Universidad de las Ciencias Informáticas

Facultad #6

Título: “BioSyS: Análisis de series temporales mediante técnicas de Clustering”

Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas

Autores: Yanet Fajardo Quesada
Liusmila Leyva Abreu
Tutores: M.Sc. Noel Moreno Lemus
Ing. Yunet González Mulet

Co-Tutor: Lic. Félix A. Martínez Nariño

Junio, 2009

"Si buscas resultados distintos, no hagas siempre lo mismo."

Albert Einstein

Declaración de autoría

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

Para que así conste firmo la presente a los ——— días del mes ———— del año ———— .

Liusmila Leyva Abreu

_____

Firma del Autor

Yanet Fajardo Quesada

_____

Firma del Autor

Yunet González Mulet

_____

Firma de Tutor

Noel Moreno Lemus

_____

Firma de Tutor

Datos del contacto

Tutores:

M.Sc. Noel Moreno Lemus.
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: noel@uci.cu

Ing. Yunet González Mulet.
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: ygonzalezmu@uci.cu

Co-Tutor:

Lic. Félix Argelio Martínez Nariño.
Universidad de las Ciencias Informáticas, Habana, Cuba.
Email: famartinez@uci.cu

Agradecimientos

A la Revolución que nos brindó la posibilidad de hacernos profesionales en una Universidad de excelencia.

A los tutores:

Yunet y Noel por su generosidad, dedicación y paciencia al brindarnos la oportunidad de recurrir a su capacidad y experiencia científica.

Y Félix por sus valiosas sugerencias y acertados aportes durante el desarrollo de este trabajo.

A Yenier por su colaboración desinteresada.

A los profesores que tuvieron que ver de una forma u otra con nuestra formación profesional.

A todos nuestros amigos pasados y presentes; pasados por ayudarnos a crecer y madurar como persona y presentes por estar siempre con nosotras apoyándonos en todas las circunstancias posibles.

No hay palabras que puedan describir nuestro profundo agradecimiento hacia nuestros padres, quienes nos han heredado el tesoro más valioso que puede dársele a un hijo: amor. A quienes sin escatimar esfuerzo alguno, han sacrificado gran parte de su vida para formarnos y educarnos. A quienes la ilusión de su vida ha sido convertirnos en personas de provecho. A quienes nunca podremos pagar todos sus desvelos ni aún con las riquezas más grandes del mundo.

Por esto y más… Gracias.

En general quisiéramos agradecer a todas y cada una de las personas que han vivido con nosotras la realización de esta tesis, gracias por habernos brindado todo el apoyo, colaboración, ánimo y sobre todo cariño y amistad.

Dedicatoria

A mi mamita, fiel amiga y consejera, gracias a su sacrificio estoy aquí en estos momentos. Serás siempre mi inspiración para alcanzar mis metas, por enseñarme que todo se aprende y que todo esfuerzo es al final recompensa. Tu esfuerzo, se convirtió en tu triunfo y el mío, TE AMO.

A mi abuelita, por guiarme sobre el camino de la educación.

A mis hermanos, que son el mejor regalo que me ha dado la vida.

A mis tíos, primos y demás familiares por siempre confiar en mí y apoyarme en todo momento.

A Pedro, por ser un como un padre para mí.

A Yanet, por su paciencia y dedicación, porque gracias a su ayuda estoy aquí hoy.

A mis hermanitas, Yoly, Linecilla, Dayo y Tata porque en su compañía las cosas malas se convierten en buenas, la tristeza se transforma en alegría y la soledad no existe.

Al Clan Moa, por compartir risas y llantos en todo este tiempo.

A mis amigos de la Universidad: Que me han apoyado, comprendido y aceptado durante estos 5 años y a los que nunca olvidaré.

Liusmila Leyva Abreu

A mi mamá, la mujer que me apoyó todos estos años, fiel amiga y consejera; por su infinito amor, cariño y comprensión. Por soportar todos estos años lejos de mí, por acompañarme en las buenas y las malas. Por ayudarme a que este momento llegara y que mi sueño y el suyo se hicieran realidad. Te quiero mucho!

A mi tía María, por su confianza en mí y su apoyo aunque lejos, pero muy cerca.

A mi tía Magalita, por su paciencia, por siempre tener palabras de aliento.

A mis primitos, Pedri y Ernesto, por quererme tanto.

A mi familia en general por la fuerza y el amor que han sabido brindarme a pesar de la distancia.

A mi novio Andy, por preocuparse más por la tesis que yo, por ser mi sostén y mi mejor amigo en esta escuela, por su amor y sobre todo por haber sido mi compañero incondicional.

A mis amigos y compañeros de estudios, por brindarme su amistad y su apoyo moral.

Y por último, quiero dedicar este momento tan importante e inolvidable a mí misma, por no dejarme vencer, ya que en ocasiones el principal obstáculo se encuentra dentro de uno...

Yanet Fajardo Quesada

Índice

Índice de figura

Índice de tablas

RESUMEN

BioSyS es un software concebido con el fin de realizar el análisis y la simulación de Sistemas Biológicos, desarrollado por la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI) y el Centro de Inmunología Molecular (CIM). Este cuenta con un Módulo de Análisis que es el encargado de brindar funcionalidades para el análisis de las simulaciones realizadas por el Módulo de Simulación y almacenándolos en una Base de Datos. En la versión 1.0 de este SW se implementaron diferentes funcionalidades, entre ellas se incluyó el análisis de los datos a través de la técnica de Clustering utilizando sólo el tiempo final de las simulaciones: únicamente se utilizaban los valores correspondientes a cada variable de la simulación de acuerdo al tiempo que el usuario definía como tiempo final, por lo que el análisis no era lo suficientemente profundo. En el presente trabajo con el objetivo de mejorar u obtener un resultado más profundo y específico se desarrolla e incluye a dicho módulo la funcionalidad de poder realizar el Clustering con toda la serie temporal, o sea, se utiliza la serie temporal con todos los valores correspondientes a cada variable de las simulaciones desde el tiempo inicial hasta el tiempo que el usuario define como tiempo final. Así el análisis por Clustering será más detallado y brindará mejor información al investigador.

Palabras Claves: Análisis, Algoritmos, Simulación, Minería de datos, Clustering, Series Temporales.

INTRODUCCIÓN

La Biología de Sistemas es un área relativamente nueva de la Biología que ha tenido un gran crecimiento e impacto en los últimos tiempos dada la gran cantidad de esferas en las que se puede aplicar. Esta nueva disciplina incluye conceptos y técnicas de otras áreas científicas como la Física, las Matemáticas, las simulaciones numéricas y el análisis numérico, los procesos estocásticos y la teoría de las fluctuaciones, así como la teoría del control y otras herramientas provenientes[12] de otras áreas científicas. Intenta crear modelos comprensibles de sistemas mediante el estudio de las relaciones y las interacciones entre las diferentes partes de un sistema biológico (por ejemplo, las redes génicas y las redes de interacción de proteínas implicadas en la señalización celular, las rutas metabólicas, los orgánulos, las células, los sistemas fisiológicos, los organismos[6], entre otras) funcionando como un todo. [6]

Los sistemas biológicos como todos los problemas o fenómenos de la vida se pueden modelar matemáticamente para ayudar a los científicos a comprender el comportamiento dinámico del sistema; aunque en la elaboración de un modelo se hacen algunos supuestos y se consideran algunas simplificaciones de la realidad. Al representar en forma matemática los elementos y relaciones que intervienen en un problema, permite evaluar distintas soluciones factibles y tomar la mejor decisión. También es útil para predecir y comparar el comportamiento de la situación representada frente a diferentes alternativas o en diferentes momentos. [24]

Según los matemáticos Demidowitsch, Maron y Schuwalowa [11]: “Un modelo matemático equivale a una ecuación matemática o un conjunto de ellas sobre la base de los cuales podemos conocer el comportamiento del sistema”. La mayoría de los modelos matemáticos se describen en forma de ecuaciones diferenciales, pero pocas ecuaciones diferenciales tienen una solución analítica sencilla, la mayor parte de las veces es necesario realizar aproximaciones y estudiar el comportamiento del sistema bajo ciertas condiciones. Por esta razón se crearon algoritmos a partir de diferentes técnicas computacionales para trabajar con ecuaciones diferenciales que representarán modelos matemáticos, la más usada es la programación lineal pues es más viable económicamente y más flexible, pero precisamente su linealidad es su mayor desventaja. No obstante, en ocasiones, las propiedades físicas del problema permiten justificar esta linealidad. Otras veces, las relaciones no lineales pueden linealizarse fácilmente aplicando transformaciones matemáticas apropiadas. [24]

Las ecuaciones diferenciales, con las que se representan sistemas biológicos, son usadas principalmente para modelar series temporales. Una serie temporal es un conjunto de observaciones,

obtenidas a lo largo del tiempo, y permite adquirir un conocimiento predicativo o pronóstico, o sea, deduciendo el comportamiento futuro del sistema a partir de los datos obtenidos hasta el momento. Es evidente que la aplicación de técnicas de seguimiento de alto rendimiento ha permitido alcanzar un soporte de información que revela un potencial creciente de modelos para predecir la susceptibilidad a las enfermedades, la respuesta al tratamiento e incluso, el mayor reto que supone el pronóstico acerca de la evolución de la enfermedad [23].

Una serie temporal puede ser representada como una matriz donde una de las variables será el tiempo y las demás variables pertenecen al sistema que está siendo objeto de estudio. Al tener sistemas biológicos representados a través de sistemas de ecuaciones diferenciales, el investigador realiza simulaciones de ese sistema en el tiempo, es decir, obtener series temporales. La simulación es una técnica para crear modelos de sistemas grandes y complejos que incluyen incertidumbre. Se diseña un modelo para repetir el comportamiento del sistema [10] cuando el proceso es muy complejo para ser estudiado de forma analítica.

Estas simulaciones traen consigo un gran cúmulo de información, generalmente se guardan todos esos datos para analizarlos más tarde y convertirlos en conocimiento, pero como casi siempre es una inmensa cantidad, los análisis se hacen superficiales y existe información oculta que es importante para la toma de decisiones. Con el objetivo de ayudar a eliminar el problema anteriormente planteado se crearon técnicas que ayudan a la interpretación y al procesamiento de grandes volúmenes de datos.

La Minería de Datos representa la posibilidad de buscar información dentro de grandes volúmenes de datos con la finalidad de extraer información nueva y útil que se encuentra oculta [19] en los mismos. La base de todos sus métodos son matemáticos, entre ellos se encuentran los métodos que incluyen el trabajo con variables estadísticas: varianza, desviación estándar, covarianza y correlación entre los atributos; análisis de componentes, análisis de factores, análisis de clusters y análisis de regresión. Se han creado potentes herramientas, o sea, aplicaciones informáticas, para facilitar el análisis de grandes volúmenes de datos, entre ellos se encuentran Oracle Data Miner, Clementine y Weka. Estas herramientas basadas en técnicas de Inteligencia Artificial, utilizan varios algoritmos de análisis, entre ellos el Clustering, que permite relacionar partes de un conjunto enorme de datos para que el investigador pueda entender mejor los resultados del análisis.

En la facultad 6 se desarrolla el software BioSyS, el cual está concebido para la simulación de problemas biológicos y como resultado se obtiene gran cantidad de simulaciones. En la versión actual de BioSyS, el análisis por Clustering que se hace de las series temporales sólo tiene en cuenta el estado final al que llega el sistema en cada simulación, pero haciendo Clustering únicamente con el estado final de las series temporales no se realiza un análisis profundo, porque puede ser que en el

tiempo determinado por el científico las dos series temporales se comporten de la misma forma, o sea, las variables de ambas series temporales tengan los mismos valores en ese tiempo y como resultado serán agrupados en el mismo cluster y en realidad son muy distintas, pues los valores correspondientes a los demás tiempos son diferentes, por lo que todavía no se obtienen resultados óptimos. Utilizando para el análisis las series temporales se tendrán en cuenta todos los valores correspondientes a las variables en todos los tiempos hasta el tiempo determinado como final.

Por esta razón surge como **problema científico**: ¿Cómo realizar Clustering de los resultados del proceso de simulación del software BioSyS utilizando series temporales?

Teniendo en cuenta el problema anteriormente descrito la investigación se plantea como **objeto de estudio:** Minería de Datos para el análisis de series temporales.

Pero como en esta área convergen muchos estudios se plantea como **campo de acción:** Clustering en series temporales.

Como en la actualidad BioSyS no realiza Clustering con las series temporales, se plantea como **objetivo general:** Integrar al software BioSyS la posibilidad de realizar Clustering utilizando las series temporales.

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

1. Definir un nuevo criterio de similitud para realizar clusters a un conjunto de series temporales.
2. Modificar la implementación del algoritmo más óptimo de Weka para la realización de Clustering de series temporales de forma secuencial.
3. Comparar el algoritmo de Clustering seleccionado en la versión 1.0 de BioSyS y el algoritmo modificado.

Para darle solución al problema planteado y dar cumplimiento a los objetivos se trazaron las siguientes **tareas**:

1. Familiarización y profundización del estado del arte del tema.
2. Estudio de la función de distancia Euclidiana.
3. Estudio de las series temporales creadas a partir de las simulaciones biológicas.
4. Estudio de las técnicas de Clustering en Weka.
5. Selección del algoritmo para realizar la adaptación.
6. Reimplementación del algoritmo con el nuevo criterio de similitud de forma secuencial.
7. Comparación del algoritmo de Clustering de BioSyS 1.0 y el algoritmo propuesto.

Estructuración del contenido con una breve explicación de sus partes

CAPÍTULO 1: REVISION BIBLIOGRAFICA: En este primer capítulo se detallan los conceptos relacionados con el análisis de una nueva funcionalidad que se incorporará a la segunda versión de BioSyS para lograr una mejor compresión. Se empezará mencionando los software que hacen posible las investigaciones biológicas, así como los que realizan Minería de Datos en el mundo de la Bioinformática. También se hace referencia a los distintos tipos de distancia y el formato .arff, que es el utilizado en Weka, software que se usa de apoyo en la aplicación. Además de describir los distintos algoritmos o técnicas de Clustering utilizados en la primera versión del software BioSyS.

CAPÍTULO 2: PROGRAMAS Y METODOLOGIAS: En el presente capítulo se hará referencia a las diferentes herramientas usadas actualmente por los ingenieros informáticos para desarrollar software, haciéndose un análisis de las mismas y escoger la que más se adecue a las necesidades de los desarrolladores y la aplicación que se va a implementar (lenguaje de programación, herramienta de desarrollo y herramienta de Minería de Datos).

CAPÍTULO 3: RESULTADOS Y DISCUSION: En este último capítulo se desarrolla la propuesta de solución para el problema que se plantea. Se describe la función distancia y el formato del fichero donde se recogerá la información. Se define la reimplementación de los algoritmos que se utilizarán en la nueva versión del software. Se realizarán las pruebas correspondientes para comprobar los resultados que se quieren alcanzar y se efectuará un análisis de los mismos.

1. Revisión bibliográfica

En este primer capítulo se detallan los conceptos relacionados con el análisis de una nueva funcionalidad que se incorporará a la segunda versión de BioSyS para lograr una mejor compresión. Se empezará mencionando los software que hacen posible las investigaciones biológicas, así como los que realizan Minería de Datos en el mundo de la Bioinformática. También se hace referencia a los distintos tipos de distancia y el formato .arff, que es el utilizado en Weka, software que se usa de apoyo en la aplicación. Además de describir los distintos algoritmos o técnicas de Clustering utilizados en la primera versión del software BioSyS.

