y su nodo, el EBI (del inglés European Bioinformatic Institute), situado en Hinxton (Gran Bretaña). Todos ellos disponen de bases de datos diferenciadas.

En los años 80-90 comenzaron a proliferar los trabajos científicos de secuenciación de genes y proteínas. Los resultados se fueron almacenando de diferentes maneras y en diferentes lugares. En Europa, Swiss-Prot (Suiza) (29) fue la primera base de datos bien organizada, hasta que le tomó el relevo UniProt (del inglés Universal Protein Resource) (30), que fusionó los datos de la base de datos suiza y otros. Y en todo el mundo surgieron más, como BioGRID, de Canadá; DIP (del inglés Database of Interacting Proteins), GenBank, de Estados Unidos; HPRD (del inglés Human Protein Reference Database), de la India; o el MIPS (del inglés Munich Information Center for Proteins Sequencing), de Alemania. Y la lista no se acaba aquí (31).

Pese a este gran intento por contar con grandes repositorios que agrupen y administren información biológica, la mayoría de las bases de datos, más populares y utilizadas, están relacionada con la Biología Molecular (tabla 1), otras, más personalizadas, están relacionadas con el estudio de la estructura de las proteínas, tarea que puede llevar entre uno y dos años, mientras que su secuenciación es cada vez más rápida. Por este motivo,

estas bases de datos contienen información de la estructura de cerca de 50.000 proteínas, de más de 500.000 secuencias con su función.

Bases de Datos (6)	Descripción
GeneBank	Colección pública de secuencias tanto de proteínas como de ácidos nucleicos con soporte bibliográfico
DDBJ, EMBL	Guarda toda la secuencia de ADN conocidas y se actualizan a diario
Protein Data Bank (PDB), Uniprot	Información funcional de proteínas
BIANA (<i>Biologic Interactions and Network Analysis</i>)	Plataforma que reúne múltiples fuentes de información biológica y es capaz de inferir, a partir de interacciones ya conocidas de proteínas y genes, nuevas posibilidades de interacción de biomoléculas
BioCyc, KEGG, BioCarta, STKE, Reactome	Bases de datos de rutas metabólicas
BIND, DIP, MINT, PPID	Bases de datos con información de interacciones de proteínas
Ensembl, UCSC Genome Bioinformatics	Visualizador de genomas completos

Tabla 2. Bases de Datos Biológicas Populares

La búsqueda de sistemas, relacionados con la Biología de Sistemas, que almacenaran los tres procesos tratados en el trabajo en un solo medio, arrojó muy pocos resultados, puesto que todas las bases de datos estudiadas se centraban en temas específicos de la biología, almacenando nada más lo relativo a los modelos matemáticos, sin guardar nada relacionado ni con las simulaciones de esos modelos, ni con los análisis de dichas simulaciones, además de usar SBML como formato de entrada y salida de los datos:

SABIO-RK (del inglés **S**ystem for the **A**nalysis of **B**iochemical Pathways - **R**eaction **K**inetics): Almacena información acerca de las reacciones bioquímicas y sus características cinéticas. Su objetivo es respaldar a los especialistas en la creación de modelos de redes bioquímicas, pero también es útil para investigadores con interés en las reacciones bioquímicas y su cinética (32).

Esta base de datos, de sólo lectura, puede ser accedida a través de una interfaz web y la información sobre las reacciones, sus leyes cinéticas (si existen) y la definición de parámetros y valores puede ser exportada en SBML, con el inconveniente de que si un valor de parámetro se define como un rango, el sistema toma los valores medio de este rango como valor del parámetro en SBML, teniendo en cuenta que el archivo SBML no apoya la definición de rangos de valores de los parámetros.

Berkeley Madonna v8.0.1: Programa para analizar y modelar la dinámica de sistemas. Es uno de los programas más utilizados para resolver ecuaciones diferenciales de propósitos generales disponibles en la actualidad. Desarrollado en Berkeley bajo el patrocinio de NSF y NIH, (33) se usa actualmente con fines académicos y comerciales en la construcción de modelos matemáticos. Entre las funciones que lo destacan se encuentra la construcción del modelo matemático (a partir del conocimiento de las ecuaciones, o a partir del conocimiento de las reacciones químicas que intervienen en el sistema en estudio) y el trabajo con parámetros (ajuste de curvas, optimización, análisis de sensibilidad y graficación).

La versión académica es totalmente funcional con las excepciones siguientes: Los modelos no se guardan, las gráficas y tablas no se guardan ni se pueden copiar, entre otras dificultades. Si se desea para una comunidad de investigadores, se debe pagar la versión comercial, que si permite exportar las gráficas y tablas en formato de mapa de bit (.bmp) y las ecuaciones del modelo matemático, en formato de texto. Debiendo el usuario almacenar cada elemento por separado y recordar los nombres usados en cada uno de ellos para un estudio más específico. Otra de las desventajas es que en la actualidad solamente corre en los sistemas operativos de Windows y Mac, aunque en la página oficial del software [<http://www.berkeleymadonna.com>], plantean que están en desarrollo de una nueva versión en java para el sistema operativo Linux.

BioModels: Es una base de datos estática de recursos, que permite a los biólogos almacenar, buscar y recuperar modelos matemáticos publicados, de interés biológico. Conteniendo vínculos a las publicaciones, a bases de datos de compuestos, vocabulario controlado, etc. Cuya funcionalidad principal es la búsqueda avanzada de modelos atendiendo a algún criterio. Cada modelo es comisariado cuidadosamente para verificar que corresponde a la publicación de referencia.

Esta base de datos es, sin duda, una de las más populares entre los especialistas, por la gran cantidad de modelos que contiene y por las facilidades que brinda a la hora de exportar dichos modelos, soportando diferentes formatos como: SBML, CellML, SciLab, XPP-Aut, BioPAX, VCell y PDF. La desventaja fundamental de esta base de datos es que, además de ser estática, no almacena nada acerca de las simulaciones que se le pueden hacer a los modelos, a pesar de tener incorporado un simulador online (JWC (34)). Permite descargar los modelos almacenados, pero no por parte, es decir, no permite el acceso directo ni a las variables (valores), ni a los parámetros (valores), que contengan los modelos, ya que para acceder a este tipo de información habría que interpretar el fichero SBML, que es donde realmente se almacena toda la información referente a los modelos matemáticos.

Un acercamiento más puntual al objetivo del presente trabajo, se puede encontrar en el sitio de Biología de Sistemas Computacionales [<http://sys-bio.org/index.htm>]. Este sitio está dedicado a los aspectos computacionales de redes bioquímicas, su dinámica, función y evolución, ya sean metabólicos, señales o redes de genes, específicamente investiga todo lo referido a cómo las redes químicas se comportan dentro de los sistemas vivos. En el mismo se presenta una herramienta online [<http://sys-bio.org/sbwWiki/compare>] que permite realizar comparaciones con los resultados obtenidos de la realización de simulaciones a modelos matemáticos. Cada modelo es simulado utilizando los simuladores (COPASI, JSim, Jarnac y RoadRunner, Oscill8, SBML ODE Solver) y la herramienta es capaz de hacer una comparación estadística en cuanto a sus semejanzas y diferencias según un criterio dado. Esta ingeniosa alternativa representa sin duda un paso de avance en esta área, aunque presenta dos limitantes fundamentales, trabaja nada más con los modelos matemáticos procedentes de la base de datos BioModels, y lo más significativo, no almacena las simulaciones comparadas o generadas de los distintos

simuladores utilizados, provocando a su vez que se tenga que simular cada vez que se quiera realizar una comparación.

Una vez revisados los sistemas de almacenamiento de datos sobre sistemas biológicos existentes, así como sus características principales, se arribó a la conclusión de que no existen bases de datos personalizadas que almacenen los temas asociados a la simulación de modelos matemáticos representativos de sistemas biológicos; ni lo referente al análisis de dichas simulaciones, enfocándose todas las encontradas sólo en la gestión de modelos matemáticos mediante ficheros SBML.

1.5 Almacenamiento de información de manera consistente

Hoy en día los mecanismos para la recolección automática de datos y el desarrollo de la tecnología de bases de datos, han permitido que se puedan almacenar grandes cantidades de datos en bases de datos, almacenes de datos y otros repositorios de información. Se pueden encontrar varias clasificaciones de base de datos según la terminología con la que se desee trabajar.

Las formas para almacenar la información en la actualidad han cobrado diversos tipos: Procesadores de texto como Word, que permiten sólo búsqueda y ordenamientos simples; Plantillas de cálculos como Excel, que como los datos están en columnas independientes, se puede ordenar en formas más complejas (pero las búsquedas que permite son búsquedas simples) y la colección de registros, comúnmente llamada bases de datos, especializadas para almacenar y organizar la información, permitir estudios posteriores y analizar a fondo estos datos para así poder extraer conocimiento (35).

En la actualidad, por su gran y divergente uso, las bases de datos son clasificadas de diferentes maneras. Las más comunes son: según el contenido a almacenar, la variabilidad de los datos a almacenar y, de acuerdo a la forma de estructurar la información almacenada:

- Según el contenido a almacenar pueden ser: Bibliográficas, contienen un resumen o extracto de publicaciones, pero nunca el texto completo (autor, fecha de publicación, editorial, título, edición); Biológicas, las que almacenan información

biológica; y de texto plano, las que almacenan publicaciones originales. En este trabajo se van a estudiar las bases de datos biológicas.

- Según la variabilidad de los datos pueden ser dinámicas las que almacenan en tiempo de ejecución; o estáticas las que almacenan información histórica. El sistema a obtenerse trabajará con una base de datos dinámica que no se conoce cuantos datos se van a almacenar hasta el momento de salvar la información; y los resultados para una misma consulta o procedimiento difieren según los datos de entrada.
- De acuerdo a la forma de estructurar la información almacenada, se pueden clasificar en relacionales, jerárquicas, de red, orientada a objetos, documentales y deductivas. Como resultado a obtenerse, la base de datos es relacional, ya que en este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. Esto tiene la ventaja de que es más fácil de entender y de utilizar para cualquier usuario de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad, lográndose además, administrar la información.

1.6 Formas de acceso a bases de datos

En el diseño de una aplicación, una parte muy importante es la manera en la cual se acceden a los datos en la base de datos, determinar esta parte se convierte en un punto crítico para el futuro desarrollo.

Tradicionalmente, el acceso a las bases de datos se hace a través de drivers (ODBC, JDBC, etc.), directamente conectados a la base de datos mediante ejecuciones de sentencias SQL. Esta primera alternativa resulta útil para proyectos o arquitecturas con pocas clases de negocio, ya que el mantenimiento del código está altamente ligado a los cambios en el modelo de datos relacional de la base de datos, puesto que un mínimo cambio, implica la revisión de casi todo el código así como su compilación y nueva instalación en el cliente.