1.1. Biología de Sistemas

La Biología de Sistema (BS) es una disciplina que pretende integrar diferentes niveles de información [6]. Intenta crear modelos comprensibles de sistemas mediante el estudio de las relaciones y las interacciones entre las diferentes partes de un sistema biológico [6] (sistemas abiertos que operan en condiciones alejadas del equilibrio termodinámico, con muchas y fuertes interacciones no lineales entre sus muchos elementos) con el fin de entender cómo funcionan los mismos desde la perspectiva de un sistema [6]. La BS representa la integración de conceptos e ideas de las ciencias de la vida, disciplinas de integración de conceptos e ideas de las ciencias de la vida, disciplinas de ingeniería y ciencias computacionales.

Este nuevo acercamiento hace mucho más énfasis en el comportamiento de colecciones de componentes funcionando como un todo que a los enfoques tradicionales del estudio de componentes de forma individual. La Biología de Sistemas emplea técnicas de predicción rigurosas. Estas técnicas surgen fundamentalmente del uso de modelos matemáticos que describen el comportamiento del ente en estudio [4].

Para el estudio de los SB la comunidad de investigadores de la BS ha propuesto un modelo al que llaman “las 4 Ms” dividiéndolo esencialmente en dos áreas, un área experimental y otra computacional.

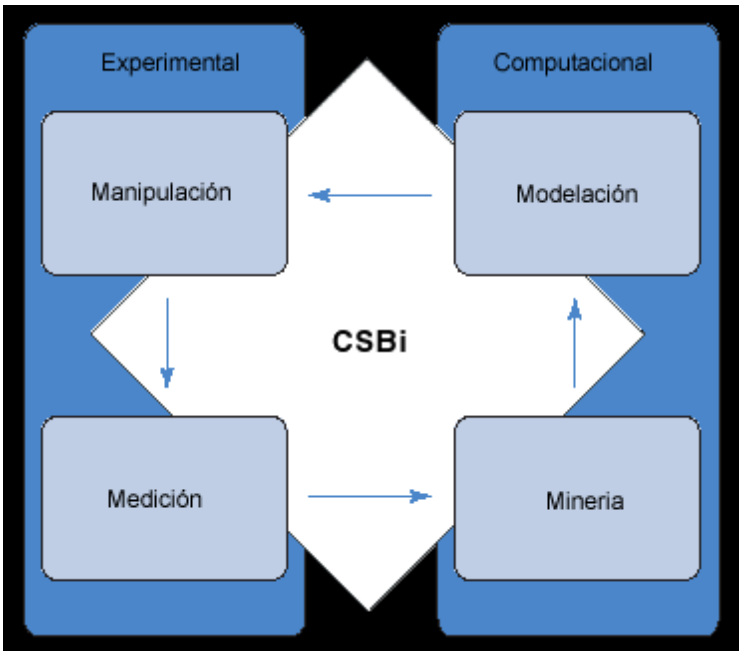

Figura 1. Las 4Ms

Un paso importante para que los científicos comiencen a utilizar el enfoque sistémico en sus investigaciones ha sido el desarrollo de herramientas computacionales para el manejo de la información en estas 4 áreas.

1.2. Simulación de Sistemas Biológicos.

La Simulación de Sistemas Biológicos surge desde la década del 40 como una forma de modelar situaciones de la vida real, es una técnica numérica para reproducir artificialmente un fenómeno o el funcionamiento de un sistema, utilizando modelos matemáticos y lógicos, específicos para cada proceso, que describen el comportamiento y la estructura de dicho fenómeno a través del tiempo. La simulación ofrece la posibilidad de comprimir el tiempo, esfuerzo y cantidad de recursos necesarios para tomar decisiones [21]. Esta técnica es una poderosa herramienta que es muy utilizada por los científicos que estudian los sistemas complejos auxiliándose fundamentalmente de ecuaciones diferenciales. Para lograr obtener resultados objetivos de los estudios realizados a partir de la gran cantidad de información que se extrae de la simulación de sistemas biológicos, anteriormente se aplicaban técnicas estadísticas clásicas de forma manual, pero esto era ineficiente para grandes volúmenes de datos, por lo que se incentivó la creación de técnicas automatizadas con la ayuda de

herramientas más complejas dirigido al descubrimiento del conocimiento permitiendo detectar fácilmente patrones en los datos almacenados, a esto se le conoce como Minería de Datos.

1.3. Técnicas o Algoritmos de minería de datos.

Una técnica de Minería de Datos es el mecanismo que crea modelos de minería de datos. Para crear un modelo, un algoritmo analiza primero un conjunto de datos, buscando patrones y tendencias específicos. Después, el algoritmo utiliza los resultados de este análisis para definir los parámetros del modelo de minería de datos [1].

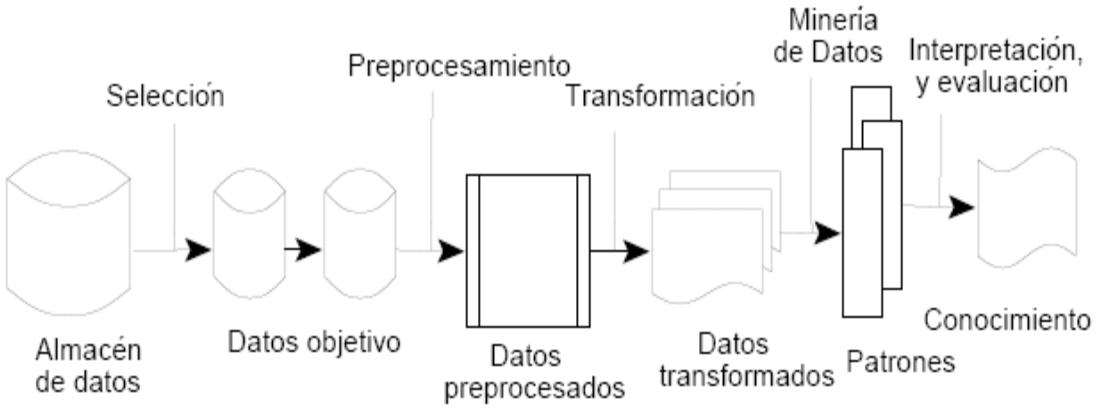

Figura 2. Proceso de Descubrimiento en una Base de Datos [19]

El modelo de Minería de Datos crea un algoritmo que puede tomar diversas formas, incluyendo [1]:

- Un conjunto de reglas que describen cómo se agrupan los productos en una transacción.
- Un árbol de decisión que predice si un cliente determinado comprará un producto.
- Un modelo matemático que predice las ventas.
- Un conjunto de clusters que describe cómo se relacionan los escenarios de un conjunto de datos.

Existen diferentes técnicas de Minería de Datos, entre ellas se encuentran el aprendizaje basado en casos, el razonamiento basado en memoria (MBR), las agrupaciones difusas (técnica de conjuntos difusos, en inglés fuzzy sets), la detección automática de clusters, los árboles de decisión, el

descubrimiento de reglas de Asociación, obtener reglas de disociación, el análisis de enlace y el análisis de series de tiempo.

Las técnicas de minería de datos más representativas son:

- **Redes neuronales:** Son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida.
- **Árbol de decisión:** Es un modelo de predicción utilizado en el ámbito de la Inteligencia Artificial, dada una base de datos se construyen estos diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema.
- **Modelos estadísticos:** Es una expresión simbólica en forma de igualdad o ecuación que se emplea en todos los diseños experimentales y en la regresión para indicar los diferentes factores que modifican la variable de respuesta.
- **Agrupamiento o 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.

Además existen otras técnicas como son:

- **Algoritmos genéticos:** Se caracterizan por ser métodos numéricos de optimización, las variables objeto de estudio y las variables que se desean optimizar representan un fragmento de información. Estos algoritmos imitan la evolución de las especies (incluyendo la mutación, reproducción y selección natural). Además tienen en cuenta el principio de supervivencia de los más fuertes y aptos de la especie representada. Así las variables que obtengan mejores valores de salida serán las que tengan mejores segmentos para la reproducción y los sujetos mejores representados serán los que podrán permutar y pasarán sus características de generación en generación.
- **Razonamiento basado en memoria (MBR):** Esta técnica se basa en las predicciones de instancias desconocidas utilizando la información de las instancias conocidas. Combina los valores vecinos de las instancias que ya se conocen y asigna clasificaciones o predicciones. Para ello es necesario identificar los vecinos más cercanos, es decir, los valores similares para

igual instancia y después observar cuál es el comportamiento de la variable de salida. Se puede ponderar atributos para expresar su importancia en esta técnica.

- **Lógica Difusa (Logica Fuzzy):** Se trata de una técnica basada en un tipo de lógica para procesar datos inciertos. Un atributo puede tener varias variaciones entre Verdadero y Falso, o sea, un atributo puede pertenecer a un grupo en cierta medida, por lo que un mismo atributo puede pertenecer a varios grupos.
- **Segmentación:** Este tipo de técnicas permiten agrupar registros en una base de datos basándose en una serie de atributos (varios cientos o sólo unos cuantos, dependiendo de la aplicación de negocio). Los registros en estos grupos o segmentos se seleccionan de forma que sean lo más parecidos posible, siendo cada grupo diferente a todos los demás. En un contexto CRM, los algoritmos de segmentación se emplean para agrupar clientes en segmentos en función de un número reducido de atributos de compra. Este esquema puede ser empleado para facilitar la comprensión de las distintas tipologías de clientes, y para construir un entorno en el que analice su cambio a lo largo del tiempo.
- **Clasificación:** Esta técnica se emplea con el fin de obtener un mayor conocimiento sobre las instancias y poder predecir valores categóricos o cualitativos. Utilizándolo en conjunto con un esquema de segmentación, este tipo de técnica puede ser empleada para clasificar a una nueva instancia, teniendo en cuenta un número de actividades realizadas por la misma. Además se puede usar para predecir acciones de las propias instancias ya existentes dentro de la Base de Datos.
- **Predicción:** Esta técnica consta de algoritmos que permiten construir modelos que estimen un valor cuantitativo. Funciona de manera similar a las técnicas de clasificación. Mediante el algoritmo se puede obtener un valor entre 0 y 1 que represente la propensión de un determinado valor a un grupo y a partir de ahí tomar decisiones.
- **Reglas de asociaciones:** Esta técnica tiene algoritmos de la forma "si esto pasa, entonces ocurre lo siguiente". En esencia los algoritmos de asociación lo que hacen es asociar acontecimientos. Realizan predicciones a partir de los valores de variables que se relacionan dentro de una determinada Base de Datos.

1.4. Software que realizan minería de datos.

Actualmente no existen muchas herramientas que se dediquen a realizar minería de datos, sin embargo, muchos ingenieros informáticos se han dedicado a crear potentes herramientas para facilitar la gestión de la información. Entre ellos se encuentran Oracle Data Miner, Clementine *y* Weka, de los cuales a continuación se muestran sus características:

Clementine

Clementine es un sistema integrado que permite encontrar patrones en la información para facilitar la toma de decisiones a los usuarios. Facilita descubrir lo que ocultan sus datos. Tiene una interfaz gráfica sencilla. Incrementa la productividad de los analistas. Funciona sobre todas las plataformas hardware y sistemas operativos, incluyendo Unix, VMS y Windows NT. Con el uso de Clementine, los analistas y usuarios de negocios pueden acceder a datos de varias fuentes para producir, evaluar, y desplegar modelos analíticos rápida y fácilmente. Permite obtener el máximo provecho de la infraestructura actual, pues tiene una arquitectura abierta y escalable del producto.

Oracle Data Mining (ODM)

Oracle Data Mining es una opción de Oracle Database Enterprise Edition. La misma permite a los clientes generar información predecible y lista para usar, y crear aplicaciones integradas de inteligencia de negocios. Al utilizar la funcionalidad Data Mining incorporada en Oracle Database, los clientes pueden descubrir los patrones y conocimientos que se encuentran ocultos en los datos. Oracle Data Mining hace todas las transformaciones necesarias automáticamente de forma interna, liberando así de este trabajo a los usuarios o desarrolladores. Además soporta la clasificación de valores dentro de un campo en grupos que tengan sentido [25].

Weka

Weka es un software que ha sido desarrollado en la universidad de Waikato (Nueva Zelanda) bajo licencia GPL, esto ha posibilitado que se convierta en una alternativa interesante y sugerente, tanto que se ha convertido en una de las suites más utilizadas en el área en los últimos años. Se emplea fundamentalmente para analizar y buscar patrones de comportamiento comunes. Está formado por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación,

agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados [3].
En la aplicación BioSyS se utiliza este software para realizar Minería de Datos después que estos son extraídos de la BD. Esta información se almacena en una estructura de datos interna que tiene el Weka, pero que es la equivalente de un fichero .arff con el que comúnmente el Weka trabaja.

Fichero .arff utilizado en la herramienta Weka.

La estructura de un fichero con formato **.arff (Attribute-Relation File Format)** es muy sencilla. Se divide en 3 partes: @relation, @atribute y @data.

@relation<relation-name>

Todo fichero .arff debe comenzar con esta declaración **en su primera línea** (no podemos dejar líneas en blanco al principio). **<relation-name>** será una cadena de caracteres y si contiene espacios la pondremos entre comillas [15].

@attribute<attribute-name><datatype>

En esta sección incluiremos una línea por cada atributo (o columna) que vayamos a incluir en nuestro conjunto de datos, indicando su nombre y el tipo de dato. Con **<attribute-name>** indicaremos el nombre del atributo, que debe comenzar por una letra y si contiene espacios tendrá que estar entrecomillado.

Con **<datatype>** indicaremos el tipo de dato para este atributo (o columna) que puede ser:

- **numeric** (numérico)
- **string** (texto)
- **date [<date-format>]** (fecha). En <date-format> indicaremos el formato de la fecha, que será del tipo "yyyy-MM-dd'T'HH:mm:ss".
- **<nominal-specification>**. Estos son tipos de datos definidos por nosotros mismos y que pueden tomar una serie de valores que indicamos [15].

@data

En esta sección incluiremos los datos propiamente dichos. Separaremos cada columna por comas y todas filas deberán tener el mismo número de columnas, número que coincide con el de declaraciones @attribute que añadimos en la sección anterior.
Si no disponemos de algún dato, colocaremos un signo de interrogación (?) en su lugar. El separador de decimales tiene que ser obligatoriamente el punto y las cadenas de tipo string tienen que estar entre comillas simples. [15]

```
@relation Simulation_clustered

@attribute Instance_number numeric
@attribute ID string
@attribute A numeric
@attribute B numeric
@attribute C numeric
@attribute D numeric
@attribute E numeric
@attribute Cluster {cluster0,cluster1,cluster2,cluster3}


@data
0,12_22_32,1,2,3,4,5,cluster0
1,13_22_32,1,2,3,4,5,cluster1
2,12_24_32,1,2,3,4,5,cluster2
```

Figura 3. Fichero .arff

1.5. Análisis de Series Temporales.

Una serie temporal puede definirse como una sucesión ordenada en el tiempo de valores de una variable. Puede ser representada como una matriz donde una de las variables es el tiempo y las demás son las variables del sistema en estudio. No es más que la evolución temporal de un sistema. Uno de los usos principales de los modelos de series temporales ha sido el de proporcionar predicciones a corto y medio plazo de las variables [18] que conforman el sistema.

Utilizando el software BioSyS para simular algún modelo asociado a un sistema biológico se pueden obtener millones de series temporales, que no son más que los resultados de las simulaciones. El Análisis de Series Temporales comprende métodos que ayudan a interpretar los datos, extrayendo información representativa, tanto referente a los orígenes o relaciones subyacentes, como a la posibilidad de extrapolar y predecir su comportamiento futuro. Algunos tipos de análisis de series temporales que existen son:

- **Dinámica de poblaciones**

  La Dinámica de Poblaciones es muy utilizada cuando se quiere ver si el modelo que se ha construido se ajusta a la realidad del problema en estudio. Para ello se le asignan valores a las variables para los cuales se conoce el resultado. Si la salida observada se aproxima al resultado

esperado entonces el modelo se ajusta al problema real y puede ser utilizado. De lo contrario habría que redefinir el mismo. Este tipo de análisis permite a los investigadores tener una idea general del comportamiento del sistema.

- **Clasificación**

Clasificación es otro tipo de análisis implementado dentro del sistema. Este consiste en clasificar simulaciones desconocidas a partir de modelos generados con simulaciones previamente clasificadas. Esta clasificación previa se ha concebido de tres formas. La primera de ellas es tomar para la creación de los modelos aquellos ficheros que salen del análisis de clusters, pues ya en este tipo de análisis se ha hecho una clasificación. Las otras dos formas consisten en que el usuario explora la base de datos y va clasificando las simulaciones de acuerdo a clases definidas por el mismo. La diferencia está en que puede explorar la base de datos manualmente o pedirle al sistema que le muestren dinámicas de forma aleatoria.

- **Análisis por Reglas**

El Análisis por Reglas 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 clusters o clasificaciones, el usuario puede estudiar hacia que comportamiento tiende el sistema cuando se varían determinados parámetros dos a dos.

- **Análisis de Bifurcaciones**

Este tipo de análisis es muy sencillo, el mismo consiste en variar sólo uno de los parámetros que componen el modelo matemático y plotear los estados finales a los que llega el sistema. El análisis de bifurcaciones aunque es simple, es muy útil ya que permite al usuario ver cuál es el efecto provoca el cambio del valor de un determinado parámetro sobre el estado final.

- **Clustering**

Se denomina “cluster” o "grupo" a un punto usado para representar un conjunto de valores de entre todos los iniciales que tienen algo en común, y que se pueden agrupar en función de determinado rasgo [28]. De esta forma, el resultado de la ejecución del algoritmo será un conjunto de puntos que representan el centro de cada cluster o grupo identificado. Teniendo en

cuenta que es muy difícil explorar de forma manual toda la información que de cada modelo haya sido almacenada en la BD, se hace necesario el uso de herramientas que permitan hacer minería de estos datos y faciliten la comprensión del sistema en estudio. Estos permiten agrupar los vectores formados por los valores finales de cada simulación de acuerdo a la distancia existente entre ellos. El usuario puede seleccionar además el algoritmo que desea utilizar. Una vez definido esto el sistema se encarga de realizar el agrupamiento y mostrar el mismo en forma de salida gráfica. El número de clusters que se obtienen se puede asociar con la cantidad de comportamientos diferentes a los que tiende el sistema.

1.6. BioSyS

BioSyS es un software dedicado a la simulación y a los análisis de sistemas biológicos e integra algoritmos y herramientas necesarias para servir de apoyo en las investigaciones a aquellos científicos que se dedican al estudio de la Biología de Sistemas. Fue desarrollado por un conjunto de especialistas en Biología de Sistemas del Centro de Inmunología Molecular (CIM) y la Universidad de las Ciencias Informáticas (UCI). Incluye herramientas que posibilitan, a partir de un sistema biológico hacer modelación, edición de ecuaciones, simulación (ya sea local o distribuida) y realizar análisis de las simulaciones obtenidas.

Funcionalidades de BioSyS 1.0

- Edición de Ecuaciones

Edición de Ecuaciones implementa una serie de funcionalidades adicionales que facilitan el trabajo con los modelos matemáticos, la reutilización de funciones y todas las funcionalidades relacionadas con la gestión de los modelos matemáticos. Al mismo se accede cuando se quiere crear un nuevo modelo o editar uno ya existente.

- Simulación local y distribuida

Es necesario contar con un módulo de simulación en el software BioSyS ya que sólo en casos particulares un sistema de ecuaciones diferenciales tiene solución analítica y aún en muchos de estos casos esta solución es muy complicada de encontrar. Al final lo que se persigue es obtener una solución aproximada para un sistema dado, con unas condiciones iniciales definidas por el usuario, un conjunto de parámetros y un tiempo inicial y final.

- Análisis de las simulaciones

  La idea que se persigue con este módulo es proveer al usuario de toda una serie de funcionalidades que le facilite la interpretación de los resultados.

Esta investigación se encuentra vinculada específicamente al Análisis por Clustering, técnica que se utiliza dentro de la funcionalidad que permite realizar análisis de las simulaciones biológicas obtenidas. Este software incluye métodos para ayudar a interpretar los resultados de las series temporales a partir de las simulaciones realizadas por el investigador y que contribuyen a obtener valiosos resultados para la predicción y el comportamiento futuro de las series temporales; entre ellos se encuentra el análisis por regla, la dinámica de poblaciones, clasificación, análisis de bifurcaciones y el Clustering o agrupamiento (explicados en la sección 1.3); de todos los anteriormente mencionados se hará énfasis en el análisis por Clustering.

1.7. Algoritmos de Clustering

El proceso de Clustering consiste en la división de los datos en grupos de objetos similares. Cuando se representan la información obtenida a través de clusters se pierden algunos detalles de los datos, pero a la vez se simplifica dicha información. Clustering es una técnica en la que el aprendizaje realizado es no supervisado (unsupervised learning). Desde un punto de vista práctico, el Clustering juega un papel muy importante en aplicaciones de Data Mining, tales como exploración de datos científicos, recuperación de la información y minería de texto, aplicaciones sobre bases de datos espaciales (tales como GIS o datos procedentes de astronomía), aplicaciones Web, marketing, diagnóstico médico, análisis de ADN en biología computacional, y muchas otras [14]. De forma general, las técnicas de Clustering son las que utilizando algoritmos matemáticos se encargan de agrupar objetos. Usando la información que brindan las variables que pertenecen a cada objeto se mide la similitud entre los mismos, y una vez hecho esto se colocan en clases que son muy similares internamente (entre los miembros de la misma clase) y a la vez diferente entre los miembros de las diferentes clases.

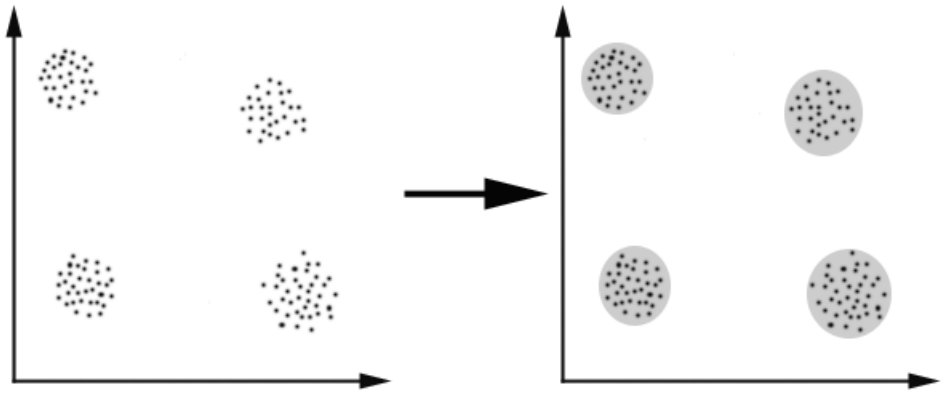

Figura 4. Clustering [7]

En este caso, se identifican fácilmente los 4 grupos en los que los datos se pueden dividir, el criterio de similitud es la distancia: dos o más objetos pertenecen al mismo grupo si son "cerca", de acuerdo a una determinada distancia (en este caso la distancia geométrica) [7].

Figura 5. Pasos para realizar análisis de cluster [15]

El análisis de clusters consiste en agrupar diferentes grupos de entidades de acuerdo a una función de distancia. En la versión 1.0 de BioSyS el análisis de clusters que se implementa hace uso de la distancia euclidiana y trabaja con datos del tipo continuo, pues lo que se pretende es agrupar los

estados finales a los que llegan las diferentes simulaciones, sin tener en cuenta la dinámica completa. Teniendo en cuenta esto, dentro de la funcionalidad que existía en la versión 2.0 de BioSyS de realizar Análisis de Clustering se incluyó la posibilidad de hacerlo con la serie temporal, pues en la versión anterior sólo se realizaba con el tiempo final.

Para que un algoritmo de Clustering sea ideal debe tener las siguientes características:

Escalabilidad: Se necesita que los algoritmos tengan una gran escalabilidad para que puedan trabajar y hacer Clustering de datos en Bases de Datos con millones de simulaciones.

Capacidad de trabajar con diferentes tipos de atributos: Es necesario implementar algoritmos que trabajen con atributos de tipo alfanuméricos, binarios o numéricos porque la mayoría de los algoritmos de Clustering están implementados para trabajar solo con datos numéricos.

Descubrimiento de clusters con formas arbitrarias: Es necesario implementar algoritmos que den como resultados clusters que no sean necesariamente circulares o sea, utilizar otras funciones de distancia.

Evitar tener que requerir parámetros de entrada: Implementar algoritmos que no soliciten al usuario datos o parámetros de entrada pues en la mayoría de los casos esos datos son difíciles de determinar y de manejar lo que hace que no se pueda controlar la calidad del algoritmo.

Tratar eficientemente datos con ruido: Algunos algoritmos de Clustering no son eficientes porque son sensibles a datos con comportamiento extraño, datos faltantes, desconocidos o erróneos.

Alta dimensionalidad: Es necesario tener algoritmos de Clustering que sean capaces de trabajar con base de datos que contengan varias dimensiones o gran cantidad de atributos.

Clustering basado en restricciones: En muchas ocasiones es necesario que los algoritmos de Clustering, además de tener en cuenta el comportamiento también sean capaces de satisfacer ciertas restricciones.

Interpretación y uso: Lograr que los investigadores o usuarios finales puedan comprender fácilmente los resultados del Clustering y que sean fáciles de usar.

Algoritmos de Clustering usados en BioSyS.

Teniendo en cuenta la cantidad de información que se maneja, en BioSyS, se utilizan los siguientes algoritmos de Clustering porque permiten agrupar los vectores formados por los valores finales de cada simulación:

Simple K-Means

Este algoritmo debe definir el número de clusters que se desean obtener, así se convierte en un algoritmo voraz para particionar. Los pasos básicos para aplicar el algoritmo son muy simples. Primeramente se determina la cantidad de clusters en los que se quiere agrupar la información, en este caso las simulaciones. Luego se asume de forma aleatoria los centros por cada clusters. Una vez encontrados los primeros centroides el algoritmo hará los tres pasos siguientes:

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

Finalmente quedarán agrupados por clusters, los grupos de simulaciones según la cantidad de clusters que el investigador definió en el momento de ejecutar el algoritmo [27].

X-Means

Es una variante mejorada del algoritmo K-Means. Su ventaja fundamental está en haber solucionado una de las mayores deficiencias presentadas en K-Means, el hecho de tener que seleccionar a priori el número de clusters que se deseen obtener, a X-Means se le define un límite inferior *K*-min *(*número mínimo de clusters*)* y un límite superior K-Max *(*número máximo de clusters*)* y este algoritmo es capaz de obtener en ese rango el número óptimo de clusters, dando de esta manera más flexibilidad al usuario. Durante este proceso, el conjunto de centroides que alcanzan el mejor valor son almacenados, y estos serían la salida final, es decir, los valores finales de cada simulación de acuerdo a la distancia entre ellos. Los mismos son aplicables cuando en la Base de Datos existen al menos 2 simulaciones para el modelo (que son ecuaciones formadas por arreglos de parámetros y condiciones iniciales). Se ha comprobado que sus resultados son más fiables que los obtenidos con el K-Means, debido a que presenta un valor de distorsión menor, son mucho mejor para realizar Clusters de un conjunto grande de datos y es incluso una variante mucho más rápida [27].

CobWeb

Pertenece a la familia de algoritmos jerárquicos. Se caracteriza por la utilización de aprendizaje incremental, esto quiere decir, que realiza las agrupaciones instancia a instancia. Durante la ejecución del algoritmo se forma un árbol (árbol de clasificación) donde las hojas representan los segmentos y el nodo raíz engloba por completo el conjunto de datos. Al principio, el árbol consiste en un único nodo raíz. Las instancias se van añadiendo una a una y el árbol se va actualizando en cada paso. La clave para saber cómo y dónde se debe actualizar el árbol la proporciona una medida denominada utilidad de categoría, que mide la calidad general de una partición de instancias en un segmento [27]. Pertenece a los métodos de aprendizaje conceptual o basado en modelos. Esto significa que cada cluster se considera como un modelo que puede describirse intrínsecamente, más que un ente formado por una colección de puntos. Además en el algoritmo también hay que tener en cuenta dos parámetros muy importantes:

Acuity: es un parámetro muy necesario, pues la utilidad de categoría está basada en la estimación de la media y la desviación estándar del valor de un atributo para un nodo en particular, el resultado es 0 si dicho nodo solo tiene una instancia; por lo que se puede decir que el valor que toma este parámetro es la medida del error de un nodo con una sola instancia (establece la varianza mínima de un atributo) [13].

Cut-off: este parámetro es usado para evitar el crecimiento descontrolado de la cantidad de segmentos. Indica el grado de mejoría que se debe producir en la utilidad de categoría para que la instancia se pueda tener en cuenta de manera individual. Resumiendo, cuando se va a añadir un nuevo nodo y no es suficiente el crecimiento de la utilidad de categoría, pues ese nodo se poda y la instancia pasa a otro nodo ya existente [13].

Para realizar cluster utilizando matrices en lugar de vectores, no se pueden cargar todas las series temporales en un solo fichero, no sería óptimo para la aplicación pues consumiría mucha memoria. Es por eso que en lugar de usar los algoritmos Simple K-Means y X-Means (estos necesitan calcular la distancia del centroide a la serie temporal en cada una de las iteraciones que realizan para agrupar cada una de estas en un cluster) se utilizará un algoritmo jerárquico, en este caso se usará el algoritmo CobWeb, ya implementado dentro del software Weka que es usada como herramienta de apoyo a BioSyS y además sólo calcula una vez la distancia a cada una de las series temporales y así el algoritmo sería menos costoso.

1.8. Consideraciones generales

Como parte de la revisión bibliográfica se llega a la conclusión que es necesario modificar el algoritmo donde se implementa la función de distancia euclidiana para realizar Clustering en el software BioSyS 1.0. En la bibliografía consultada se encontraron diferentes criterios de similitud, pero se eligió el Coeficiente de Correlación Lineal de Pearson, por las ventajas que presenta el mismo para darle solución al problema planteado en la investigación. CobWeb fue el algoritmo escogido para incorporarle el nuevo criterio de similitud. Además, se incorporan al capítulo aspectos generales que fueron necesarios para la utilización del software y la interpretación de los resultados obtenidos.

2. Programas y metodologías

En el presente capítulo se hará referencia a las diferentes herramientas usadas actualmente por los ingenieros informáticos para desarrollar software, haciéndose un análisis de las mismas y escoger la que más se adecue a las necesidades de los desarrolladores y la aplicación que se va a implementar (lenguaje de programación, herramienta de desarrollo y herramienta de Minería de Datos).