Una aproximación más avanzada sería la creación de unas clases de acceso a datos (**DAO** Data Acces Object). De esta manera la capa de negocio interactuaría con la capa

DAO y esta sería la encargada de realizar las operaciones sobre la base de datos; aunque sigue siendo un problema el mantenimiento de la misma así como su portabilidad. Lo único que se puede decir es que se tiene el código de transacciones encapsulado en las clases DAO.

Evidentemente se debe separar el código de las clases de negocio de la realización de las sentencias SQL, aquí es donde entra Hibernate, que no sería más que el puente entre la aplicación y la base de datos. Sus funciones van desde la ejecución de sentencias SQL a través de JDBC hasta la creación, modificación y eliminación de objetos persistentes.

El enfoque que se tomó para este trabajo, es una combinación de los tres enfoques anteriores, en donde se tiene a Hibernate como puente entre la aplicación y el patrón de acceso a los datos DAO. Hibernate va a utilizar el driver JDBC para realizar una conexión transparente a la base de datos. Y se va a conectar a las clases DAO para realizar todas las modificaciones y consultas a la base de datos.

1.7 Conclusiones del capítulo

En este capítulo se abordaron e identificaron temas centrales acerca de los procesos de modelado, de sistemas biológicos, simulación de estos modelos y el análisis de estas simulaciones. Para ello una vez caracterizados los citados procesos, se explicó los problemas existentes en cuanto al almacenamiento y recuperación de la información asociado a cada uno de ellos. La investigación arrojó los siguientes resultados:

- Existen diferentes técnicas de análisis de datos. Los resultados generados al aplicar las mismas, constituye una información que debe ser gestionada con el fin de apoyar el proceso de la toma de decisiones por parte de los especialistas.
- La base de datos más genérica en cuanto a la gestión de modelos matemáticos de sistemas biológicos es: BioModels.
- Los formatos más utilizados para al almacenamiento e intercambio de datos biológicos son: SBML y CellML.
- No se encontraron referencias a base de datos biológicas que almacenaran la información en estudio.

2 MATERIALES Y MÉTODOS

2.1 Introducción

En este capítulo se describen las herramientas y la metodología utilizada durante el desarrollo del trabajo, se presentan los elementos fundamentales en la arquitectura del software, así como, se especifica el proceso de almacenamiento de la información y sus características.

2.2 Metodología de desarrollo de software

2.2.1 OpenUP

OpenUP/Basic es un Framework de procesos de desarrollo de software de código abierto, que con el tiempo espera cubrir un amplio conjunto de necesidades en el campo del desarrollo de software. Permite un abordaje ágil al software en análisis con sólo proveer un conjunto simplificado de contenidos, fundamentalmente relacionados con orientación, productos de trabajo, roles, y tareas. Es un proceso interactivo de desarrollo de software simplificado, completo y extensible; 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 (36).

2.2.2 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en la generación automática de códigos a partir de diagramas y modelos, y otros tipos de tareas.

2.2.2.1 Visual Paradigm

Es una herramienta que soporta el lenguaje de modelado UML y permite:

- Dibujar todos los tipos de diagramas de clases.
- Generar código directo, inverso y desde diagramas.
- Generar informes para generación de documentación (expediente de proyecto).

- Realizar ingeniería directa e inversa, proceso ingenieril en el que se obtienen modelos conceptuales a partir de los artefactos software como código fuente, ejecutables, binarios y ficheros intermedios.
- Importación y exportación de ficheros XMI.
- Generación de objetos Java desde la base de datos, entre otras posibilidades.

2.2.3 Gestores de Bases de Datos

Los sistemas de gestión de bases de datos (SGBD) son un tipo específico de software, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos y ayudan 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.

- MySQL. El cual posee una licencia dual. Es un software propietario y está patrocinado por una empresa privada, pero existen dos versiones: una gratuita que sería equivalente a la edición "express" SQL server de Windows bajo licencia GPL y otra más completa de pago.
 - Es un gestor de muy rápida lectura cuando utiliza el motor no transaccional MyISAM, (37) pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos lo que hace a MySQL ideal para este tipo de aplicaciones.
 - Funciona sobre múltiples plataformas, incluyendo: Linux.
- PostgreSQL (<http://www.postgresql.org>): Es un software dirigido por una comunidad de desarrolladores y organizaciones comerciales. Algunas de sus principales características son:
 - Permite alta concurrencia.
 - Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
 - Los usuarios pueden crear sus propios tipos de datos, los que son por completo indexables.

Cada sistema tiene sus características, ventajas e inconvenientes, la elección de uno u otro sistema para gestionar la base de datos vendrá definida por las necesidades.

2.2.4 Lenguajes utilizados

2.2.4.1 Lenguaje de marcado

Un lenguaje de marcado o lenguaje de marcas, es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

XML (**eXtensible Markup Language**):

El XML es un meta-lenguaje que permite crear etiquetas adaptadas a las necesidades (de ahí lo de "extensible"). El estándar define cómo pueden ser esas etiquetas y qué se puede hacer con ellas. Es además especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

El lenguaje XML logra un equilibrio entre simplicidad y flexibilidad. Este proporciona una sintaxis superficial para documentos estructurados, pero que no impone restricciones semánticas sobre el significado de los mismos. XML permite suministrar, recibir y procesar información en la Web y puede interoperar tanto con SGML como con HTML. Permite enlaces multidireccionales (esto es, que apunta a varios documentos) (38) (39).

2.2.4.2 Lenguaje Java

Java es un lenguaje moderno, de alto nivel, que recoge los elementos de programación que típicamente se encuentran en todos los lenguajes de programación, permitiendo la realización de programas profesionales (40).

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es la capacidad de que su código funcione sobre cualquier plataforma de software y hardware. Esto significa que el mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. (41) Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras.

2.2.4.3 Lenguaje de Consulta Estructurado (SQL)

El Lenguaje de consulta estructurado (SQL, del inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Es utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos, de una forma sencilla (42).

2.3 Herramientas de desarrollo

Las herramientas de desarrollo son aquellos programas o aplicaciones utilizadas en el desarrollo de un programa (programación), permitiendo el soporte al desarrollo e implantación de los mismos. Pueden ser de importancia vital (como un ensamblador, un compilador o un editor) o de importancia secundaria, como una IDE (Interfaz de Desarrollo Estructurada).

2.3.1 Netbeans

NetBeans es una plataforma para el desarrollo de aplicaciones usando el lenguaje Java y un entorno de desarrollo integrado (IDE). Es un programa con licencia CDDL que permite su uso libremente, brinda el código del mismo y es totalmente gratuito. Permite escribir, compilar, debugear, ensamblar y desplegar aplicaciones, brindado soporte para toda clase de lenguajes de programación a pesar de estar escrito en Java.

Posee un editor de código con sistema de coloración de la sintaxis y de auto-completado, anotaciones, macros e identificación automática del código, soporte para Java, C, C++, XML y HTML, así como para JSP, RMI, CORBA, JINI, JDBC y Servlet. Y por último, para los sistemas de control de versión Ant y CVS. Además dispone de herramientas de concepción visual para crear y manipular componentes visuales, junto a un ejército de asistentes y utilidades de gestión y generación de código; y brinda soporte para creación de plugins y servicios web (43).

Netbeans cuenta también con un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las API de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados

independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

2.3.2 Hibernate

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Las siglas mapeo objeto/relacional (ORM Object/Relational Mapping) hace referencia a técnicas de mapeo de representación de datos desde un modelo de objetos a un modelo relacional a través de un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas automáticamente (conocida como "criteria") con un SQL estándar (44).

Hibernate elimina la vieja forma de conexión JDBC (muy dependiente de la estructura de los datos) por el uso de objetos, a través de un motor de persistencia, que es el componente de software encargado de traducir entre objetos y registros. Es de código abierto, y permite guardar un objeto en la base de datos simplemente con `session.save(miObjeto)` o borrarlo con `session.delete(miObjeto)`; y examinar y manipular un objeto en ejecución. Para ellos cuenta con un archivo `properties` (`hibernate.properties`) o un archivo `xml` (`hibernate.cfg.xml`) para la configuración, una serie de JavaBeans que son las clases a persistir y en las que cada campo se asociará con una columna de la base de datos, y un archivo `xml` por cada una de estas clases (`NombreClase.hbm.xml`) que indica el mapping entre objetos y relaciones. Permite además generar base de datos en cualquiera de los entornos soportados: Oracle, DB2, MySQL, PostgreSQL, etc.

Con la creación de la capa de persistencia se consigue que los desarrolladores no necesiten conocer nada acerca del esquema utilizado en la base de datos. Tan solo conocerán el interface proporcionado por el motor de persistencia. De esta manera conseguimos separar de manera clara y definida, la lógica de negocios de la aplicación con el diseño de la base de datos. Esta arquitectura conllevará un proceso de desarrollo más costoso pero una vez se encuentre implementada las ventajas que conlleva merecerán la pena (44).

2.3.2.1 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.

Hibernate le permite a la aplicación, manipular los datos mediante objetos, con todas las características de la programación orientada a objeto. Esta herramienta que 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. Esta alternativa reduce de forma significativa el tiempo de desarrollo, que de otra forma se gastaría, manejando los datos de forma manual con SQL, encargándose así de alrededor del 95% de las tareas comunes relacionadas con la persistencia de datos. Esto facilita que si se cambia el manejador de base de datos no será necesario modificar toda la programación en SQL, sólo 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 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.

Existen otras tecnologías que permiten la manipulación de bases de datos. Sin embargo Hibernate sobresale entre ellas por las bondades que ofrece en cuanto al trabajo con los datos almacenados. Los sistemas que operan con información biológica generan un gran volumen de datos; por tanto la selección de una herramienta como Hibernate que permita el acceso y manipulación de estos datos es una característica a tener en cuenta para su selección.

2.3.3 Apache JMeter

JMeter es una herramienta de carga para llevar a cabo 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 (45).

2.4 Arquitectura de software

2.4.1 Web service (WS)

El World Wide Web Consortium (W3C) (<http://www.w3.org/>), organización dedicada a la definición de los estándares Web, establece: Un WS es un sistema de software identificado por un Localizador Uniforme de Recursos (URI), basado en tecnología como: HTTP, XML, SOAP, WSDL, SPARQL, entre otras, que permite la transferencia de datos de un dominio a otro dominio o entre aplicaciones, estos datos son normalmente definidos como vocabularios y gramáticas.

Caudwell Patrick, autor del libro "Profesional Servicios Web Xml" define el WS como: "Aplicaciones modulares autodescriptivas que se pueden publicar, ubicar e invocar desde cualquier punto de la Web o desde el interior de una red local basada en estándares abiertos de Internet".