2.1. Lenguaje de Programación

- **C++**

  C++ es un lenguaje imperativo orientado a objetos derivado del C**.** En realidad es un súperconjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción [33]. Es un lenguaje complicado y requiere páginas y páginas de código para hacer cosas que con otros lenguajes se hacen con pocas líneas. Además se puede decir que la reutilización de código en forma de librería de usuario es otra de sus propiedades. Los programas escritos en C++ tienen otra ventaja sobre el resto, con la excepción del ensamblador, genera los programas más compactos y rápidos. El código es transportable, es decir, un programa en ANSI en C++ podrá ejecutarse en cualquier máquina y bajo cualquier sistema operativo [30]. Desde sus inicios, C++ intentó ser un lenguaje que incluye completamente al lenguaje C (quizás el 99% del código escrito en C es válido en C++), pero al mismo tiempo incorpora muchas características sofisticadas no incluidas en aquél, tales como: POO, excepciones, sobrecarga de operadores, templates o plantillas [31]. Pero este lenguaje de programación presenta algunas desventajas como el hecho de que no es multiplataforma, su arquitectura para el desarrollo orientado a Internet no es estándar. Además de que el lenguaje es difícil de aprender y en ocasiones muy abierto.

- **C#**

C# es un lenguaje de programación de uso general sencillo, con seguridad de tipos y orientado a objetos [32]. Además, elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular [17]. C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo [8]. La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros) [8].

- **Java**

Java, en la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la Informática en general [2]. Debido a la alta productividad, y que requieren poco tiempo de desarrollo, se utiliza Java frente a otros lenguajes de programación como C y C++. Fue ideado para el desarrollo de aplicaciones en Internet [16]. Java, emplea el concepto de máquina virtual y el código que genera no es específico a una plataforma en particular. Implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje [22]. Este lenguaje trabaja con sus datos como objetos y con interfaces a esos objetos y soporta las características propias del paradigma orientado a objetos como son: abstracción, encapsulación, herencia y polimorfismo. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente [29]. Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas [29].

Después de hacer un estudio de algunos de los lenguajes de programación existentes se decidió escoger Java debido a que con él se puede realizar cualquier tipo de programa y es el usado actualmente en BioSyS.

2.2. Herramienta de Desarrollo

- **Eclipse**

  Eclipse es un IDE de código abierto, desarrollado en Java y que no está específicamente orientado sólo al mundo Linux, sino a cualquier tipo de sistema. Actualmente es la apuesta más fuerte dado que se basa en Java, su organización es robusta y su desarrollo es continuo, además de fácilmente extensible, proporcionando a la comunidad de desarrolladores un sistema de enganche de plug-ins muy adecuado para poder adaptar el entorno a las necesidades de desarrollo sobre las que se esté trabajando.

- **Visual Studio**

  El sistema de desarrollo Microsoft Visual Studio es un conjunto de herramientas de desarrollo diseñadas para ayudar a los desarrolladores de software (tanto si son principiantes como profesionales con experiencia) a enfrentarse a los desafíos complejos y crear soluciones innovadoras. La función de Visual Studio es mejorar el proceso de desarrollo y facilitar el trabajo necesario para lograr grandes avances y hacerlo con mayor satisfacción [26].

- **NetBeans 5.5.1**

  NetBeans es una herramienta para el desarrollo de aplicaciones de escritorio usando Java, es de código abierto, de gran éxito y con una gran base de usuarios. Además, es un proyecto GNU que tiene un excelente diseñador de interfaces integrado, es muy rápido y fácil de usar.

Teniendo en cuenta los aspectos mencionados de las herramientas de desarrollo se llegó a la conclusión de utilizar el NetBeans 5.5.1 para el desarrollo de la aplicación teniendo en cuenta el lenguaje de programación seleccionado.

2.3. Herramienta para realizar Minería de Datos

- **Weka**

Como ya se mencionó en el capítulo anterior, Weka es un software programado en Java que está orientado a la extracción de conocimientos desde bases de datos con grandes cantidades de información. Es una colección de algoritmos de aprendizaje de máquina para tareas de Minería de Datos. Se emplea fundamentalmente para analizar y buscar patrones de comportamiento comunes. Está formado por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados. También es muy apropiada para el desarrollo de nuevos planes de aprendizaje de máquina. La licencia de Weka es GPL, lo que significa que este programa es de libre distribución y difusión. Además, ya que Weka está programado en Java, es independiente de la arquitectura, teniendo en cuenta que funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible. Para realizar la búsqueda de datos útiles, el Weka almacena la información en un fichero de tipo .arff. A pesar de ser lento y muy limitado para Redes Neuronales Artificiales (RNA), presenta muchas ventajas por las cuales se decidió utilizar este software, entre ellas [20]:

- Es de libre distribución.
- Es multiplataforma.
- Tiene muchos algoritmos de regresión/clasificación.
- Incluye meta-algoritmos de aprendizaje (Bagging, AdaBoosting, ...)
- Tiene preprocesado de datos (selección, estadísticas,...)
- Incorpora herramientas para la visualización de los datos y resultados.
- Se distribuye también su código fuente JAVA.
- Se pueden añadir nuevas clases de clasificadores y filtros.
- Tiene versiones de consola y con interfaz gráfico.

2.4. Procedimientos

En la primera versión del software BioSyS se generaba un vector con los valores de las variables en su estado final para realizar el agrupamiento o Clustering. Para optimizar el resultado que se le muestra al investigador, se utilizarán las series temporales, teniendo en cuenta todas las variaciones que sufre el

sistema desde su estado inicial hasta el estado definido por el usuario como final. Fue necesario sustituir la clase FastVector por una capaz de facilitar la manipulación de las series temporales.

En la versión 1.0 de BioSyS se implementaron 3 algoritmos para realizar Clustering (Simple K-Means, X-Means y CobWeb), de los cuales se realizaron estudios en el presente trabajo para determinar cuál es el más adecuado para llevar a cabo la investigación:

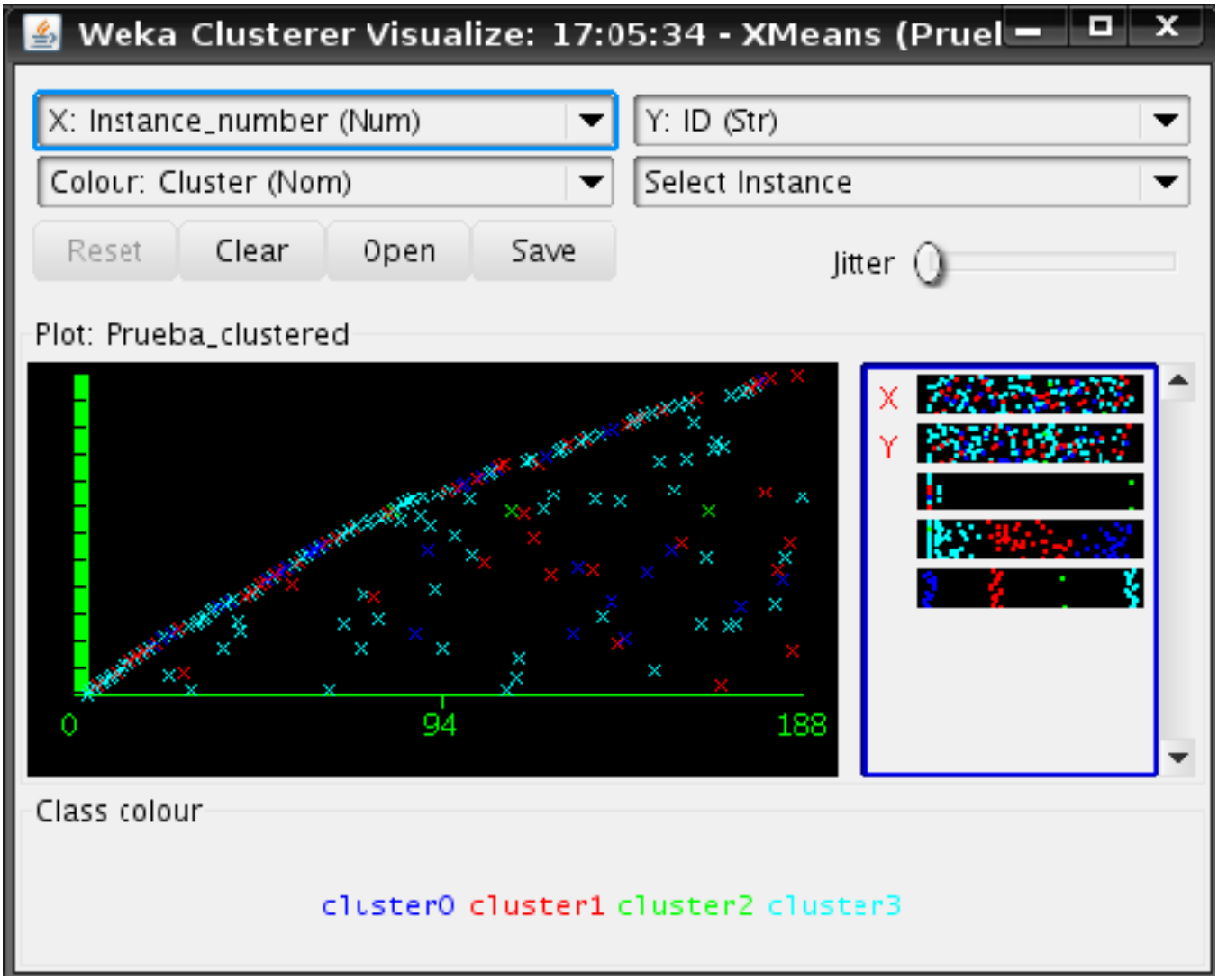

Figura 6. Resultados con el algoritmo X- Means

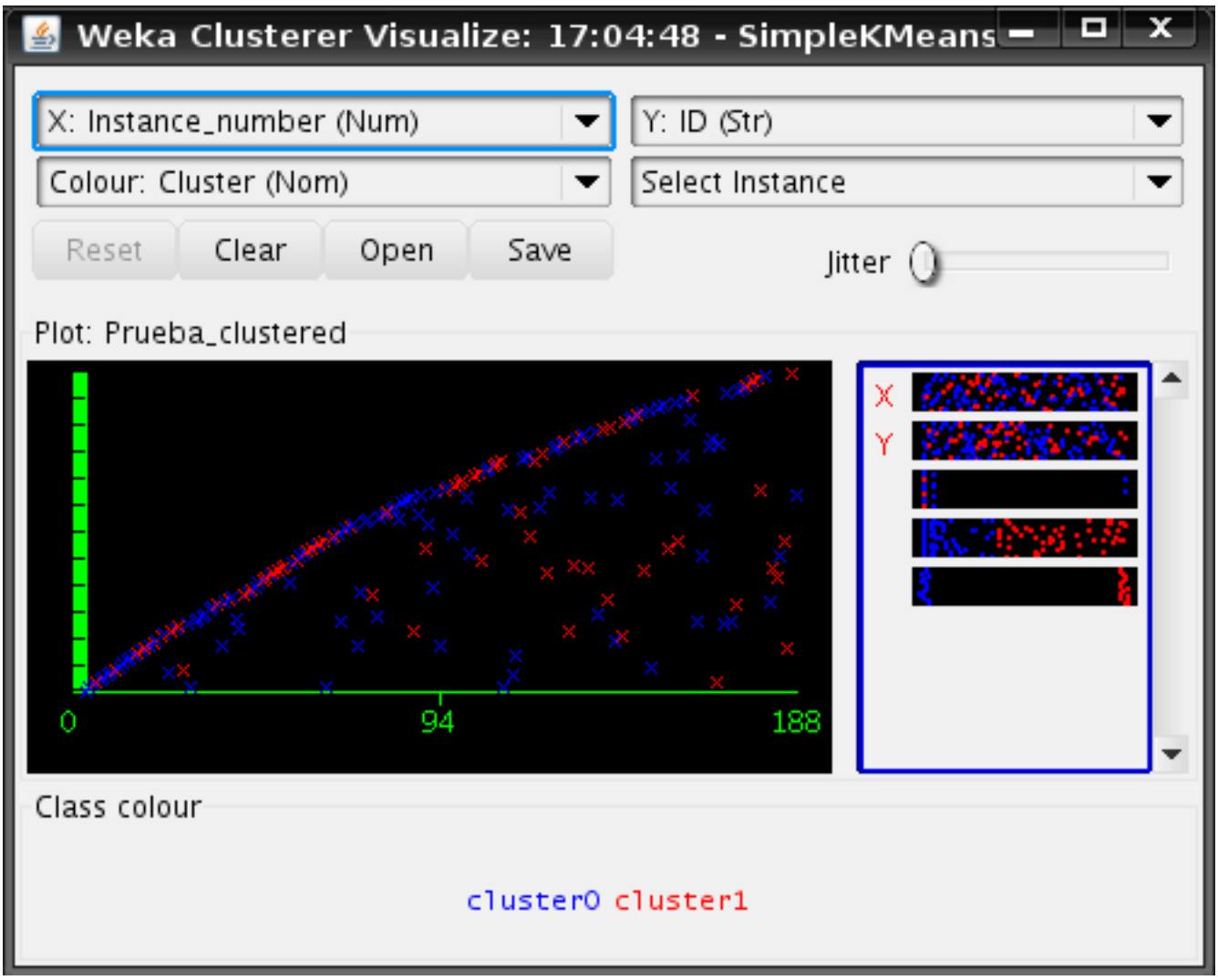

Figura 7. Resultados con el algoritmo Simple K-Means

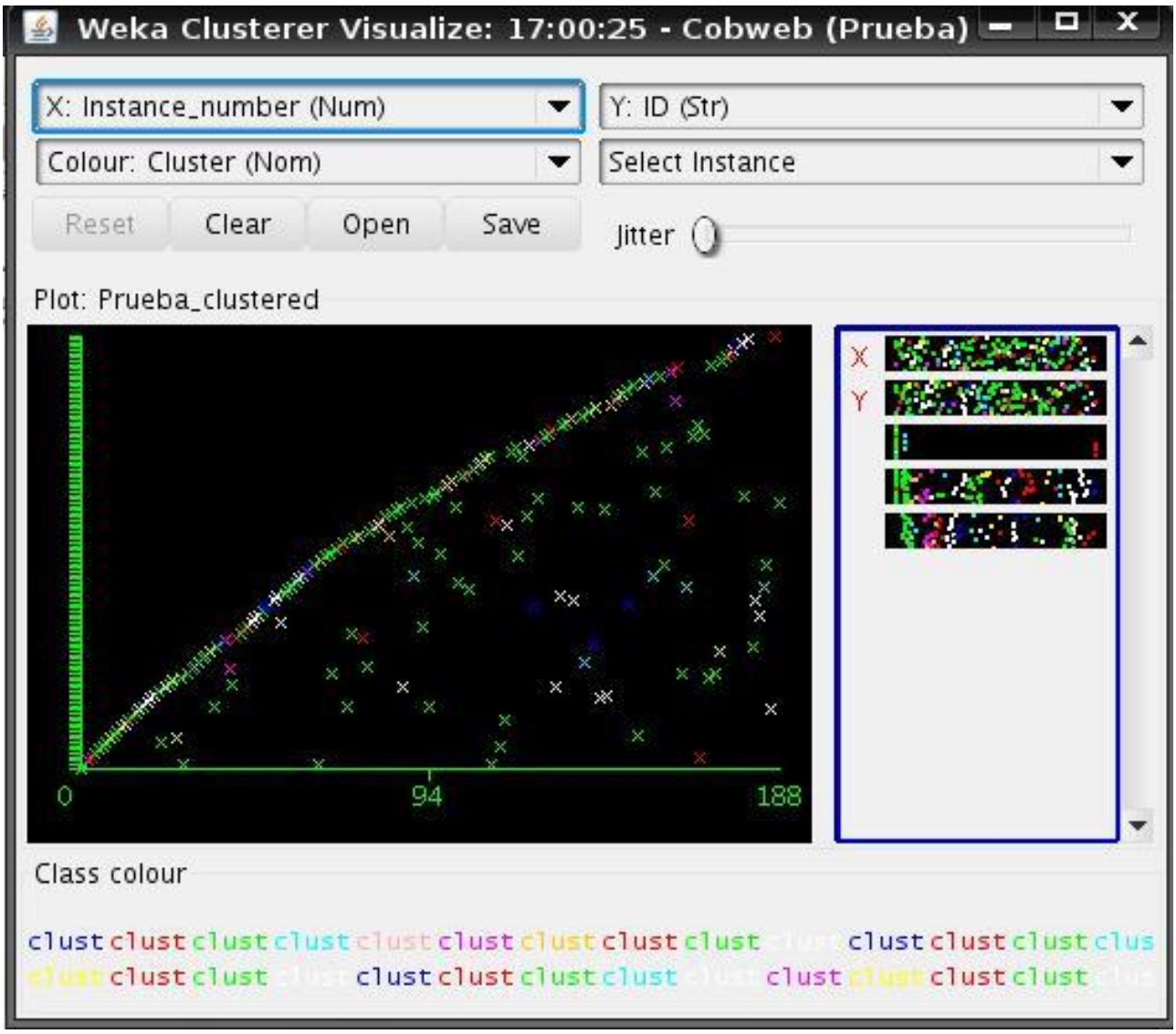

Figura 8. Resultados con el algoritmo CobWeb

Como se puede observar por las imágenes de la representación gráfica de los resultados del Clustering del mismo conjunto de series temporales, el algoritmo CobWeb es el que se destaca, teniendo en cuenta que realiza un mayor número de clusters para la misma cantidad de simulaciones.