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos (46).

Entre ellas es común la definición de protocolos y estándares, para intercambiar datos a través de la red por un protocolo estándar y con un formato específico de datos (XML).

Razones para crear servicios Web

La principal razón para usar servicios Web es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.

Una segunda razón se basa en el uso de HTTP sobre TCP (Transmission Control Protocol) en el puerto 80. Dado que las organizaciones protegen sus redes mediante *firewalls* -que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores.

Una tercera razón es el acceso a los datos según las necesidades personales, de proyectos o empresas y prestaciones a terceros, sin tener que duplicar información, ni las funciones implementadas para acceder a ella (reutilización de código) (47).

Y la última razón, es para estandarizar los servicios ofrecidos según estándares en Bioinformática, para poder intercambiar información y acceder a ella desde distintos lugares del mundo.

WSDL

Una vez implementado el Web Service, para que otras aplicaciones lo puedan ver, ellas deben acceder a un documento WSDL, en donde se podrán conocer los métodos que expone el Web Service y como acceder a ellos (nombre de los métodos y tipo de parámetros para cada uno de ellos). En este caso el WSDL (Web Services Description Language) es un estándar utilizado para describir los mensajes SOAP y el intercambio de los mismos. Gracias a los archivos descritos en este lenguaje se pueden descubrir los servicios web y utilizarlos de acuerdo a sus características, para una especificación completa, se puede referir a <http://www.w3.org/TR/wsdl>.

SOAP

Una vez localizado el servicio, el estándar utilizado para su invocación es el SOAP (Simple Object Access Protocol), protocolo base de los Web Service, por su naturaleza orientada a objeto. Su especificación se encuentra en <http://www.w3.org/TR/SOAP>.

SOAP es un protocolo para el intercambio de información, basado en XML. Este protocolo necesita de otro protocolo para la transmisión de los datos (generalmente http).

2.4.2 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) (48).

Este patrón encapsula el acceso a la base de datos, de manera que cuando la capa lógica del negocio necesite interactuar con la base de datos, ésta utilice los métodos ofrecidos por DAO. De esta forma, a las clases persistentes no les preocupa la capa de negocios, es decir, no le interesa si los datos se guardan en una tabla MySQL, XML, Texto plano o se imprime, de esto se encarga el patrón DAO.

Por cada tabla de una base de datos relacional existirá una clase DAO de Interfaz y otra de programación. Esto consiste básicamente en una clase que es la que interactúa con la base de datos. Los métodos de esta clase dependen de la aplicación y de lo que se quiera hacer. Pero generalmente se implementan las 4 funciones básicas (crear, modificar, eliminar y actualizar).

Los DAOs son independientes del servidor de aplicaciones Java y exponen servicios para inserciones individuales y masivas, consultas de alto rendimiento con paginación,

ordenamiento y filtros, actualizaciones parciales o totales y eliminación de información. Tienen, además, implementados las mejores prácticas de programación con Java y JDBC, y están integrados con los framework para manejo de Excepciones y de Calidad de Servicio.

2.5 Conclusiones del capítulo

Para el desarrollo del trabajo se usó OpenUP/Basic como metodología de trabajo, con algunas adaptaciones debido a que no presenta el rol de desarrollador de bases de datos; Visual Paradigm como herramienta case para el modelado de los artefactos, Java como lenguaje de programación, Hibernate como framework de persistencia de datos, MySQL y Postgres como gestores de base de datos, por ser sistemas de administración de bases de datos, fuertes, seguros y además multiplataforma. XML como lenguaje de marcado, NetBeans como IDE de desarrollo y Apache para la construcción de los servicios web, así como la herramienta JMeter para las pruebas de carga y stress.

3 RESULTADOS

3.1 Análisis del proceso de almacenamiento de los datos persistentes

Lo primero que se debe almacenar en la base de datos, es la información relativa a los sistemas biológicos a estudiar, pues a partir de ella es que concibe todo el proceso.

El sistema debe ser capaz de:

1. Gestionar todo lo referente a los sistemas biológicos.

Cada sistema se compone por su identificador y nombre, y en campo sbml, que permite almacenar todos los datos de un sistema en formato SBML, por lo que es un XML, hasta este punto se garantiza además:

2. Importar y exportar el SBML.

Lográndose obtener una tabla con todos estos atributos.

De cada sistema biológico se obtiene el modelo que describe matemáticamente su comportamiento. Los modelos están compuestos por arreglos de condiciones iniciales y parámetros. Cada modelo contiene un identificador, el identificador del sistema al que pertenece (cada sistema puede tener un conjunto de modelos matemáticos asociados), su nombre, la ecuación diferencial que define dicho modelo en formato XML así como, un fichero con el código de la ecuación matemática por cada programa que permite ejecutar dicha ecuación, que puede ser: java, Matlab y Octave. La tabla modelo a su vez, está relacionado con las tablas que almacenan la información de las condiciones iniciales y los parámetros, aquí se debe garantizar:

3. Almacenar el XML con la ecuación del modelo matemático que representa al sistema.
4. Almacenar los ficheros de ejecución del modelo para cada uno de sus respectivos formatos.
5. Almacenar las condiciones iniciales y los parámetros y sus respectivos valores.

En este punto vale destacar que, a diferencia de la base de datos BioModels, la información referente a los modelos matemáticos, aquí se almacena separada, es decir, en un XML el conjunto de ecuaciones diferenciales y en 4 tablas relacionales toda la información referente a las condiciones iniciales, los parámetros y los valores de ambos.

Esta nueva variante de almacenamiento, posibilita un acceso más rápido, pues no hay necesidad de interpretar el fichero XML para obtener las condiciones iniciales ni los parámetros que participan en el SED.

El tercer requerimiento se resuelve relacionando la tabla sistema con la de modelos, un sistema puede tener muchos modelos que lo describen. La tabla modelo guarda todos los datos descritos anteriormente y se relaciona además, con las tablas que almacenan las condiciones iniciales y los parámetros respectivamente, cumpliendo así con el quinto requerimiento. El cuarto requerimiento, permite que la ejecución de modelo matemático sea más rápida, pues se almacena ya el método con el código que es reconocido por cada uno de los programas que se utilizan.

Al tener los modelos matemáticos almacenados, ya se pueden simular, para luego almacenar dichas simulaciones. Las simulaciones pueden ser simples o múltiples y en cada una de ellas hay que especificar los valores a tomar por parte de las condiciones iniciales y los parámetros que intervienen, se definen además, cuales de estos se quieren variar, designando un valor final y la cantidad de puntos a variar dentro de dicho intervalo. Para la realización de estas simulaciones hace falta definir el método numérico a aplicar y el programa al que pertenece este método, por lo que el sistema debe garantizar en esta parte del trabajo:

6. Que se almacene las simulaciones realizadas.
7. Que se almacenen los valores fijados para las condiciones iniciales y los parámetros.
8. Que se registre el método numérico utilizado para resolver las simulaciones, así como el programa al que pertenece.

El requerimiento seis y siete se cumple, teniendo una tabla donde se almacenan las simulaciones realizadas con su identificador, un campo que indica el modelo que se simuló y otro con el identificador del método numérico utilizado para resolver dicho modelo, para este último se tiene una tabla con el algoritmo que se empleó, el programa al que pertenece y los valores de tolerancia absoluta y relativa del método en cuestión.

Junto con el almacenamiento de las simulaciones, se registra además, los resultados que éstas arrojan, por lo que el sistema debe permitir:

9. Que se almacene los resultados de las simulaciones realizadas.

Una vez almacenada las simulaciones realizadas se le pueden aplicar diferentes técnicas de análisis y almacenar los resultados de estas últimas, permitiendo tener datos e información que favorezca la toma efectiva de decisiones y la realización de predicciones sobre los estudios realizados. En las simulaciones se pueden utilizar diferentes técnicas de análisis (clustering, regla, manual, Weka), los datos generados de este análisis son almacenados en tablas destinadas para cada caso, guardando además, los parámetros utilizados por cada técnica, por lo que el sistema debe ser capaz de:

10. Almacenar qué técnica se aplicó a qué simulación.
11. Almacenar los parámetros ingresados por cada técnica utilizada.
12. Almacenar los datos generados de la aplicación de la técnica escogida.
13. Permitir exportar los datos generados de la aplicación de la técnica mediante un fichero.

Para darle cumplimiento a los requerimientos nueve y diez, se creó una tabla por cada técnica, agrupando en cada una de ellas sus respectivos parámetros, almacenando además la fecha en la que se realizó cada clasificación, así como el tiempo consumido por cada análisis. Para el almacenamiento de los resultados del análisis, requerimiento once, se creó una tabla donde se guarda en un campo de tipo Blob dicho resultado, que constituye además el contenido del fichero que se exporta. El sistema brinda la posibilidad de almacenar este fichero en la máquina donde se esté trabajando en el instante de generar estos resultados.

El sistema brinda además, la posibilidad de crear nuevas ecuaciones agrupadas por categorías para el trabajo con el editor de ecuaciones, brindando la facilidad de reutilizar elementos de un ecuación previamente almacenada durante la edición de un SED. En esta parte del trabajo el sistema debe permitir:

14. Que se almacenen las categorías creadas así como sus nuevos elementos.

Para establecer las relaciones entre cada uno de los datos de la base de datos se emplearon las reglas de transformación propuestas en los tres tomos del Date y el libro de pregrado de Bases de datos de la Dr. Rosa María Matos.

3.1.1 Análisis del proceso de recuperación de la información almacenada

Para recuperar la información almacenada, todas las consultas básicas estarán implementadas en la capa de acceso, aprovechando las facilidades que brindan el framework de persistencia hibernate y el patrón DAO, en accesibilidad, rapidez y transparencia.

En el caso de las tablas con atributos corrientes (enteros, decimales, cadenas, binarios) se resuelven con consultas a la base de datos y en el caso de los ficheros, lo que se almacena en la base es el cuerpo del mismo en un atributo texto, y para recuperarlo, se realiza la operación inversa, se crea el fichero y se le adiciona como cuerpo lo que está en el atributo texto.

3.2 Diseño de la solución

Una vez identificados los requerimientos del sistema, y las posibles soluciones a cada uno de ellos, se procedió al diseño de la base de datos basada en la descripción del epígrafe 3.1. (Ver figura 3)

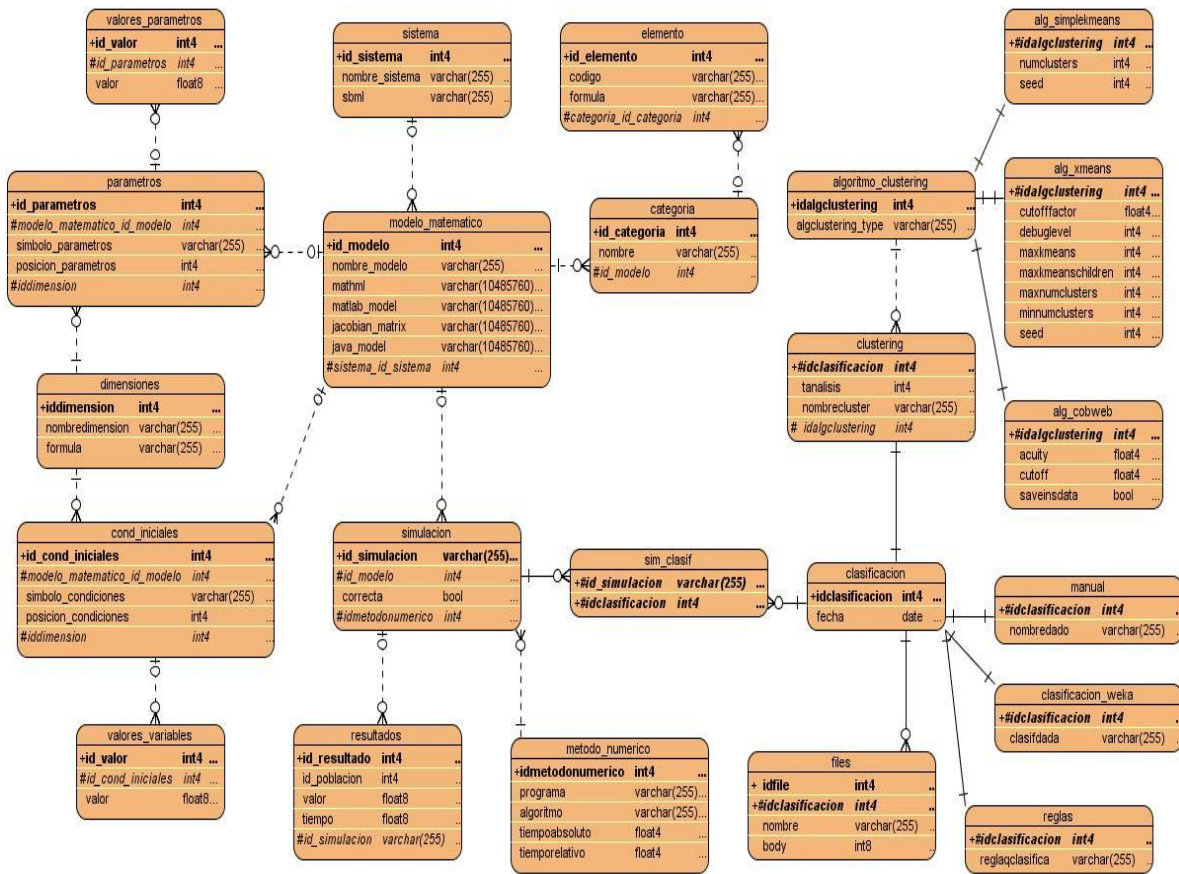


Figura 3. Modelo físico de la BD

3.3 Implementación del sistema

Después de diseñado el modelo físico, se procedió a implementar el sistema. Para ello en este trabajo se propone una arquitectura orientada a servicios como se muestra en la figura 4.



Figura 4. Interacción entre los componentes del sistema con los servicios implementados

3.3.1 Implementación de la capa de acceso a datos

El empleo del framework hibernate, posibilita la obtención de las clases persistentes y ficheros de mapeo por cada tabla y se confeccionaron las clases interface e implementación del patrón DAO. Se creó la clase “BSConnectionFactory” para garantizar

la conexión con la base de datos. Un diagrama de los componentes del sistema por paquetes, quedaría de la siguiente manera (figura 5):

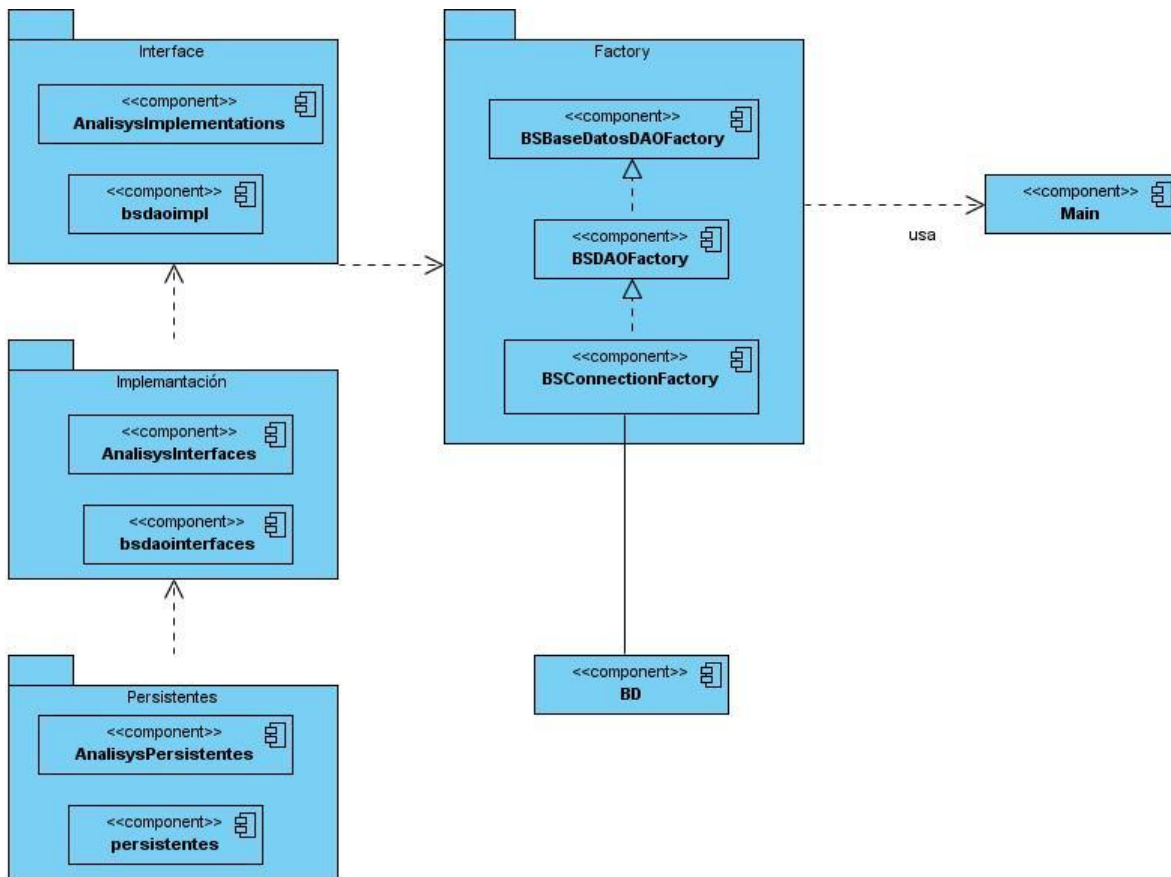


Figura 5. Interacción entre los componentes del sistema con los servicios implementados

En donde el paquete de Interface va a contener todas las clases interface de cada una de las persistentes almacenadas. El de Implementación va a contener la programación de las Interface (una por cada clase Interface) y el paquete Persistentes va a contener las tablas persistentes de la base de datos.

3.3.2 Implementación del Servicio Web

Para la implementación del Web Service se tuvo en cuenta la creación de un cliente y la del servidor: En la parte del servidor primero se crea el servidor Web y luego las operaciones o servicios que va a brindar. Luego se pone en marcha y él publica la dirección del documento WSLD [Sección 2.4.1].

En el Web Service se implementaron todas las funciones básicas de inserción y selección de datos. Brindando hasta el momento 52 operaciones sobre la base de datos a través de la dirección <http://10.8.23.110:8084/WebAppBioSyS/WebAppBioSyS?wsdl> del documento WSLD. A continuación se muestra un ejemplo de uno de los servicios web.

```
/**
 * Web service operation
 */
@WebMethod(operationName = "InsertarSistema")
@Oneway
public void InsertarSistema(@WebParam(name = "sistema")
Sistema sistema) {
    try {
        b.getSistemaDAO().insertarSistema(sistema);
    } catch (Exception ex) {
        Logger.getLogger(WSBioSyS.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

El resto de las funciones se encuentran anexadas al siguiente trabajo en el documento Javadoc generado por Netbeans.

Para tener acceso a las funciones publicadas en el servidor, el cliente debe identificar la dirección descrita en el documento WSLD del Web Service, para luego invocarlas según sea necesario. En el caso de la función insertar estudio, sería de la siguiente manera:

```
try { // Call Web Service Operation
    src.WSBioSySService service = new src.WSBioSySService();
    src.WSBioSyS port = service.getWSBioSySPort();
    // TODO initialize WS operation arguments here
```