En el algoritmo CobWeb las instancias se van añadiendo una a una y el árbol se va actualizando en cada paso. La actualización consiste en encontrar el mejor sitio donde incluir la nueva instancia. La clave para saber cómo y dónde se debe actualizar el árbol la proporciona una medida denominada utilidad de categoría, que mide la calidad general de una partición de instancias en un segmento [14]

esta función da mayor valor a las clases que presentan una alta similaridad entre sus miembros y una baja similaridad con el resto de las clases [5].

Se realizaron pruebas y se comprobó que sería muy engorroso almacenar todas las series temporales correspondientes a la elección del investigador de acuerdo al tiempo final que haya señalado este para realizar el análisis por Clustering y luego cargar del fichero creado las series temporales para realizar las operaciones necesarias para realizar el agrupamiento.

Se guarda un conjunto de identificadores que se utilizarán en tiempo de ejecución, posteriormente se irá consultando a la Base de Datos, pidiéndole en cada iteración del algoritmo, de acuerdo al identificador almacenado en la lista, cada serie temporal sin sobrecargar la memoria de la computadora. Al concluir las operaciones requeridas con las series temporales que en ese momento se estén usando, se libera memoria y se piden las siguientes series temporales, de forma tal que no se estará cargando en la memoria de la máquina todas las series temporales al mismo tiempo.

Después que se extraen de la BD las series temporales correspondientes a los identificadores sacados de la lista almacenada, se calcula el Coeficiente de Correlación Lineal de Pearson como criterio de similitud (denotado con la letra r), que permite valorar si la relación entre ambas series temporales es fuerte o débil, positiva o negativa. Este coeficiente está comprendido entre -1 y 1 y de acuerdo a esto se tienen los siguientes criterios para r [9].

r = 1, la correlación lineal es perfecta, directa o correlación lineal positiva;
r = 0, no existe correlación lineal o correlación lineal nula;
r = -1, la correlación lineal es perfecta, inversa o correlación lineal negativa.

Entre más se aproxima a los valores 1 y -1 la aproximación a una correlación se considera buena. Cuando más se aleja de 1 o de -1 y se acerca a cero se tiene menos confianza en la dependencia lineal por lo que una aproximación lineal será lo menos apropiado, sin embargo no significa que no existe dependencia, lo único que podemos decir es que la dependencia no es lineal. Un valor positivo para r indica que a medida que una variable crece la otra también lo hace, por el contrario si su valor es negativo, lo que podemos decir es que a medida que una variable crece la otra decrece [9].

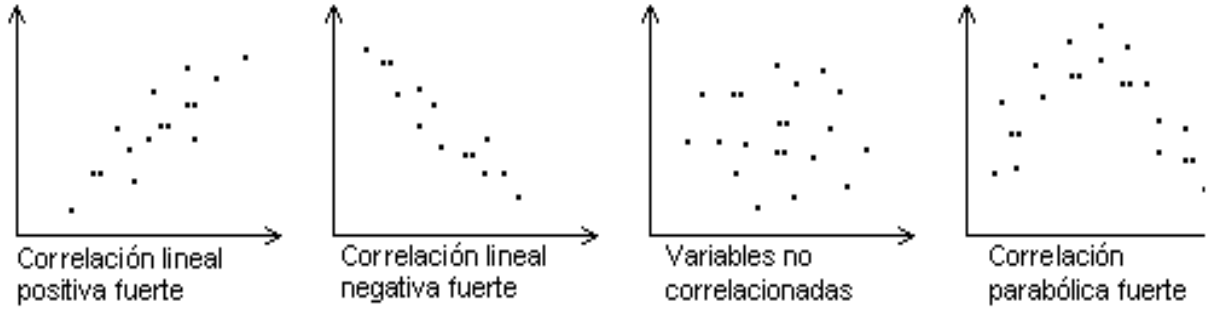

Figura 9. Comportamiento del Coeficiente 1

Es una medida del grado de asociación lineal entre las variables X e Y. Se representa por r:

$$r = \frac{S_{xy}}{s_x \cdot s_y}$$

Donde sx, sy son las desviaciones típicas de las variables X e Y respectivamente, y Sxy es la covarianza muestral de X e Y, que se define como la media de los productos de las desviaciones correspondientes de X e Y y de sus medias muestrales.

Desviación típica de la variable X.	$s_x = \sqrt{\frac{1}{n-1}\sum_{i-1}^{n}(x_i - x_m)^2}$
Desviación típica de la variable Y.	$s_y = \sqrt{\frac{1}{n-1}\sum_{i-1}^{n}(y_i - y_m)^2}$
Covarianza muestral de las variables X e Y.	$S_{xy} = \frac{1}{n-1}\sum_{i-1}^{n}(x_i - x_m)(y_i - y_m)$
Media de la variable X	$x_m = \frac{1}{n}\sum_{i-1}^{n} x$
Media de la variable Y	$y_m = \frac{1}{n}\sum_{i-1}^{n} y$

Tabla 1. Ecuaciones derivadas de la ecuación de Correlación Lineal

Finalmente la ecuación del Coeficiente de Correlación Lineal de Pearson quedaría de la siguiente manera:

$$r = \frac{\frac{1}{n-1}\sum_{i-1}^{n}(x_i - x_m)(y_i - y_m)}{\sqrt{\frac{1}{n-1}\sum_{i-1}^{n}(x_i - x_m)^2}\ \sqrt{\frac{1}{n-1}\sum_{i-1}^{n}(y_i - y_m)^2}}$$

El algoritmo mide el nivel de similitud que existe entre dos matrices, realizando una comparación atributo a atributo de cada serie temporal. Seguidamente se muestra un ejemplo del cálculo del Coeficiente de Correlación Lineal:

T	a	b	c	d
1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,5
6,1	6,2	6,3	6,4	6,5
7,1	7,2	7,3	7,4	7,5
8,1	8,2	8,3	8,4	8,5
9,1	9,2	9,3	9,4	9,5
10,1	10,2	10,3	10,4	10,5
11,1	11,2	11,3	11,4	11,5
12,1	12,2	12,3	12,4	12,5
13,1	13,2	13,3	13,4,	13,5
14,1	14,2	14,3	14,4	14,5

15,1	15,2	15,3	15,4	15,5

Tabla 2. Serie Temporal #1

T	a	b	c	d
1,5	1,8	1,9	1,3	1,2
2,5	2,8	2,9	2,3	2,2
3,5	3,8	3,9	3,3	3,2
4,5	4,8	4,9	4,3	4,2
5,5	5,8	5,9	5,3	5,2
6,5	6,8	6,9	6,3	6,2
7,5	7,8	7,9	7,3	7,2
8,5	8,8	8,9	8,3	8,2
9,5	9,8	9,9	9,3	9,2
10,5	10,8	10,9	10,3	10,2
11,5	11,8	11,9	11,3	11,2
12,5	12,8	12,9	12,3	12,2
13,5	13,8	13,9	13,3	13,2
14,5	14,8	14,9	14,3	14,2
15,5	15,8	15,9	15,3	15,2

Tabla 3. Serie Temporal #2

Teniendo la ecuación anteriormente explicada, se sustituyen los valores en la misma y se obtiene como resultado final: r = 0,98.

Por lo que en este ejemplo se llega a la conclusión de que las series temporales tienen un coeficiente de 0,98 de similitud (lo que quiere decir que las series temporales se parecen mucho). Teniendo en cuenta el coeficiente de utilidad se va construyendo el árbol con los nodos (clusters) donde se ubicarán todas las instancias. Puede que en algún caso se llegue a la cantidad máxima de cluster, esta está predeterminada por el programador, en ese caso se determina en cuál de los clusters existentes la serie temporal tiene un Coeficiente de Utilidad mayor y se ubica en ese cluster. Para mostrar la información final, que es la que le será de utilidad al investigador se mostrará una ventana con una gráfica, donde se recogen todas las series temporales y el color con el que se muestra es el color del cluster al que pertenece. Al dar clic en una de las series se muestra la información específica referente a esta, además la ventana constará de un botón para mostrar la tabla donde se recogen todos los valores de las variables pertenecientes a la serie temporal, haciendo una consulta a la Base de Datos.

3. Resultados y discusión

En este último capítulo se desarrolla la propuesta de solución para el problema que se plantea. Se describe la función distancia y el formato del fichero donde se recogerá la información. Se define la reimplementación de los algoritmos que se utilizarán en la nueva versión de BioSyS. Se realizarán las pruebas correspondientes para comprobar los resultados que se quieren alcanzar y se efectuará un análisis de los mismos.

El principal propósito del presente trabajo se ha dirigido a investigar si al realizar análisis por Clustering utilizando series temporales se obtenían mejores resultados que utilizando sólo los valores de las variables en el estado final de la serie temporal. Para llegar al resultado principal se obtuvieron otros resultados que contribuyeron a facilitar el análisis de la investigación.

En la versión 1.0 de BioSyS se implementaron 3 algoritmos para realizar Clustering (Simple K-Means, X-Means y CobWeb). Los algoritmos Simple K-Means y X-Means ofrecen resultados muy parecidos, esto se debe a que se encuentran dentro del mismo grupo de algoritmos de Clustering: de particionado y recolocación; CobWeb por otro lado ofrece resultados diferentes y además realiza un mejor Clustering de la misma cantidad de resultados, pues realiza una división más específica, de ahí que se obtenga una mayor cantidad de clusters.

Por estas razones se escogió el algoritmo CobWeb para realizar la investigación, pues el científico no tiene que determinar a priori la cantidad de clusters que se van a crear sino que utilizando Clustering jerárquico el propio algoritmo va creando un árbol (árbol de clasificación) donde cada nodo es un cluster, esto es posible pues se caracteriza porque utiliza aprendizaje incremental, es decir, realiza las agrupaciones instancia a instancia [14].

3.1. Primer resultado: Cambio de vector a matriz

En la versión 1.0 del software BioSyS se realizaba el Clustering teniendo en cuenta el estado final definido por el usuario, o sea, un vector. De acuerdo a la investigación realizada para un mejor rendimiento del software, y optimizar la calidad del resultado que se le muestra al usuario final, se decidió utilizar las series temporales, teniendo en cuenta no sólo el estado final del sistema, sino todas las variaciones que sufre el sistema hasta llegar al estado definido por el usuario como final, es decir, una matriz.

T	A	B	C	D	E
10	4	18	5	40	20

Tabla 4. Vector que define la serie temporal teniendo en cuenta el tiempo final.

T	A	B	C	D	E
1	6	5	6	3	8
2	4	2	7	9	6
3	2	6	8	8	3
4	1	4	3	2	5
5	7	4	7	8	9
6	8	6	2	4	5
7	4	8	9	6	6
8	9	5	3	4	2
9	2	4	6	3	1
10	1	6	5	2	9

Tabla 5. Matriz que define una serie temporal para un rango de tiempo.

3.2. Segundo Resultado: Cambio de fichero .arff a una lista con los identificadores de las series temporales

Anteriormente se generaba un archivo de tipo .arff con la parte de la serie temporal correspondiente al tiempo determinado por el usuario, pues era muy sencillo guardar vectores, pero utilizando series temporales es muy engorroso crear un archivo para guardar tantas series temporales, además la memoria de la computadora no soportaría tanto procesamiento de datos y se determinó que se demoraba mucho tiempo y era un tiempo que se perdía innecesariamente. Por lo que se decidió guardar en memoria una lista con todos los identificadores de las series temporales que contengan el tiempo que el investigador haya solicitado.

3.3. Tercer Resultado: Cambio del criterio de similitud.

En BioSyS 1.0 se usaba como criterio de similitud la distancia euclidiana, pero para el cálculo de matrices no se puede usar esa distancia; por tanto, se usará un parámetro, llamado Coeficiente de Correlación Lineal de Pearson (usado para atributos numéricos) que va de acuerdo con los valores que se guardan de cada simulación en la Base de Datos. Así se resuelve el problema del criterio a utilizar

para que el algoritmo CobWeb ubique a una serie temporal en un determinado cluster. Se determinó [0.7, 1.0] como intervalo de tolerancia para este criterio de similitud.