```
src.Sistema sistema = new src.Sistema();  
    //Recogida de datos desde la interfaz  
    port.insertarSistema(sistema); //Uso del método pasándole los datos recogidos  
} catch (Exception ex) {  
    // TODO handle custom exceptions here  
}  
}
```

3.4 Conclusiones del capítulo

Como resultados de este capítulo, se aplicaron varias técnicas de almacenamiento y recuperación de la información de los procesos de modelado, simulación y análisis. Se implementó la base de datos en cuestión, y la capa de acceso a datos a la misma. Se implementó además una arquitectura basada en servicios para que sea accesible desde cualquier parte de la red y aprovechar los aportes que pudiera incorporar la comunidad científica.

4 VALIDACIÓN

4.1 Introducción

Finalizado ya el sistema, se pasa a la fase de comprobar que tan buena resultaron las decisiones tomadas. Para ello se diseñaron varios planes de pruebas con el objetivo de evaluar la funcionalidad, confiabilidad y desempeño de la base de datos

4.2 Pruebas de volumen

La prueba de volumen somete al sistema a grandes cantidades de datos, para determinar si se alcanzan los límites que lo hacen fallar.

Para realizar la prueba de volumen se pobló –con la herramienta EMS Data Generator 2005 para MySQL y para PostgreSQL -cada tabla con cantidades de registros variando desde 100, 200, 400, 1000 y hasta 5000 datos por tabla. Los tiempos de procesamiento del sistema y los porcentos de errores –para cada una de las pruebas- se muestran en la tabla 3:

Datos x tabla	PostgreSQL		MySQL	
	% Error	TGeneral (s)	% Error	TGeneral (s)
100	11,17	0:01:03	11,13	0:03:20
200	11.06	0:01:19	10.86	0:06:28
400	11.18	0:02:53	11.14	0:12:54
1000	11.31	0:07:18	11.26	0:35:21

Tabla 3. Resultados de los programas Data Generators para MySQL y PostgreSQL

En la figura 6, se muestra una gráfica comparativa sobre el comportamiento de las métricas de datos insertados contra el tiempo, entre ambos gestores, para una mejor visión del impacto de la simultaneidad de datos soportados.

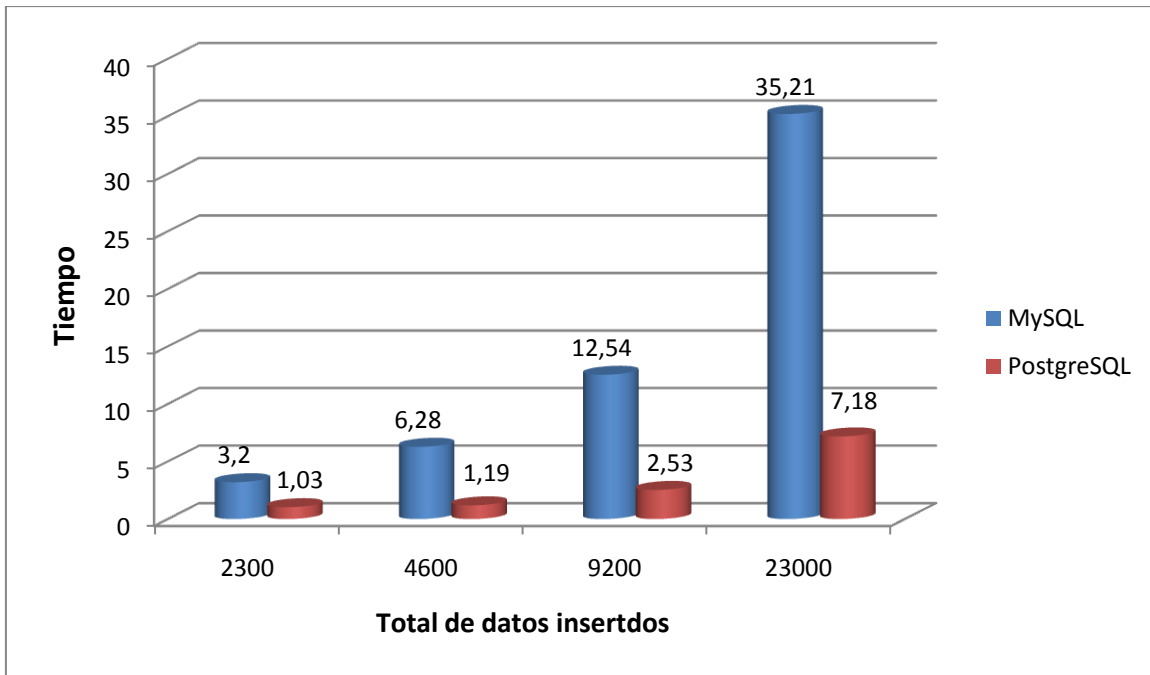


Figura 6. Comparación entre el tiempo consumido en cada operación por lo dos gestores

Los resultados obtenidos, demuestran un rápido procesamiento de PostgreSQL, evidenciando las potencialidades y rapidez de este gestor frente a MySQL y ambos revelaron un ligero porcentaje de errores. Los errores reportados en las pruebas fueron del tipo:

- Error: ERROR: duplicate key value violates unique constraint "NomRestricción"

Estos errores de llaves duplicadas van a estar determinados por el carácter aleatorio de los programas Data Generators y las características particulares de las tablas en las que se reportan, ya que son tablas que heredan su identificador de las entidades padres, y el programa ha insertado valores duplicados para una misma entidad hija.

Independientemente de los errores detectados, se puede apreciar que el sistema detecta correctamente las restricciones que se implementaron; y por otra parte, en tiempo real de trabajo, estos errores no deberían darse, ya que a cada dato procesado se le incorpora un identificador, el investigador al trabajar con el sistema inserta el identificador de la tupla.

4.3 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.

Estas pruebas a la base de datos se van a realizar mediante la simulación de concurrencia, conectándose un número determinado de usuarios a la misma. 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.

Planes de Pruebas

El primer plan de prueba (plan1), sería un muestreo de los datos, con consultas básicas de selección de datos, tres consultas de complejidad simple, media y alta respectivamente. El segundo plan de pruebas (plan2), con 16 consultas de selección también, donde varía la complejidad. (Figura 7)

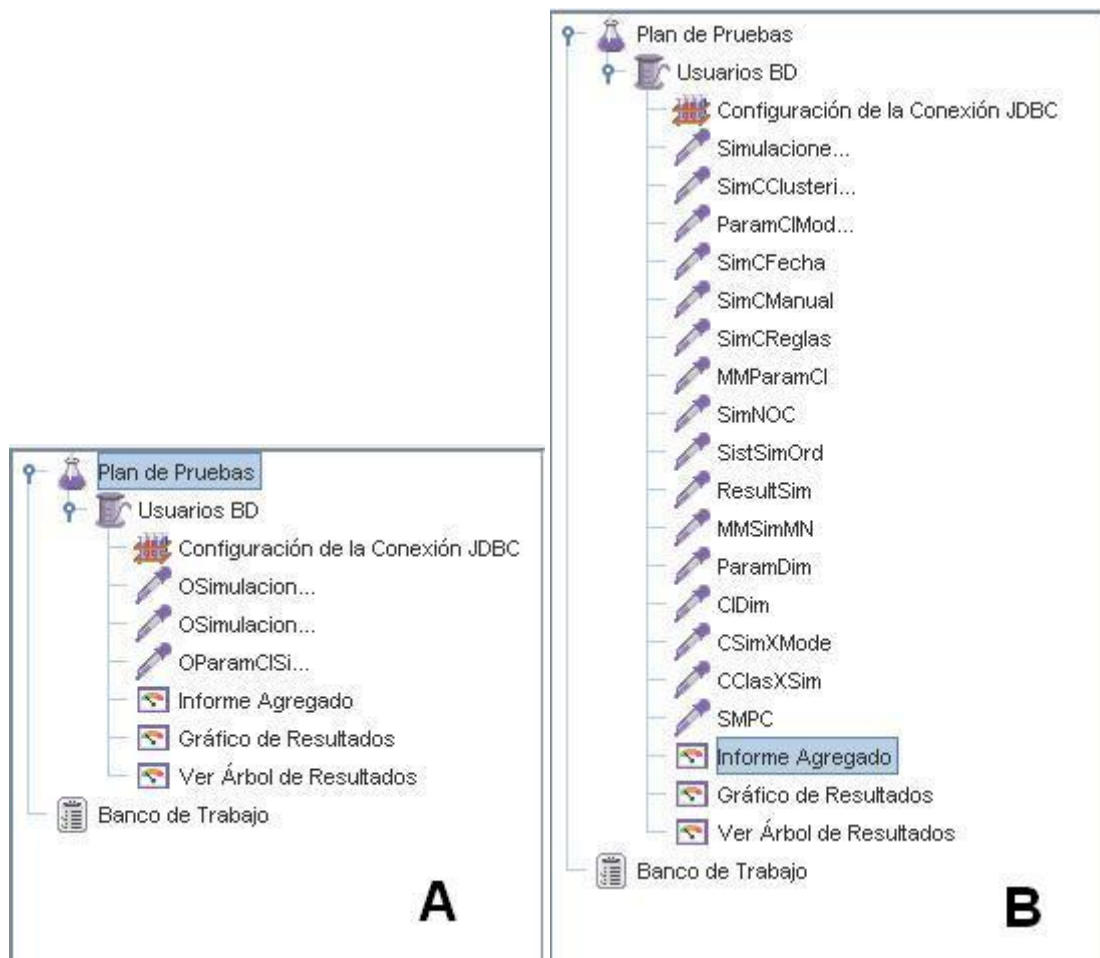


Figura 7. Configuración de los planes de prueba: A) Plan de prueba 1, B) Plan de prueba 2

Para realizar la prueba de carga, se utilizó el programa JMeter, con el objetivo de verificar la respuesta del sistema más allá de la carga de trabajo esperada. En ambos planes de pruebas se trabajó con un nivel de concurrencia de 30 usuarios, con un contador de bucle igual a 5. En el caso de la plan1 se realizaron 450 peticiones a la base de datos y en el plan2 2400 peticiones. El cumplimiento exitoso de estas pruebas, es determinar el límite de carga que soporta el sistema dentro de los tiempos aceptables para el usuario.

De cada prueba se tomaron los tiempos de respuestas y algunos datos estadísticos que proporciona el JMeter en el informe agregado [ver figura 8 y 9]. Informe que contiene por las filas, cada una de las solicitudes realizadas al servidor y por las columnas los elementos medidos en la prueba. Además se trabajó con el árbol de resultados (otro

componente del Jmeter) para identificar los errores detectados por el programa, o las salidas de las consultas en caso de ejecutarse correctamente.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Usuarios BD:OSimulaciones	150	108	94	240	1	546	0,00%	40,3/sec	30,7
Usuarios BD:OSimulacionCCLustering	150	121	80	326	5	572	0,00%	39,9/sec	294,3
Usuarios BD:OParamCISimCCLustering	150	160	105	361	18	585	0,00%	39,7/sec	790,4
TOTAL	450	130	94	331	1	585	0,00%	118,7/sec	1110,4

Figura 8. Resultados del informe agregado para el plan de prueba 1 con 30 hilos concurrentes

Para esta prueba, el tiempo promedio de ejecución de una consulta es de 130 ms, realizándose un total de 450 solicitudes al servidor sin errores, lo que significa que todas las consultas fueron ejecutadas correctamente. Además se puede observar un rendimiento de 118,7 consultas por segundos y un tiempo máximo de ejecución de consulta que no se acerca al segundo. Por lo que se puede afirmar que el tiempo de respuesta del sistema es rápido para 450 solicitudes que le realicen las 30 máquinas que se esperan se conecten a él.

0 / 30

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Display Only: Escribir en Log Sólo Errores Successes

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Simulaciones1Modelo	150	255	217	587	1	909	0,00%	4,2/sec	3,2
SimCClustering	150	257	185	604	5	1013	0,00%	4,2/sec	30,8
ParamCIModCC	150	329	265	708	22	1321	0,00%	4,2/sec	82,9
SimCFecha	150	381	337	729	21	1669	0,00%	4,1/sec	57,9
SimCManual	150	310	250	646	8	1243	0,00%	4,1/sec	35,9
SimCReglas	150	295	259	645	5	1451	0,00%	4,1/sec	35,4
MMParmCI	150	454	413	780	58	1448	0,00%	4,1/sec	134,6
SimNOC	150	249	200	553	3	1001	0,00%	4,1/sec	15,3
SistSimOrd	150	342	263	715	13	1451	0,00%	4,1/sec	56,3
ResultSim	150	251	211	598	4	1262	0,00%	4,2/sec	19,7
MMSimMN	150	257	235	544	2	1505	0,00%	4,2/sec	27,0
ParamDim	150	512	470	977	36	1788	0,00%	4,2/sec	100,9
CIDim	150	438	403	755	46	1655	0,00%	4,2/sec	101,6
CSimXMode	150	374	343	740	29	1609	0,00%	4,2/sec	38,4
CClasXSim	150	310	266	631	7	1283	0,00%	4,3/sec	40,1
SMPC	150	738	658	1258	75	2336	0,00%	4,3/sec	187,4
TOTAL	2400	359	296	735	1	2336	0,00%	65,3/sec	943,8

Figura 9. Resultados del informe agregado para el plan de prueba 2 con 30 hilos concurrentes

Para esta prueba, el tiempo promedio de ejecución de una consulta es de 372 ms, realizándose un total de 2400 solicitudes al servidor sin errores, lo que significa que todas las consultas fueron ejecutadas correctamente. Además se puede observar un rendimiento de 65,3 consultas por segundos y un tiempo máximo de ejecución de consulta que no supera los 3 segundos. Por lo que se puede afirmar que el tiempo de respuesta del sistema es rápido para 2400 solicitudes que le realicen las 30 máquinas que se esperan se conecten a él.

Se siguieron realizando las mismas transacciones de carga de trabajo, pero para una cantidad variada de hilos concurrentes, obteniéndose los resultados del plan de prueba 1 en la tabla 4 y los resultados del plan de prueba 2 en la tabla 5:

Plan de prueba 1 (bucle=5)					
Hilos	Cantidad de solicitudes	Tiempo de Respuesta (ms)	% de Errores	Rendimiento (Consultas/seg)	Tiempo Máx. de una consulta (ms)
30	450	130	0	118,7	585
60	900	146	0	118,2	795
120	1800	420	0	116,7	4705

Tabla 4. Resultados generales del Informe Agregado de la prueba de cargas, para el plan de pruebas 1

Plan de prueba 2 (bucle=5)					
Hilos	Cantidad de solicitudes	Tiempo de Respuesta (ms)	% de Errores	Rendimiento (Consultas/seg)	Tiempo Máx. de una consulta
30	2400	359	0	65,3	2336
60	4800	688	0,21	64,0	5501
120	9600	888	1,15	37,8	13642

Tabla 5. Resultados generales del Informe Agregado de la prueba de cargas, para el plan de pruebas 2

Aquí puede observarse que el tiempo de respuesta del sistema para los planes de prueba 1 y 2, se comportan muy similares en cada una de las cargas medidas, con un promedio de respuestas menor al segundo en cada caso.

En el plan2, el volumen de datos a mostrar por cada transacción afecta grandemente el rendimiento del sistema a partir de 120 usuarios realizando 9600 solicitudes a la base de datos, el sistema realiza grandes picos en el uso de la memoria [ver figura 10]

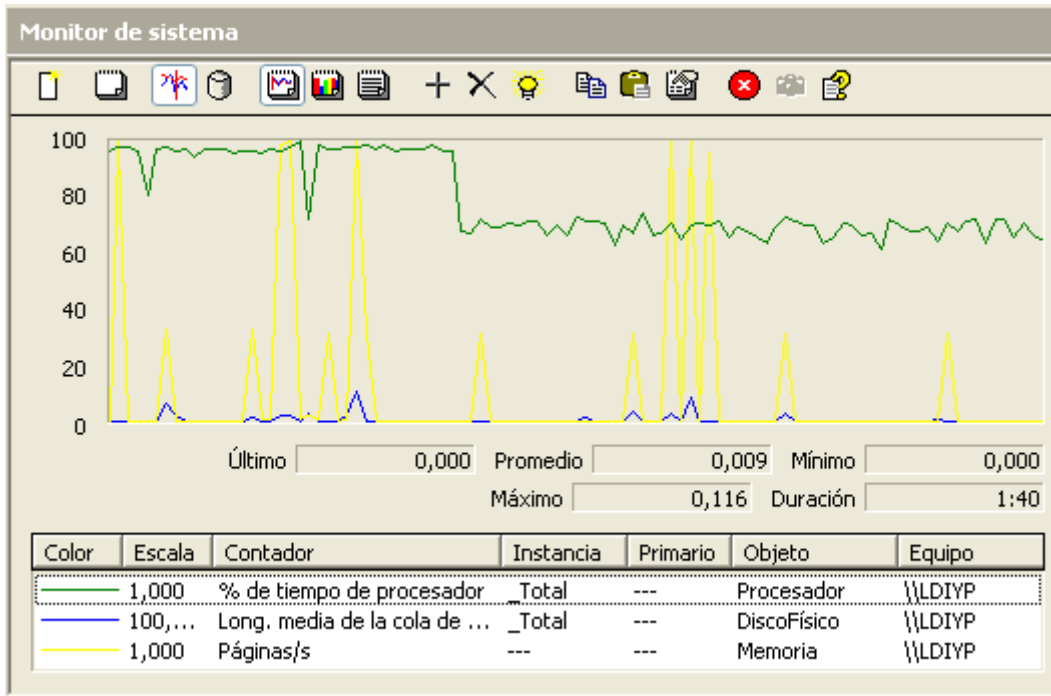


Figura 10. Monitor del sistema en el plan2 con 9600 solicitudes.

4.4 Prueba 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.

Para realizar la prueba de estrés, se utilizó el programa JMeter, con el objetivo de verificar los tiempos de respuesta del sistema cuando se tienen 120 o más usuarios interactuando con la base de datos al mismo tiempo. Se ejecutaron los mismos planes de prueba, realizados para la prueba de carga, pero con un nivel de concurrencia mayor (120 y 240 hilos). Al igual que las pruebas anteriores, los tiempos de respuestas son recogidos en el informe agregado [ver figura 11 y 12] y el árbol de resultados.

0 / 120

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo LogDisplay Only: Escribir en Log Sólo Errores Successes

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Usuarios BD:OSimulaciones	600	401	146	1259	0	4029	0,00%	39,2/sec	30,0
Usuarios BD:OSimulacionCclustering	600	384	132	1187	6	4705	0,00%	39,1/sec	288,6
Usuarios BD:OParamCISimCClustering	600	473	217	1267	21	4112	0,00%	38,9/sec	775,5
TOTAL	1800	420	167	1238	0	4705	0,00%	116,7/sec	1092,1

Figura 11. Resultados del informe agregado para el plan de prueba 1 con 120 hilos concurrentes

Para esta prueba, el tiempo promedio de ejecución de consultas es de 420, realizándose un total de 1800 solicitudes al servidor sin errores, lo que significa que todas las consultas fueron ejecutadas correctamente.

0 / 120

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo LogDisplay Only: Escribir en Log Sólo Errores Successes

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Simulaciones1Modelo	600	473	5	1360	1	6856	1,50%	2,4/sec	1,8
SimCClustering	600	511	17	1387	4	7599	0,50%	2,4/sec	17,6
ParamCIModCC	600	839	201	2335	4	7526	0,33%	2,4/sec	47,5
SimCFecha	600	1203	392	3825	15	10301	0,50%	2,4/sec	33,4
SimCManual	600	763	28	2977	6	6841	0,83%	2,4/sec	20,7
SimCReglas	600	582	15	2299	5	6769	0,50%	2,4/sec	20,6
MMPParamCI	600	1668	835	4498	44	10530	0,33%	2,4/sec	78,7
SimNOC	600	430	6	1549	2	6221	1,17%	2,4/sec	8,9
SistSimOrd	600	565	29	2207	11	11332	0,50%	2,4/sec	33,0
ResultSim	600	394	8	1370	3	6825	1,00%	2,5/sec	11,6
MMSimMN	600	399	7	1407	3	7200	1,00%	2,5/sec	15,8
ParamDim	600	1364	462	4260	8	7655	1,17%	2,5/sec	59,2
CIDim	600	1591	691	4645	6	13642	1,83%	2,5/sec	59,5
CSimXMode	600	800	68	3361	4	10747	4,33%	2,5/sec	21,7
CClasXSim	600	570	27	1844	5	6689	1,33%	2,5/sec	23,0
SMPC	600	2053	1472	5026	4	10794	1,50%	2,5/sec	107,0
TOTAL	9600	888	114	3127	1	13642	1,15%	37,8/sec	540,7

Figura 12. Resultados del informe agregado para el plan de prueba 2 con 120 hilos concurrentes

Para esta prueba, el tiempo promedio de ejecución de consultas es de 37,8, realizándose un total de 9600 solicitudes al servidor un por ciento de errores muy bajo (1,15). Para los informes de las pruebas con 240 usuarios concurrentemente, ver Anexo 1 y 2. Para el monitoreo del sistema durante la ejecución de las pruebas para 240 usuarios simultáneamente, ver Anexo 3, 4 a), b) y c).