3.4. Cuarto Resultado: Visualización del resultado del Clustering.

Al concluir el algoritmo de Clustering, para un mejor entendimiento por parte del científico a la hora de interpretar los resultados obtenidos de la Minería de Datos realizada, se muestra al investigador una gráfica donde se representan todas las series temporales. En el gráfico, el color indica en qué cluster fue ubicada cada serie temporal:

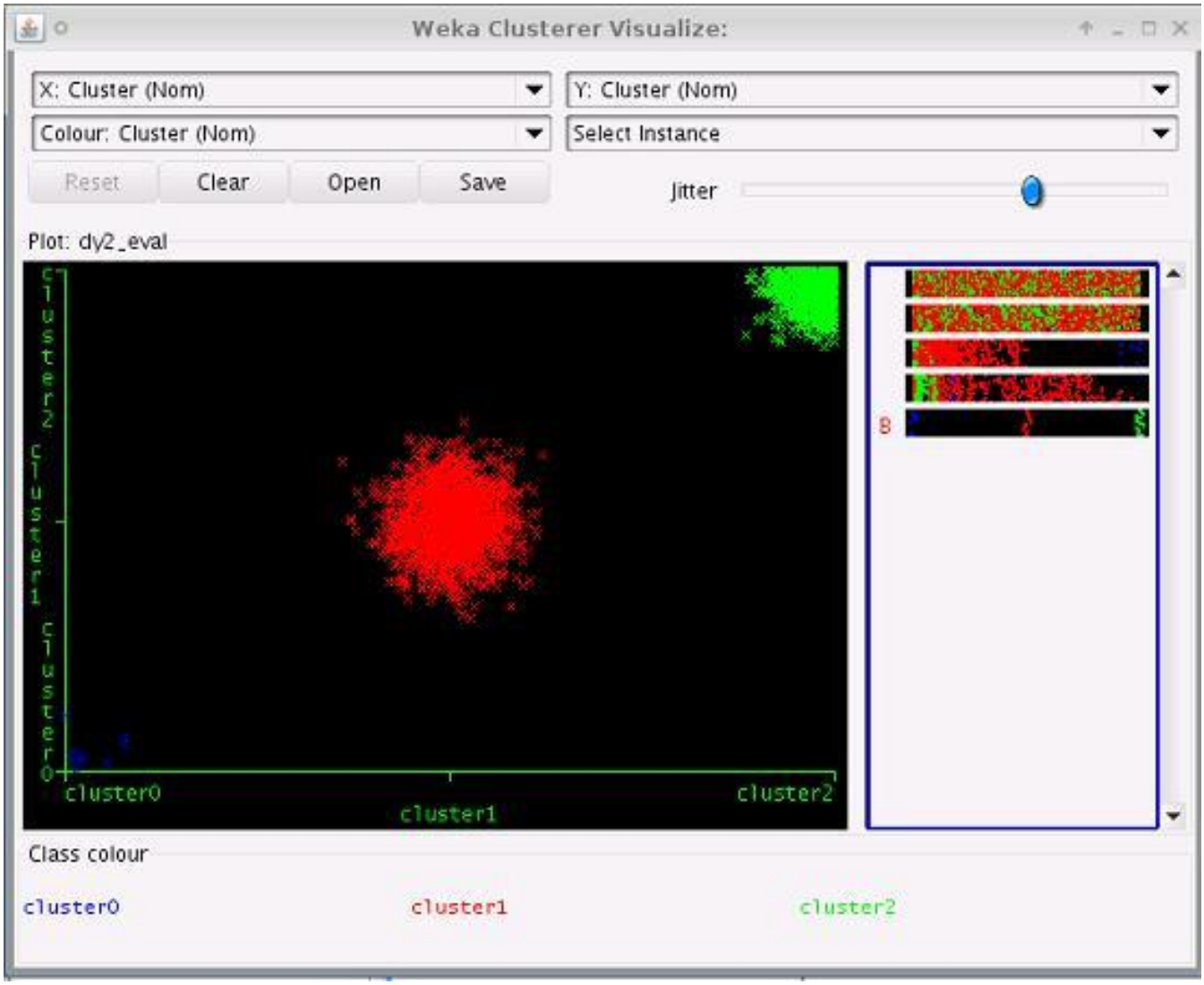

Figura 10. Clustering

Para obtener más información acerca de cada una de las series temporales el científico sólo tiene que dar clic sobre la serie temporal deseada (representada en la Figura 10) y se muestra el comportamiento de las variables en un intervalo de tiempo, desde 0 hasta el tiempo determinado por el usuario como tiempo final (Figura 11). También se le agrega la facilidad de poder obtener la tabla de los valores correspondientes a las variables que conforman el sistema en el caso de que el científico requiera de información más detallada (Figura 12).

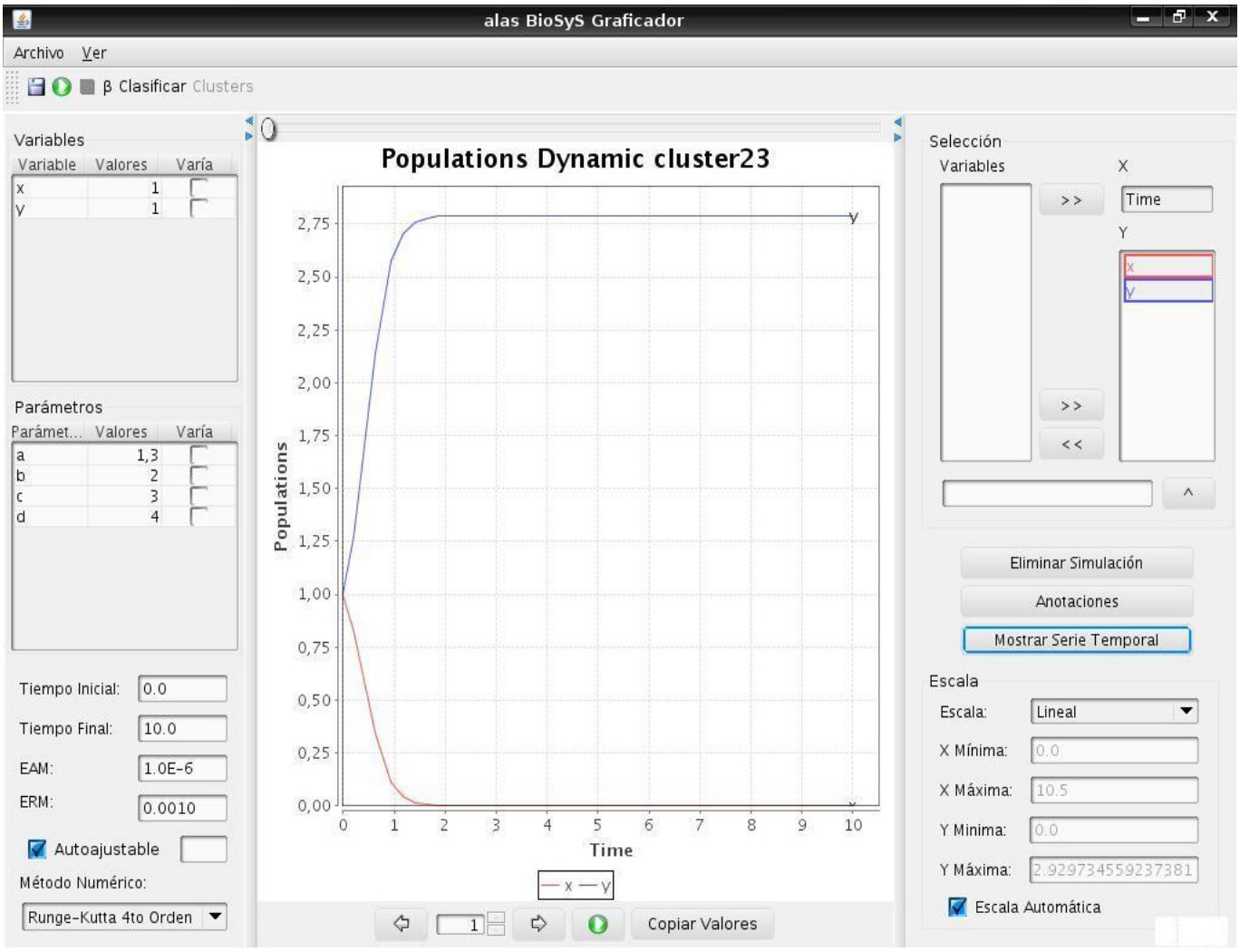

Figura 11. Gráfico de Series Temporales

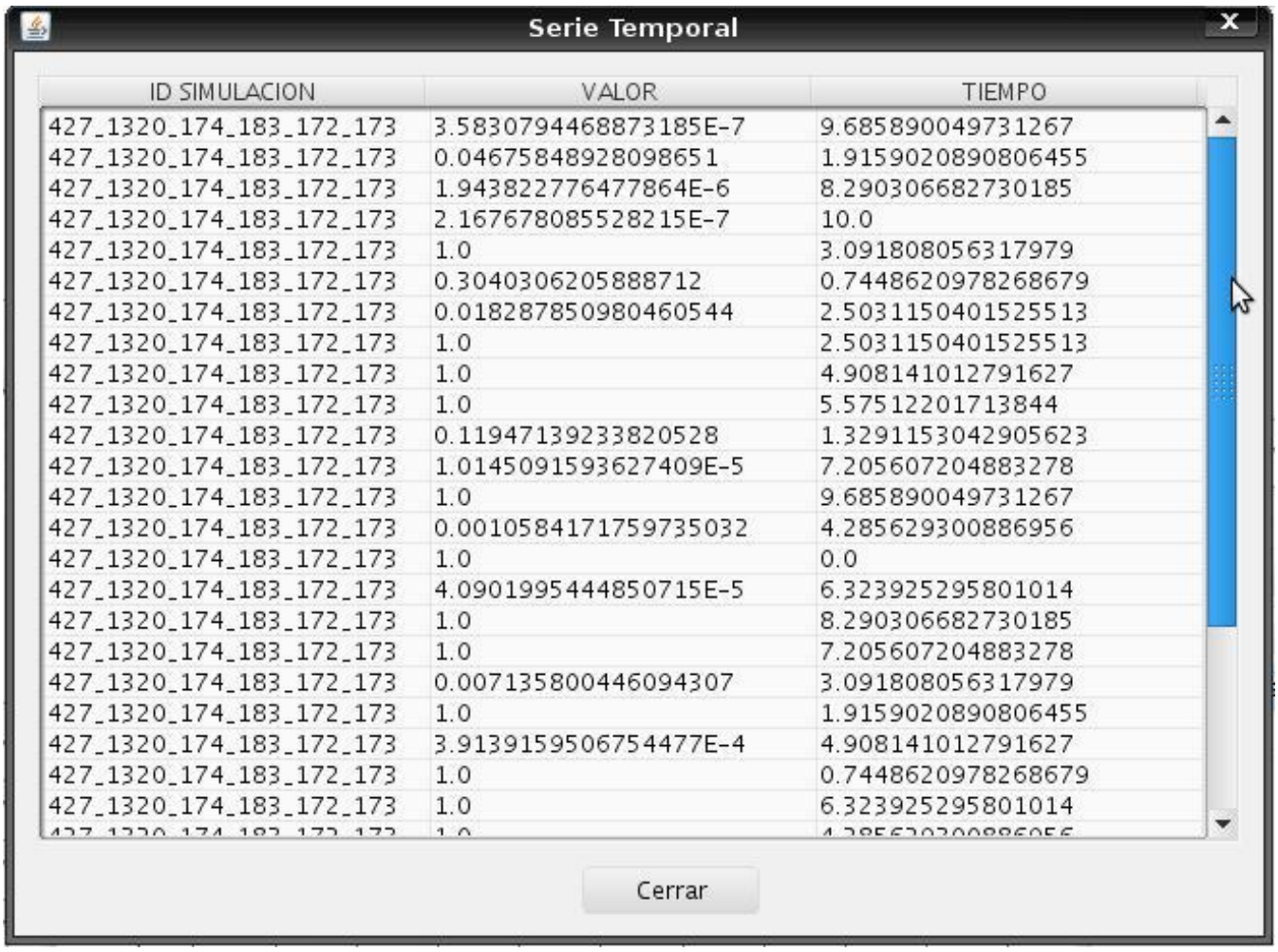

Figura 12. Tabla de valores de la Serie Temporal

3.5. Quinto Resultado: Resultados de las pruebas experimentales.

Se realizó el análisis por Clustering de todas las series temporales pertenecientes al modelo matemático Presa-Depredador. Se utilizó una PC Pentium IV a 2.00 GHz con 994.4 de RAM.

Después de realizar una comparación entre BioSyS 1.0 y BioSyS 2.0 para verificar si el Clustering de series temporales utilizando el algoritmo CobWeb es más específico teniendo en cuenta la similitud entre las series temporales pertenecientes a un mismo cluster, se demostró que en la versión 1.0 de

BioSyS las series temporales correspondientes a un mismo cluster son muy diferentes, no siendo así en la versión 2.0 de BioSyS.