4.5 Conclusiones del capítulo

Las pruebas realizadas a la base de datos demuestran que:

- El sistema no presenta errores de diseño y los tiempos de respuesta satisfacen los requerimientos deseados, lo que consolida la selección de las técnicas adoptadas en cada caso.
- Las pruebas de carga y estrés demuestran que a medida que aumenta el número de usuarios concurrentes, aumenta la limitación entre usuarios lo que, a su vez, provoca que aumente el tiempo de respuesta y descienda el rendimiento general del sistema.
- Los tiempos de respuestas del sistema son óptimos para 120 usuarios concurrentes haciendo 9600 solicitudes de datos. Por lo que se puede afirmar- según los límites que se proponen en el libro “Usability Engineering” (49) de Jakob Nielsen- que el sistema responde a las necesidades del cliente.

Estas pruebas permiten concluir que el sistema soportará la cantidad de usuarios requeridos y los tiempos de respuestas del sistema para esa cantidad de usuarios conectados con una carga de trabajo cinco veces por encima de lo normal, no llega al segundo.

CONCLUSIONES

- Se realizó una revisión bibliográfica sobre la temática en estudio, concluyendo que no existía ningún sistema con las características requeridas con respecto al almacenamiento de las simulaciones realizadas a los modelos matemáticos representativos de sistemas biológicos, así como el análisis de estas simulaciones.
- Se desarrolló la aplicación informática que permita el almacenamiento y recuperación de la información almacenada sobre los procesos de modelado, simulación y análisis de sistemas biológicos, que facilitará el trabajo de muchos investigadores y especialistas del área en cuestión.
- Se implementó una técnica nueva para el almacenamiento y recuperación de los modelos matemáticos.
- Se permite la importación y exportación de los modelos en formato SBML.
- Se realizaron pruebas que validan el correcto funcionamiento de la aplicación.
- Los tiempos de respuestas del sistema son óptimos para 120 usuarios concurrentes haciendo 9600 solicitudes de datos.

RECOMENDACIONES

- Hacer el sistema compatible a ficheros con formato CellML.
- Realizar las pruebas de funcionamiento en el ambiente real de trabajo, para los datos arrojados del proceso de análisis.

REFERENCIAS BIBLIOGRÁFICAS

1. Institute for cell Dynamics and Biotechnolgy. [En línea] [Citado el: 11 de junio de 2010.] <http://www.icdb.cl/ES/nosotros.php>.
2. **Lemus, Noel Moremo.** *BioSyS: Software para la simulación de Sistemas Biológicos.* Ciudad Habana : s.n., 2007.
3. JWS Online Cellular Systems Modelling. [En línea] [Citado el: 30 de mayo de 2010.] <http://jji.biochem.sun.ac.za/>.
4. Entelos. [En línea] [Citado el: 30 de mayo de 2010.] <http://www.entelos.com/index.php>.
5. Physiome. [En línea] [Citado el: 29 de mayo de 2010.] <http://www.physiome.com/>.
6. ExPASy. [En línea] [Citado el: 29 de mayo de 2010.] <http://www.expasy.org/links.html>.
7. **Trelles, Dr. Oswaldo.** Biología de Sistemas. Bioinformática Básica. [En línea] [Citado el: 1 de junio de 2010.] http://ocw.unia.es/ciencias-de-la-vida/bioinformatica-basica/materiales/bloque-4/06-SisBio1_Biologia%20de%20Sistemas.pdf.
8. Biología de Sistemas. Informe de Vigilancia Tecnológica. [En línea] [Citado el: 13 de junio de 2010.] http://www.gen-es.org/12_publicaciones/docs/pub_76_d.pdf.
9. **Posadas, Juan Pablo Amador.** Teoría General de Sistemas. [En línea] [Citado el: 11 de noviembre de 2009.] http://www.elprisma.com/apuntes/administracion_de_empresas/teoriageneraldesistemas.
10. Biotecnología computacional: Biología de Sistemas . [En línea] [Citado el: 11 de noviembre de 2009.] http://www.uco.es/master_nutricion/asignatura2.htm.
11. Introducción a las técnicas de modelado y simulación. [En línea] [Citado el: 3 de junio de 2010.] <http://www.scribd.com/doc/11314683/Simulacion-de-Sistemas>.
12. Introducción a la simulación. [En línea] [Citado el: 4 de junio de 2010.] http://www.material_simulacion.ucv.cl/INDICE.htm.
13. **Poblet, José Mompín.** *INTRODUCCIÓN A LA BIOINGENIERÍA.* s.l. : Marcombo, 1988.

14. **Canfrán, Javier García Cámara y Oscar Borja García.** Diagnóstico en Cardiología. [En línea] [Citado el: 1 de junio de 2010.] <http://www.it.uc3m.es/jvillena/irc/practicas/04-05/19mem.pdf>.
15. Método de Runge-Kutta. [En línea] [Citado el: 18 de junio de 2010.] http://www.itmorelia.edu.mx/electrica/Notas/Lino_Coria/Metodos_Numericos/METODOS_DE_RUNGE_KUTTA.pdf.
16. Ecuaciones Diferenciales. [En línea] [Citado el: 18 de junio de 2010.] <http://www.sc.ehu.es/sbweb/fisica/cursoJava/numerico/eDiferenciales/rungeKutta/rungeKutta.htm>.
17. Método multistep linear . [En línea] [Citado el: 18 de junio de 2010.] http://www.worldingo.com/ma/enwiki/es/Linear_multistep_method.
18. **Ramírez, Justino Alavez.** Estimación de Parámetros en Ecuaciones. [En línea] 2007. [Citado el: 19 de junio de 2010.] <http://www.red-mat.unam.mx/foro/volumenes/vol021/estimacion.pdf>.
19. Rosenbrock Method. [En línea] [Citado el: 19 de junio de 2010.] <http://www.applied-mathematics.net/optimization/rosenbrock.html>.
20. Ecuaciones Diferenciales Numéricas.Método de Euler . [En línea] [Citado el: 19 de junio de 2010.] <http://mate.uprh.edu/~pnm/notas4061/odes1/odes1.htm>.
21. DataEngine. [En línea] [Citado el: 19 de junio de 2010.] <http://www.idg.es/iworld/articulo.asp?id=146575>.
22. **Garrel, Migue y Sicilia, Juan José Cuadrado y Miguel Ángel.** Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. [En línea] [Citado el: 1 de junio de 2010.] <http://www.sc.ehu.es/jiwdocoj/remis/docs/GarreAdis05.pdf>.
23. Algoritmos de Clustering. [En línea] [Citado el: 20 de junio de 2010.] <http://www.scribd.com/doc/13735717/Clustering-Agrupamientos>.
24. Clustering Algorithms for Categorical Data Sets . [En línea] [Citado el: 1 de junio de 2010.] <http://csie.org/~dm/clustering.2.1107.ppt> .
25. X-means. [En línea] [Citado el: 20 de junio de 2010.] <http://www.cs.cmu.edu/~dpelleg/download/xmeans.pdf>.
26. Sitio de CellML. [En línea] [Citado el: 7 de junio de 2010.] <http://www.cellml.org/> .

27. MathML . [En línea] [Citado el: 20 de junio de 2010.] <http://www.w3.org/Math/>.
28. Sitio del SBML . [En línea] [Citado el: 2 de junio de 2010.] <http://sbml.org/> .
29. Swiss-Prot . [En línea] [Citado el: 20 de junio de 2010.] <http://www.expasy.org/sprot/>.
30. UniProt . [En línea] [Citado el: 20 de junio de 2010.] <http://www.uniprot.org/>.
31. Datos biológicos de Babel a BIANA. [En línea] [Citado el: 7 de junio de 2010.] <http://www.es.globaltalentnews.com/actualidad/reportajes/2515/Datos-biologicos-de-Babel-a-BIANA.html> .
32. Sitio de SABIO-RK. [En línea] [Citado el: 1 de junio de 2010.] <http://sabio.villabosch.de>.
33. **Oster, Robert Macey y George.** Berkeley Madonna . [En línea] [Citado el: 4 de junio de 2010.] <http://www.berkeleymadonna.com/> .
34. Sitio de BioModels. [En línea] [Citado el: 1 de junio de 2010.] <http://www.ebi.ac.uk/biomodels-main/> .
35. fBioinformática. Portal de Bioinformática de la Universidad de la Habana. [En línea] abril de 2005. [Citado el: 4 de junio de 2010.] <http://fbio.uh.cu/sites/bioinfo/index.html>.
36. Sitio Web de AddLink. [En línea] [Citado el: 31 de mayo de 2010.] <http://www.addlink.es/>.
37. Bases de datos MySQL. [En línea] [Citado el: 11 de junio de 2010.] http://www.bizboost.net/index.php?option=com_content&view=article&id=102:base-de-datos-mysql&catid=110:mysql&Itemid=363.
38. Lenguaje de etiquetas. [En línea] [Citado el: 11 de junio de 2010.] <http://www.traductores.org.ar/nuevo/files/adjuntos/1176838204.pdf>.
39. **ZAPATA, Inmaculada CEREZO GRAU y Francisco ALCANTARA.** APLICACIONES DE XML . [En línea] [Citado el: 11 de junio de 2010.] http://www.google.com/cu/search?hl=es&q=XML+es+un+meta-lenguaje+que+permite+crear+etiquetas+adaptadas+a+las+necesidades+&aq=f&aql=&oq=&gs_rfai=.
40. Lenguaje Java. [En línea] [Citado el: 11 de junio de 2010.] <http://www.lab.dit.upm.es/~lprg/asignatura/java.htm>.

41. Evolución Histórica de los lenguajes de programación. [En línea] [Citado el: 20 de junio de 2010.] <http://html.rincondelvago.com/evolucion-historica-de-los-lenguajes-de-programacion.html>.
42. SQL Structured Query Language. [En línea] [Citado el: 10 de junio de 2010.] <http://www.solodisenio.com/sql-structured-query-language/>.
43. NetBeans. [En línea] 2010. [Citado el: 13 de junio de 2010.] <http://www.guia-ubuntu.org/index.php?title=NetBeans>.
44. **González, Héctor Suárez.** *Manual Hibernate*. 2003.
45. JMeter. [En línea] octubre de 2005. [Citado el: 2 de junio de 2010.] <http://www.osmosislatina.com/jmeter/basico.htm> citado.
46. **Mata, Mirna Ariadna Muñoz.** Web Services. [En línea] 2004. [Citado el: 12 de junio de 2010.] http://www.etmk.cl/in72j/documentos/webservices/web_services.pdf.
47. **González, Carlos Daniel.** Curso introducción a los servicios web. [En línea] 2008. [Citado el: 7 de junio de 2010.] http://www.usabilidadweb.com.ar/servicios_web.php.
48. **JOSE LUIS MES, ANDRES RICARDO TORRES, CLAUDIA PATRICIA OVIEDO, JENNIFER ANDREA TENORIO.** PATRON DAO . [En línea] [Citado el: 2 de junio de 2010.] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/expoDAO.ppt .
49. **Nielsen, Jakob.** *Usability Engineering*. San Francisco : s.n. 0-12-518406-9.

ANEXOS

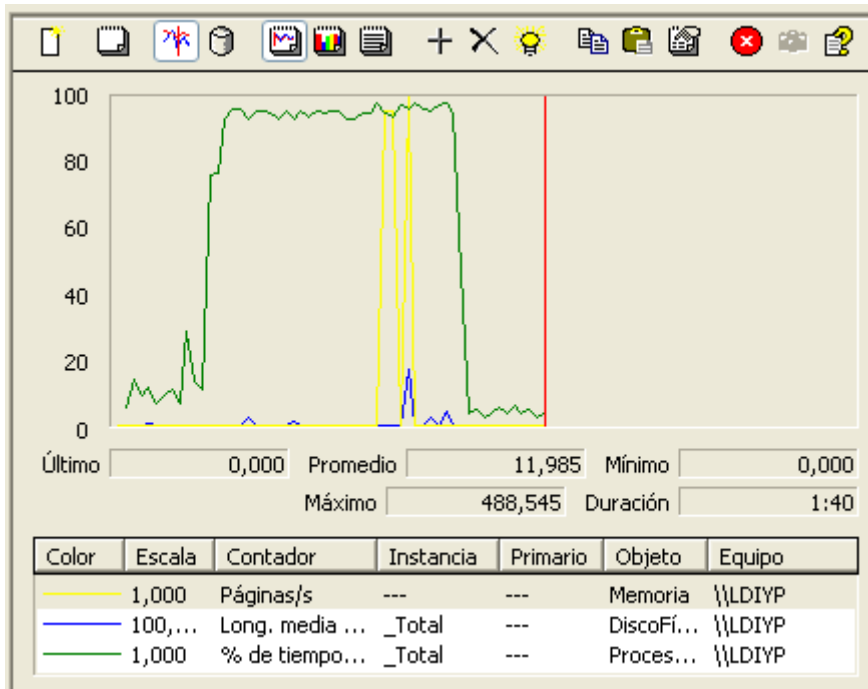
Anexo 1: Informe Agregado del Plan1 para 240 usuarios concurrentemente. (3600 peticiones a la base de datos)

0 / 240											
Informe Agregado											
Nombre: Informe Agregado											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo				Navegar...		Log/Display Only: <input type="checkbox"/>		Escribir en Log Sólo Errores <input type="checkbox"/>		Successes <input type="checkbox"/>	Config
Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec		
Usuarios BD:OSimulaciones	1200	932	288	2119	1	7879	3,33%	37,5/sec	27,7		
Usuarios BD:OSimulacionCClustering	1200	539	254	1345	3	6517	3,08%	37,2/sec	266,1		
Usuarios BD:OParamCISimCClustering	1200	666	323	1695	4	6510	1,83%	37,2/sec	726,8		
TOTAL	3600	712	287	1752	1	7879	2,75%	111,5/sec	1019,9		

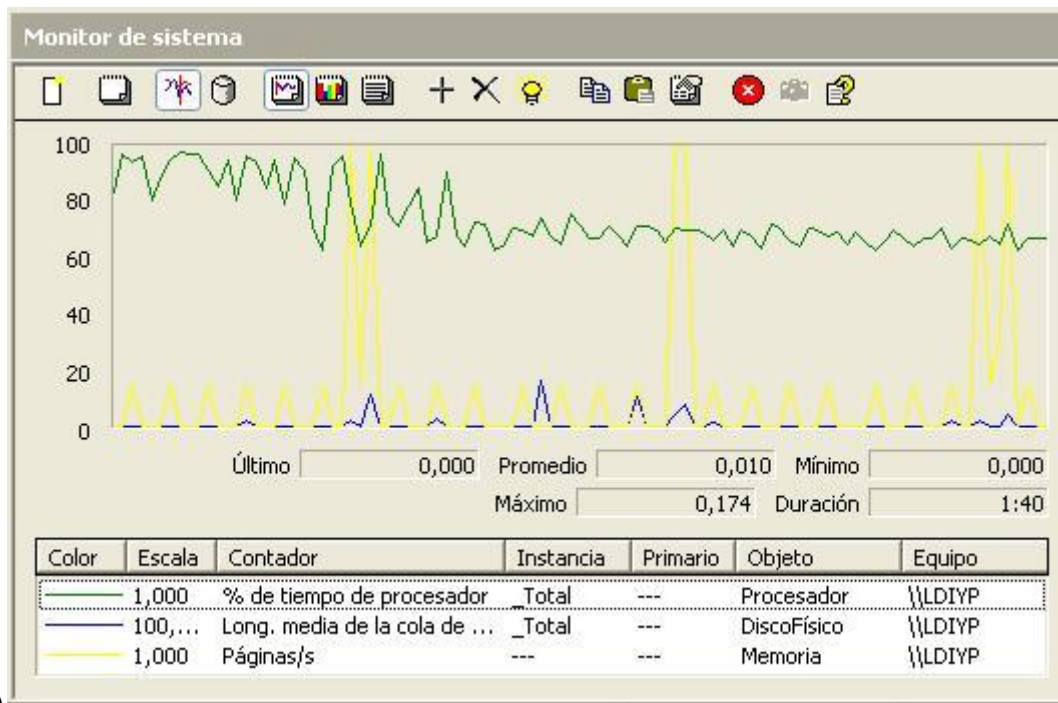
Anexo 2: Informe Agregado del Plan2 para 240 usuarios concurrentemente. (12983 peticiones a la base de datos)

0 / 240											
Informe Agregado											
Nombre: Informe Agregado											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo				Navegar...		Log/Display Only: <input type="checkbox"/>		Escribir en Log Sólo Errores <input type="checkbox"/>		Successes <input type="checkbox"/>	Config
Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec		
Simulaciones1Modelo	926	34179	842	10005	1	1738679	7,88%	9,4/min	0,1		
SimCClustering	910	31321	415	10004	3	1780022	8,88%	9,2/min	1,0		
ParamCIModCC	895	28613	1409	10092	3	1899044	6,15%	9,0/min	2,8		
SimCFecha	870	19283	2114	10469	3	1107952	4,60%	8,6/min	1,9		
SimCManual	860	23277	455	11149	4	1657569	4,53%	8,4/min	1,2		
SimCReglas	839	20203	117	10025	4	1523083	3,93%	8,1/min	1,1		
MMPParamCI	819	19281	3153	12902	16	1851744	3,05%	8,7/min	4,6		
SimNOC	816	23444	18	10018	3	1717690	2,21%	8,6/min	0,5		
SistSimOrd	802	15169	437	10124	10	1173120	2,49%	8,3/min	1,8		
ResultSim	792	21876	68	10021	3	1497390	3,03%	8,2/min	0,6		
MMSimMN	782	28256	13	9490	3	1622655	2,05%	8,0/min	0,8		
ParamDim	768	27735	2676	12839	38	1697302	2,21%	7,8/min	3,1		
CIDim	751	20302	2807	11743	37	1092622	1,46%	7,4/min	3,0		
CSimXMode	736	25761	472	10083	15	2098027	1,90%	7,2/min	1,1		
CClasXSim	723	30677	468	10054	6	1655648	2,21%	7,0/min	1,1		
SMPC	694	29521	5796	20473	166	1225793	2,31%	6,7/min	4,7		
TOTAL	12983	24970	1126	11040	1	2098027	3,82%	2,1/sec	28,3		

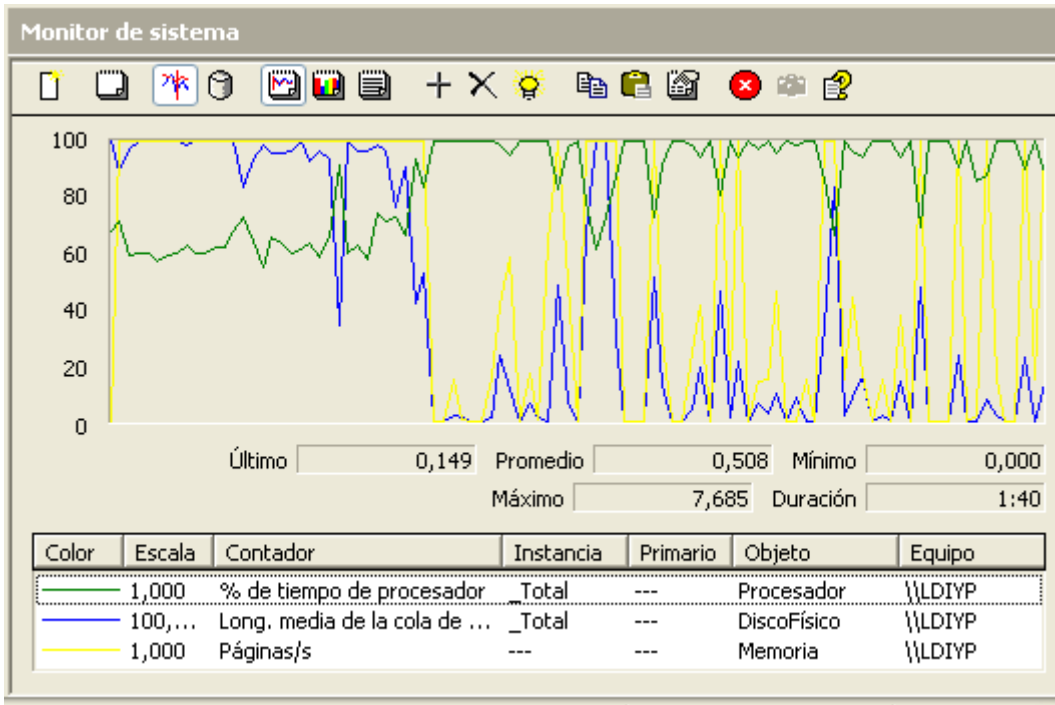
Anexo 3: Monitor del sistema para el Plan1 con 240 usuarios.



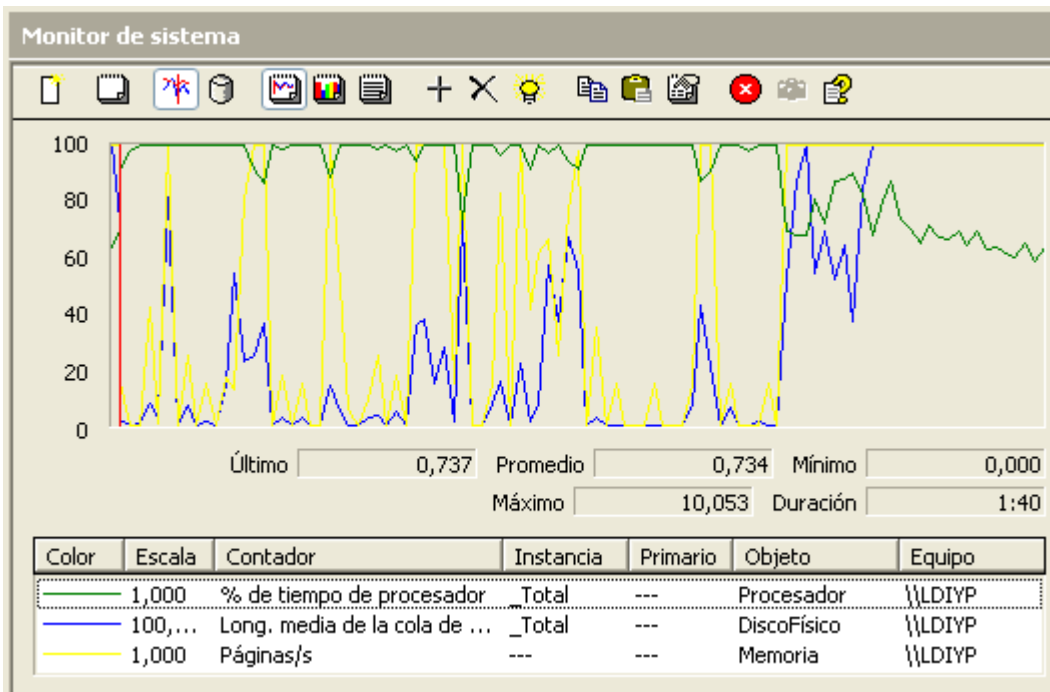
Anexo 4: Monitor del sistema para el Plan2 con 240 usuarios.



a)



b)



c)