CobWeb en la versión 1.0

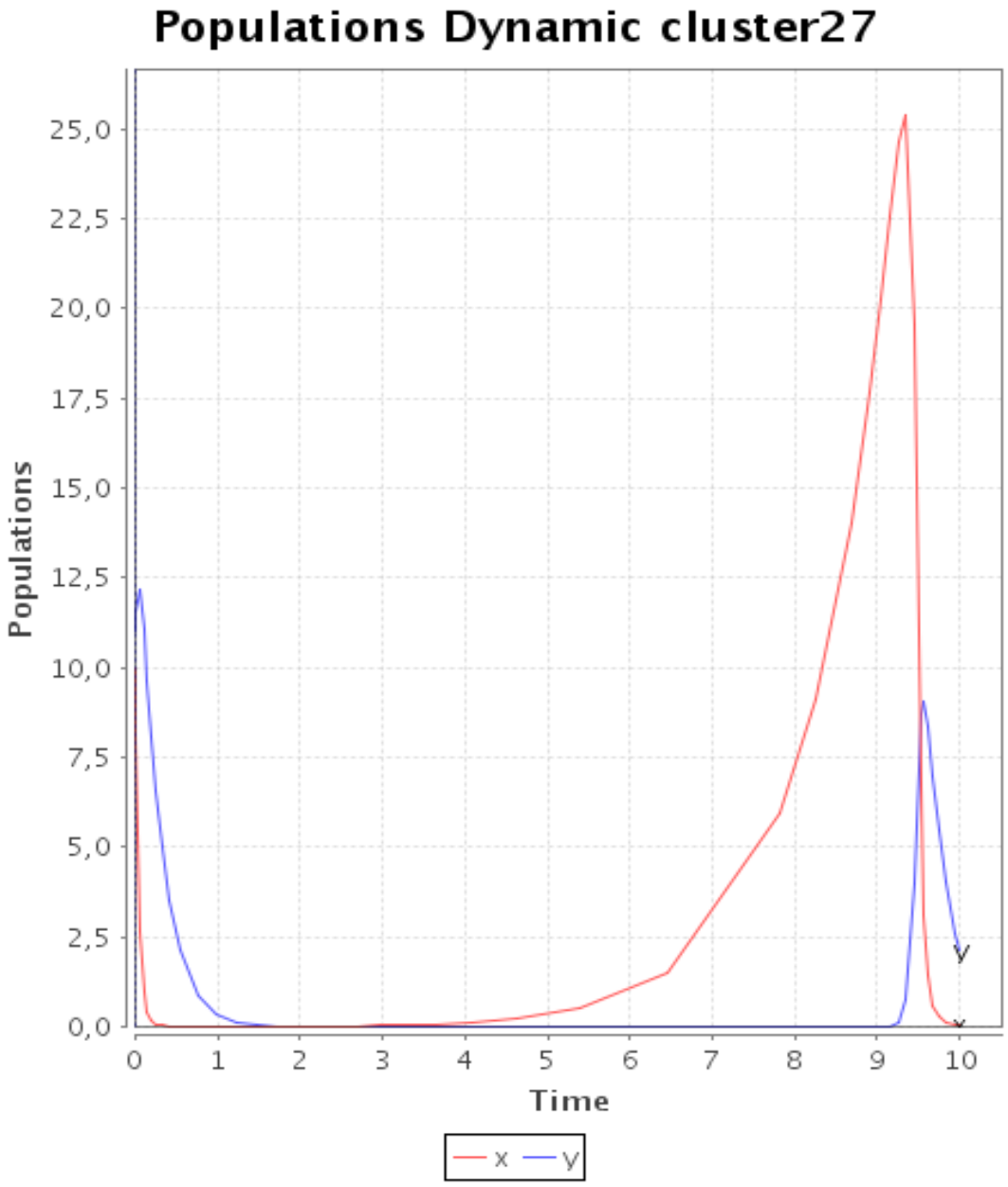

Figura 13. Dinámica de Población 1 en BioSyS 1.0

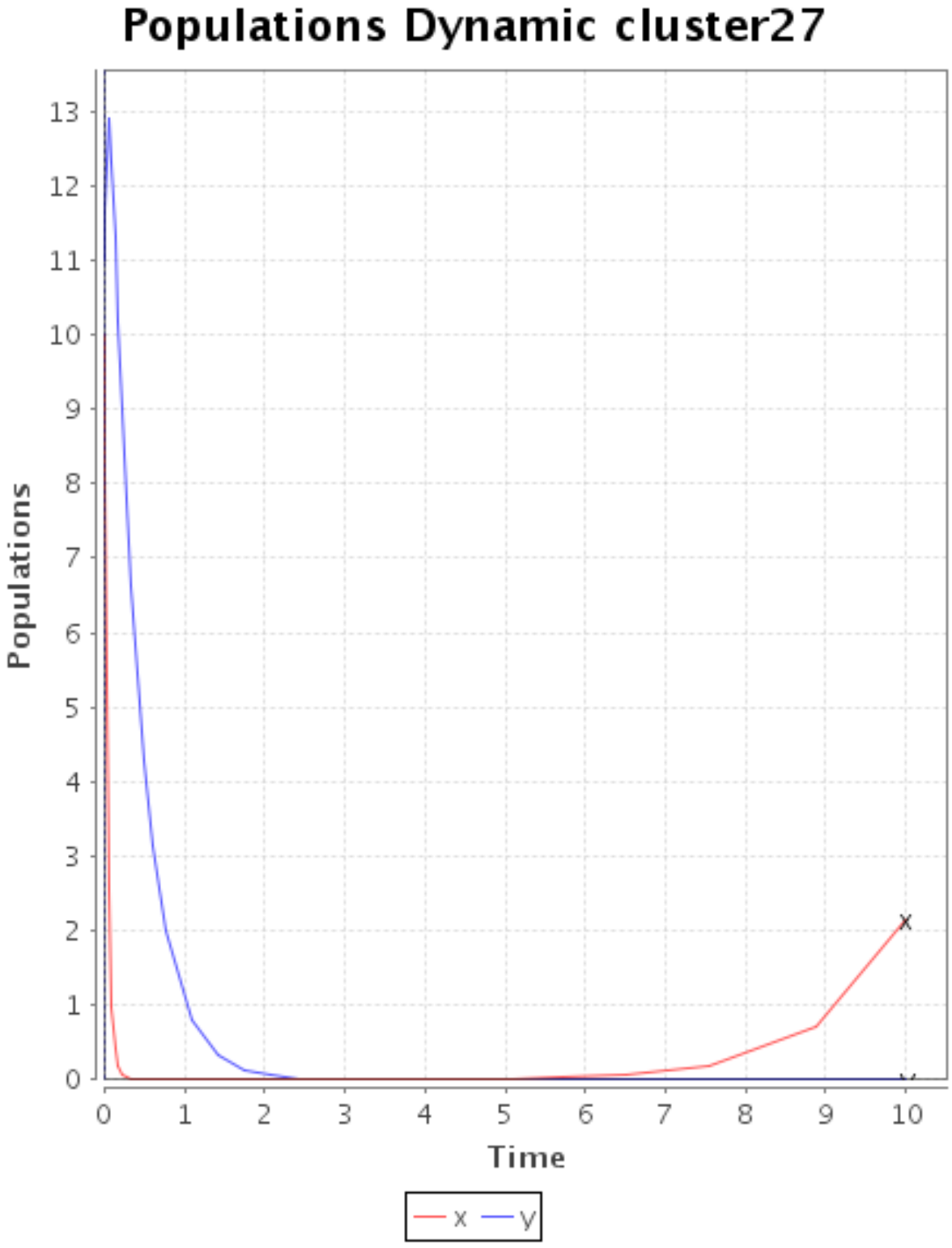

Figura 14. Dinámica de Población 2 en BioSyS 1.0

CobWeb en la versión 2.0

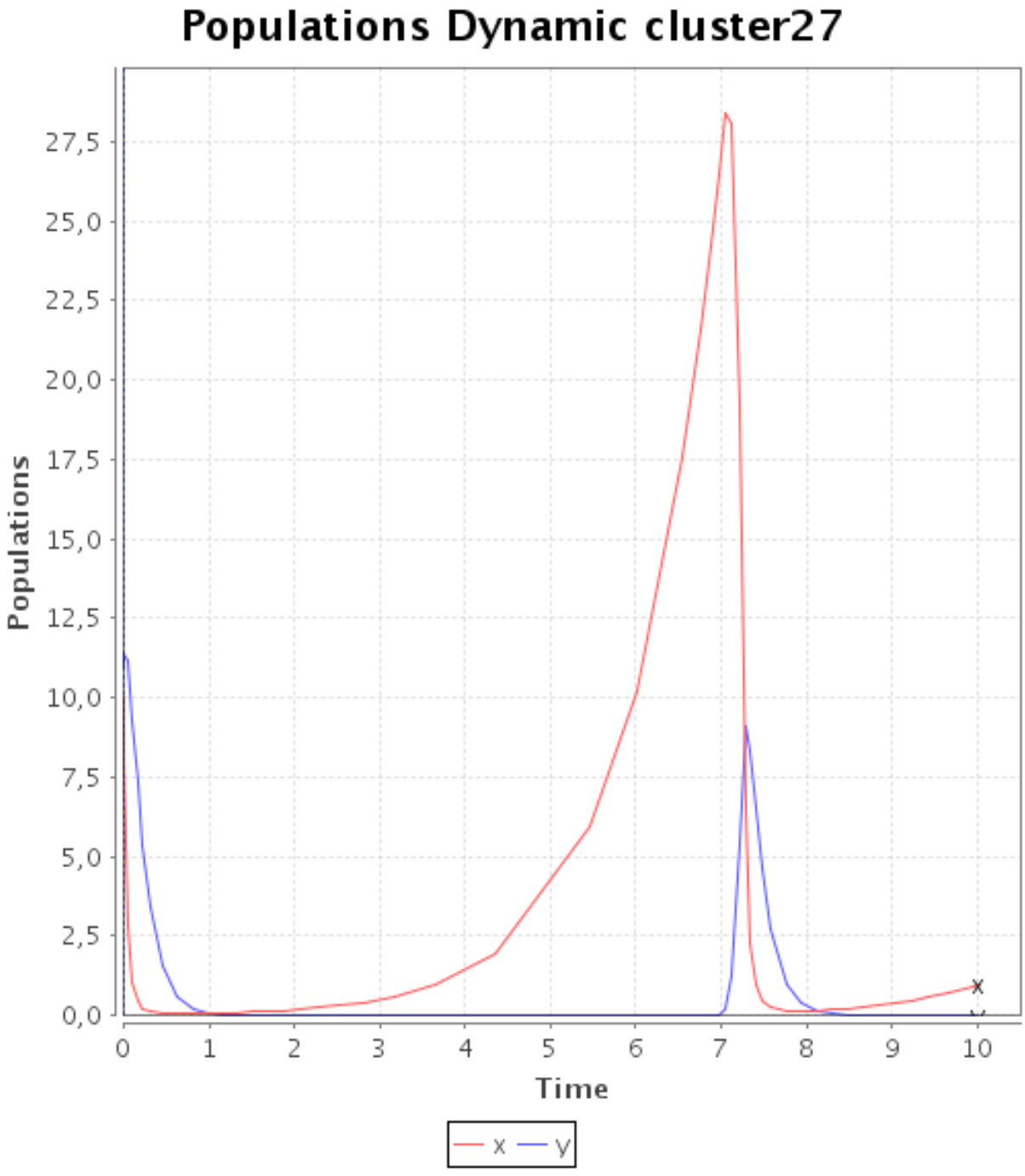

Figura 15. Dinámica de Población 1 en BioSyS 2.0

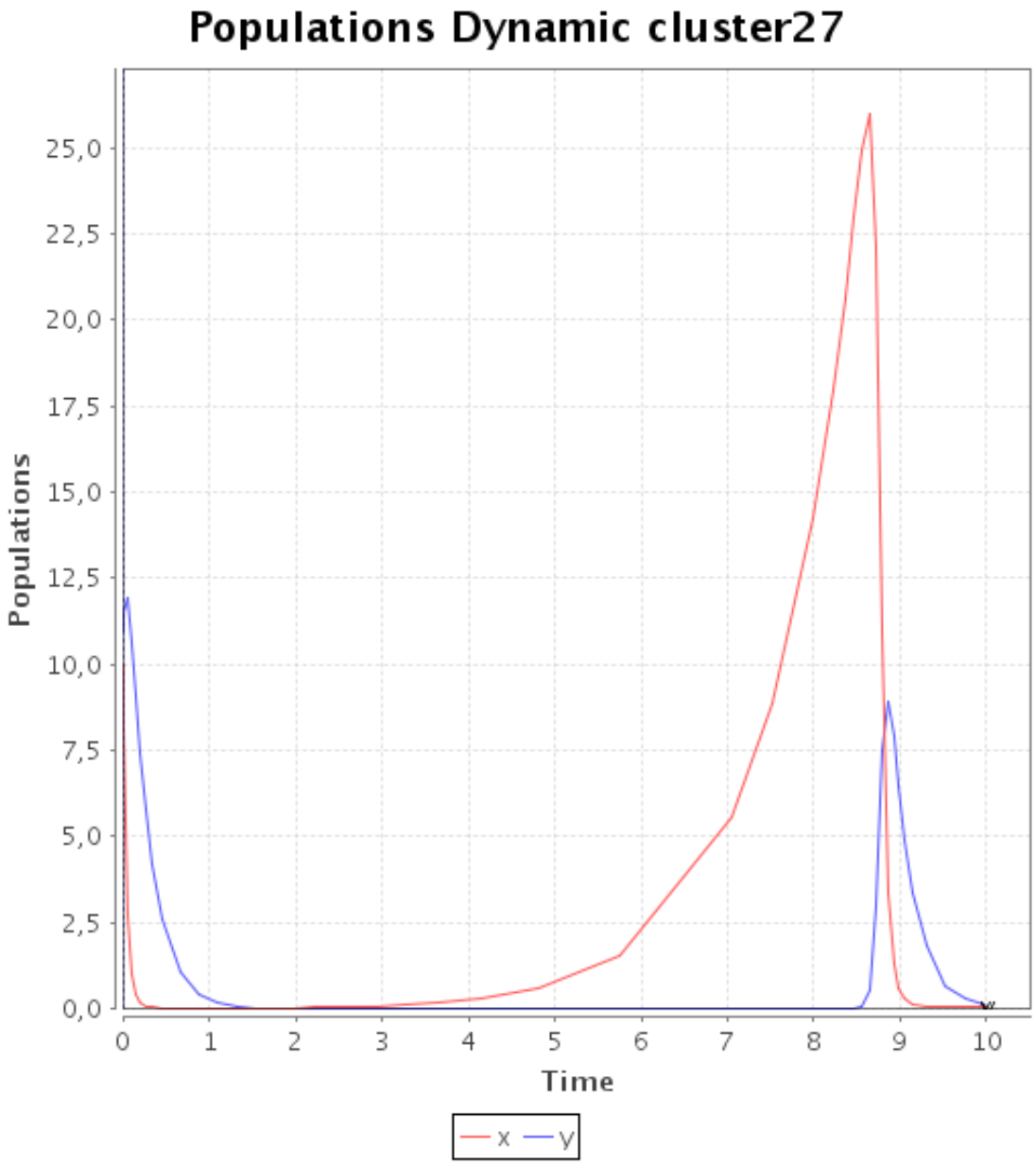

Figura 16. Dinámica de Población 2 en BioSyS 2.0

CONCLUSIONES

- Se definió un nuevo criterio de similitud para la realización de clusters a series temporales.
- Se modificó la implementación del algoritmo CobWeb para la realización de Clustering de series temporales de forma secuencial.
- Se comparó el algoritmo de Clustering (CobWeb) de BioSyS 1.0 con el algoritmo propuesto y se obtuvo mayor similitud entre las series temporales pertenecientes a un mismo cluster. El tiempo para la modificación realizada es mayor con respecto a la versión anterior a la hora de ejecutar el algoritmo, pero es aceptable.

RECOMENDACIONES

- Profundizar más en el tema para encontrar técnicas que permitan el manejo más eficiente de las series temporales.
- Implementar el algoritmo de forma distribuida para así tener un mejor rendimiento de la aplicación en tiempo de ejecución y obtener los resultados en menos tiempo.

BIBLIOGRAFÍA

1. **Acuna, E.** TIPOS DE DATOS Y ATRIBUTOS. [En línea] 2008. [Citado el: 1 de abril de 2009.] http://math.uprm.edu/~edgar/espol2.pdf.

2. Algoritmos de minería de datos (Analysis Services: Minería de datos). [En línea] 2009. [Citado el: 20 de noviembre de 2008.] http://msdn.microsoft.com/es-es/library/ms175595.aspx .

3. **Alvarez, M.A.** Descripción y características de este potente y moderno lenguaje de programación. [En línea] 2001. [Citado el: 3 de marzo de 2009.] http://www.desarrolloweb.com/articulos/497.php .

4. Ancell, R., Gutiérrez, J.M., San-Martín, D., Sordo, C.M. Minería de Datos. Redes Bayesianas y Neuronales. [En línea] [Citado el: 26 de marzo de 2009.] http://www.mdm.unican.es/es/research/mineria_datos

5. **Antolín A., M. and Barcenilla M., M.Á.** Minería de Datos: Intrusiones de Red. [En línea] [Citado el: 26 de febrero de 2009.] http://www.it.uc3m.es/jvillena/irc/practicas/07-08/IntrusionesDeRed.pdf) .

6. Arquitectura de Innovacion.Protagonicemos el Futuro o ¿Porqué es importante Un centro de Inteligencia Artificial para Chile? (Consultado:7/02/2009)(. [En línea] 2007. [Citado el: 7 de febrero de 2009.] http://duranarquitectos.cl/2007/03/18/protagonicemos-el-futuro-o-porqu-es-importante-un-centro-de-inteligencia-artificial-para-chile/ .

7. **Avalo, C.R.** ¿Cómo hacer una tesis? Una guía para noveles investigadores de Educacion Fisica. [En línea] marzo de 2008. [Citado el: 26 de febrero de 2009.] http://www.efdeportes.com/efd118/como-hacer-una-tesis.htm.

8. **Béjar, J.** Aprendizaje Inductivo no Supervisado: COBWEB. [En línea] 2000. [Citado el: 28 de abril de 2009.] http://www.lsi.upc.edu/~bejar/apren/docum/cobweb.ps.gz .

9. **Bertran C., J.** Descripción del Departamento de Biología de Sistemas. (Consultado: 25/02/2009. [En línea] [Citado el: 25 de febrero de 2009.] http://www.uvic.cat/eps/dept/biologiasistemes/es/inici.html .

10. **Cañedo A., R. And Arencibia J., R.** Bioinformática: en busca de los secretos moleculares de la vida. [En línea] 30 de 12 de 2005. [Citado el: 22 de noviembre de 2008.] http://www.bvs.sld.cu/revistas/aci/vol12_6_04/aci02604.htm .

11. **Castro B., N.O.** Evaluacion de la Infraestructura Asociada a Zonas de Desarrollo EÓLICO en el Sector Norte de SIC. [En línea] 2007. [Citado el: 20 de marzo de 2009.] http://146.83.6.25/literatura/memorias_tesis/memoria_Nicolas_Castro.pdf .

12. Características de C#. [En línea] 2007. [Citado el: 3 de marzo de 2009.]

https://www.ibercom.com/soporte/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=935&nav=0,78,121 .

13. **Celis, S. and Musicant,D.R.** *Weka-Parallel:Machine Learning in Parallel.* Northfield : s.n.

14. Correlacion. [En línea] 2003. [Citado el: 7 de abril de 2009.] http://dieumsnh.qfb.umich.mx/estadistica/correlacion.htm).

15. **Cuevas M., R.J.** Matemáticas para la toma de decisiones. [En línea] [Citado el: 2 de marzo de 2009.] http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r24492.PDF.

16. Dataprix. [En línea] 30 de 4 de 2007. [Citado el: 13 de noviembre de 2008.] http://www.dataprix.com/herramientas-de-data-mining.

17. **Demidowitsch, B. P., Maron, I. A. y Schuwalowa., E. S.** Solución numérica de las ecuaciones diferenciales. [En línea] 1980. [Citado el: 12 de noviembre de 2008.] http://www.sc.ehu.es/sbweb/fisica_/numerico/diferencial/ecuacion_diferencial.xhtml.

18. Eclipse Home Page. [En línea] 2009. [Citado el: 16 de noviembre de 2008.] http://www.eclipse.org/.

19. **Ferreira, G.,Araujo,R.,Orair,G.,Gonc¸alves ,L.,Guedes, D.,Ferreira,R.,Furtado,V.,Meira W.Jr.** *Paralelizac¸ ˜ao eficiente de um algoritmo de agrupamento hierarquico.* Belo Horizonte,Brazil : s.n.

20. **F.Sanjuán, M. A.** Más sobre Biología de Sistemas. [En línea] 2006. [Citado el: 2 de marzo de 2009.] http://weblogs.madrimasd.org/complejidad/archive/2006/06/09/29137.aspx.

21. **García J., M. and Álvarez S., A.** Análisis de Datos en WEKA – Pruebas de Selectividad. [En línea] [Citado el: 1 de marzo de 2009.] http://www.it.uc3m.es/jvillena/irc/practicas/06-07/28.pdf.

22. **Garre, M., y otros.** Comparación de diferentes algoritmos de Clustering en la estimación de coste en el desarrollo de software. [En línea] 2007. [Citado el: 30 de abril de 2009.] http://www.ati.es/IMG/pdf/GarreVol3Num1.pdf.

23. **Gondar N., J.E.** Análisis Cluster. [En línea] 2000. [Citado el: 28 de febrero de 2009.] http://www.estadistico.com/arts.html?20001023-2.

24. **Gonzalez-Matesanz, F.J., J.Celada, A.Dalda,R.Quiros.** APLICACION WEB CLIENTE-SERVIDOR DE CALCULOS GEODÉSICOS DEL INTITUTO GEOGRÁFICO NACIONAL. [En línea] 2004. [Citado el: 20 de febrero de 2009.] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=515.

25. **González S., J.A.** Introducción a C#, Origen y necesidad de un nuevo lenguaje. [En línea] 2006. [Citado el: 3 de marzo de 2009.] http://www.devjoker.com/asp/ver_contenidos.aspx?co_contenido=125.

26. **Granger, C.** Analisis de Series Temporales,Cointegracion y Aplicaciones. [En línea] 2004. [Citado el: 28 de abril de 2009.] http://www.revistaasturianadeeconomia.org/raepdf/30/GRANGER.pdf.

27. **Hernández M., E.** *Cómo escribir una tesis.* 2006. [Citado el: 3 de diciembre de 2008.]

28. **Hernandez V., E.** Algoritmo de Clustering basado en entropıa para descubrir grupos en atributos de tipo mixto. [En línea] 2006. [Citado el: 26 de febrero de 2009.] http://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2006/tesisEdnaHernandez.pdf.

29. **Isaac, Q. And Simón,A.** Introducción al Diseño de Experimentos para el Reconocimiento de Patrones. Capítulo 7: Herramientas. [En línea] 2004. [Citado el: 27 de abril de 2009.] http://www.infor.uva.es/~isaac/doctorado/Cap08_Herramientas.pdf.

30. **Ji, W.** Data Mining by Clementine . [En línea] [Citado el: 3 de marzo de 2009.] http://www.comp.rgu.ac.uk/staff/chb/teaching/cmm510/clementine_lab_handout.pdf..

31. **Lahoz-Beltra, R.** Bioinformática: Simulación, Vida Artificial e Inteligencia Artificial. [En línea] 2007. [Citado el: 22 de noviembre de 2008.] http://bioinformatica.net/libros/libro1/index.html.

32. La Simulación Computarizada y Estrategias de Manejo en el Ambito Agrícola. [En línea] 2006. [Citado el: 7 de febrero de 2009.] http://www.azete.com/view/37786.

33. **Macías R., M.** Técnicas de Minería de Datos para la Retención de Clientes en el Sector Asegurador. [En Línea] 2008 [Citado el: 12 de mayo de 2009.] http://www.cnsf.gob.mx/Eventos/Premios/2008%20Seguros/ANIVDELAREV.pdf

34. Manual de Java. [En línea] 2008. [Citado el: 21 de febrero de 2009.] http://www.webtaller.com/manual-java/caracteristicas-java.php.

35. **Muñoz, E.** La Biología de sistemas: ¿hacia un círculo virtuoso de la investigación biomédica? . [En línea] 2009. [Citado el: 26 de enero de 2009.] http://www.institutoroche.es/doc.php?op=biotecnologia2&id=29&menu=2.

36. **Narro R., A.E.** Aplicación de algunos Modelos Matemáticos a la toma de decisiones. [En línea] 1996. [Citado el: 2 de marzo de 2009.] http://www.xoc.uam.mx/~polcul/pyc06/183-198.pdf.

37. Oracle. [En línea] [Citado el: 13 de noviembre de 2008.] http://www.oracle.com/global/lad/solutions/business_intelligence/data-mining.html.

38. Oracle Data Mining. [En línea] [Citado el: 28 de febrero de 2009.] http://www.oracle.com/global/lad/solutions/business_intelligence/data-mining.html).

39. Programas en Java, manuales en Java. [En línea] 2007 . [Citado el: 11 de noviembre de 2008.] http://todojava.awardspace.com/ .

40. ¿Qué es Visual Studio? [En línea] 2009. [Citado el: 25 de marzo de 2009.] http://msdn.microsoft.com/es-es/vstudio/products/default.aspx.

41. **Rodríguez C., Y. And Noa G.,Y.** *BioSyS: Implementación del Módulo de Análisis .* UCI.Cuidad de la Habana(Págs 11-12) : s.n., 2008.

42. **Sánchez, O.** Algoritmos de Clustering. [En línea] [Citado el: 4 de marzo de 2009.]

http://omarsanchez.net/agrupamiento.aspx.

43. Tutorial de oracle. [En línea] 2008. [Citado el: 13 de noviembre de 2008.] http://www.oracle.com/global/es/database/docs/oracle_data_mining.pdf.

44. Un Enfoque al Lenguaje de Programación Java. [En línea] 2006. [Citado el: 20 de febrero de 2009.] http://www.nexos-software.com.co/Articulo_25.htm.

45. **Velásquez V., S.** Programación, ¿Qué clase de programas y aplicaciones se pueden crear usando C y C++? [En línea] [Citado el: 26 de enero de 2009.] http://viels.wordpress.com/programacion/.

46. Ventajas y Desventajas: Comparación de los Lenguajes C, C++ y Java. [En línea] 2006. [Citado el: 20 de febrero de 2009.] http://www.americati.com/doc/ventajas_c/ventajas_c.html.

47. Visual C#. [En línea] 2009. [Citado el: 1 de marzo de 2009.] http://msdn.microsoft.com/es-es/vcsharp/default.aspx.

48. Zator Systems: Tecnología de la información para el conocimiento. El lenguaje C++. . [En línea] [Citado el: 3 de marzo de 2009.] http://www.zator.com/Cpp/E1_2.htm#TOP.

REFERENCIAS BIBLIOGRÁFICAS

[1]Algoritmos de minería de datos (Analysis Services: Minería de datos).2009. (Consultado: 20/11/2008) (Disponible en: http://msdn.microsoft.com/es-es/library/ms175595.aspx)

[2]Álvarez, M.A. Descripción y características de este potente y moderno lenguaje de programación.2001 (Consultado: 3/03/2009) (Disponible en: http://www.desarrolloweb.com/articulos/497.php)

[3]Antolín A., M. and Barcenilla M., M.Á. Minería de Datos: Intrusiones de Red. (Consultado: 26/02/2009) (Disponible en: http://www.it.uc3m.es/jvillena/irc/practicas/07-08/IntrusionesDeRed.pdf)

[4]Arquitectura de Innovacion.Protagonicemos el Futuro o ¿Porqué es importante Un centro de Inteligencia Artificial para Chile?2007. (Consultado: 7/02/2009) (Disponible en: http://duranarquitectos.cl/2007/03/18/protagonicemos-el-futuro-o-porqu-es-importante-un-centro-de-inteligencia-artificial-para-chile/)

[5]Béjar, J.Aprendizaje Inductivo no Supervisado: COBWEB. 2000. (Consultado: 28/04/2009).(Disponible en: http://www.lsi.upc.edu/~bejar/apren/docum/cobweb.ps.gz)

[6]Bertran C., J. Descripción del Departamento de Biología de Sistemas. (Consultado: 25/02/2009) (Disponible en: http://www.uvic.cat/eps/dept/biologiasistemes/es/inici.html)

[7]Castro B., N.O. Evaluacion de la Infraestructura Asociada a Zonas de Desarrollo EÓLICO en el Sector Norte de SIC. 2007. (Consultado: 20/03/2009) (Disponible en: http://146.83.6.25/literatura/memorias_tesis/memoria_Nicolas_Castro.pdf)

[8]Características de C#.2007. (Consultado: 3/03/2009) (Disponible en: https://www.ibercom.com/soporte/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=935&nav=0,78,121)

[9]Correlación.2003 (Consultado: 7/04/2009). (Disponible en: http://dieumsnh.qfb.umich.mx/estadistica/correlacion.htm)

[10]Cuevas M., R.J.Matemáticas para la toma de decisiones. (Consultado: 2/03/2009) (Disponible en: http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r24492.PDF)

[11]Demidowitsch, B. P.; I. A. Maron; E. S Schuwalowa. Solución numérica de las ecuaciones diferenciales.1980. (Consultado: 12/11/2009). (Disponible en: http://www.sc.ehu.es/sbweb/fisica_/numerico/diferencial/ecuacion_diferencial.xhtml)

[12]F.Sanjuán, M. A.Más sobre Biología de Sistemas. 2006. (Consultado: 2/03/2009) (Disponible en: http://weblogs.madrimasd.org/complejidad/archive/2006/06/09/29137.aspx)

[13]García J., M. and Álvarez S., A. Análisis de Datos en WEKA – Pruebas de Selectividad. (Consultado: 1/03/2009) (Disponible en: http://www.it.uc3m.es/jvillena/irc/practicas/06-07/28.pdf)

[14]Garre, M.; J.J.Cuadrado; M.A.Sicilia; D.Rodríguez; R.Rejas. Comparación de diferentes algoritmos de Clustering en la estimación de coste en el desarrollo de software.2007. (Consultado: 30/04/2009.(Disponible en: http://www.ati.es/IMG/pdf/GarreVol3Num1.pdf)

[15]Gondar N., J.E. Análisis Cluster.2000. (Consultado: 28/02/2009) (Disponible en: http://www.estadistico.com/arts.html?20001023-2)

[16]Gonzalez-Matesanz, F.J., J.Celada, A.Dalda, R.Quiros. APLICACION WEB CLIENTE-SERVIDOR DE CALCULOS GEODÉSICOS DEL INTITUTO GEOGRÁFICO NACIONAL.2004. (Consultado: 20/2/2009). (Disponible en: http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=515)

[17]González S., J.A. Introducción a C#, Origen y necesidad de un nuevo lenguaje.2006 (Consultado: 3/03/2009) (Disponible en: http://www.devjoker.com/asp/ver_contenidos.aspx?co_contenido=125)

[18]Granger, C. Análisis de Series Temporales, Cointegración y Aplicaciones.2004. (Consultado: 28/04/2009). (Disponible en: http://www.revistaasturianadeeconomia.org/raepdf/30/GRANGER.pdf)

[19]Hernandez V., E. Algoritmo de Clustering basado en entropıa para descubrir grupos en atributos de tipo mixto. 2006. (Consultado: 26/02/2009) (Disponible en: http://www.cs.cinvestav.mx/Estudiantes/TesisGraduados/2006/tesisEdnaHernandez.pdf)

[20]Isaac, Q. And Simón, A. Introducción al Diseño de Experimentos para el Reconocimiento de Patrones. Capítulo 7: Herramientas. 2004(Consultado: 27/04/2009) (Disponible en: http://www.infor.uva.es/~isaac/doctorado/Cap08_Herramientas.pdf)

[21]La Simulación Computarizada y Estrategias de Manejo en el Ambito Agrícola.2006. (Consultado: 7/02/2009). (Disponible en: http://www.azete.com/view/37786)

[22]Manual de Java.2008. (Consultado: 21/2/2009).(Disponible en: http://www.webtaller.com/manual-java/caracteristicas-java.php)

[23]Muñoz, E. La Biología de sistemas: ¿hacia un círculo virtuoso de la investigación biomédica? .2009. (Consultado: 10/11/2008). (Disponible en: http://www.institutoroche.es/doc.php?op=biotecnologia2&id=29&menu=2)

[24]Narro R., A.E. Aplicación de algunos Modelos Matemáticos a la toma de decisiones.1996 (Consultado: 2/03/2009) (Disponible en: http://www.xoc.uam.mx/~polcul/pyc06/183-198.pdf)

[25]Oracle Data Mining. (Consultado: 28/02/2009) (Disponible en: http://www.oracle.com/global/lad/solutions/business_intelligence/data-mining.html)

[26]¿Qué es Visual Studio? 2009.(Consultado:25/03/2009) (Disponible en: http://msdn.microsoft.com/es-es/vstudio/products/default.aspx)

[27]Rodríguez C., Y. And Noa G., Y. BioSyS: Implementación del Módulo de Análisis.UCI.Ciudad de la Habana. Cuba. Págs. 11-12. (Consultado: 14/02/2009)

[28]Sánchez, O.Algoritmos de Clustering. (Consultado: 14/03/2009). Disponible en: http://omarsanchez.net/agrupamiento.aspx

[29]Un Enfoque al Lenguaje de Programación Java.2006. (Consultado: 20/2/2009). Disponible en: http://www.nexos-software.com.co/Articulo_25.htm

[30]Velásquez V., S. Programación, ¿Qué clase de programas y aplicaciones se pueden crear usando C y C++? (Consultado: 26/1/2009). (Disponible en: http://viels.wordpress.com/programacion/)

[31]Ventajas y Desventajas: Comparación de los Lenguajes C, C++ y Java.2006. 2006. (Consultado: 20/2/2009). (Disponible en: http://www.americati.com/doc/ventajas_c/ventajas_c.html)

[33]Visual C#. 2009. (Consultado: 1/03/2009) (Disponible en: http://msdn.microsoft.com/es-es/vcsharp/default.aspx)

[33]Zator Systems: Tecnología de la información para el conocimiento. El lenguaje C++. (Consultado: 3/03/2009) (Disponible en: http://www.zator.com/Cpp/E1_2.htm#TOP)

GLOSARIO DE TÉRMINOS

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

Biología de Sistemas (BS): Área de investigación científica que se preocupa del estudio de procesos biológicos usando un enfoque sistémico.

Sistema Biológico (SB): Es un conjunto de órganos y estructuras análogas que trabajan en conjunto para cumplir alguna función en el ser vivo.

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

Centroide: Centro de masa de un objeto con densidad uniforme. El centroide de un cluster se define como el punto equidistante de los objetos pertenecientes a dicho cluster, es como decir el punto de equilibrio.

Clustering: Es un conjunto de técnicas destinadas a realizar clasificaciones de datos en un entorno no supervisado disponiendo de datos que se quieren agrupar y separar por clases.

Cluster: Clases en las que se van a separar los datos (para el presente trabajo: series temporales) de acuerdo a un criterio de similitud (para el presente trabajo: coeficiente de correlación lineal).

Inteligencia Artificial: Es la rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos.

Minería de datos: Es un proceso de extracción de información y búsqueda de patrones de comportamiento que permanecen ocultos entre grandes cantidades de información.

Series Temporales: Se define como una sucesión ordenada, a través del tiempo, de un conjunto de variables objeto de estudio.

Simulación: Es imitar una situación del mundo real en forma matemática.

Software libre: Software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Redes Neuronales Artificiales (RNA): Las Redes Neuronales Artificiales son una rama de la Inteligencia Artificial que intentan reproducir el comportamiento del cerebro humano.