

Universidad de las Ciencias Informáticas



Título: Metodología para el análisis de estabilidad de Sistema de Ecuaciones Diferenciales n-dimensionales

Tesis en opción al grado de Máster en Bioinformática

Autor: Ing. Yunet González Mulet

Ciudad de La Habana, Cuba
2010

Tesis en opción al grado de Máster en
Bioinformática

Título:

Metodología para el análisis de estabilidad
de Sistema de Ecuaciones Diferenciales
n-dimensionales

Autor: Ing. Yunet González Mulet

Universidad de las Ciencias Informáticas (UCI)

ygonzalezmu@uci. cu

Tutor: MSc Noel Moreno Lemus

Universidad de las Ciencias Informáticas (UCI)

noel@uci. cu

Consultante:

MSc: Mario Pupo Meriño

Universidad de las Ciencias Informáticas (UCI)

mpupom@uci. cu

**Ciudad de La Habana, Cuba
2010**

Dedicatoria

A mis padres Elvia y Roberto.

Agradecimientos

- ***A mi tutor por tener tantas ideas y guiarme desde que me conoció.***
- ***A Edel por ayudarme a cualquier hora, por su apoyo inmenso.***
- ***A Yenier y a Leandro que estuvieron conmigo trabajando aún en los fines de semanas.***
- ***A mis compañeros maestrantes porque siempre nos dimos fuerzas para seguir.***
- ***A Ariel, Elvismary, Lilianne, Marleisy y Yadira por su preocupación por mí.***
- ***A las excelentes críticas y a la ayuda oportuna de Alina, Mario, Orlando, Aurelio y Ballester.***
 - ***A Reina, por estar siempre, su amistad inmensa me acompaña.***
 - ***A Alexis, claro... si es mi hermano.***
- ***A mi mamá y a mi papá porque nunca podré igualarme en todo lo que me han dado.***
- ***A todos los que me ayudaron durante todo este tiempo.***

Resumen

En el presente trabajo se propone una metodología para el estudio de la estabilidad en sistemas dinámicos n-dimensionales. La misma se basa en el análisis de estabilidad según el espectro de los exponentes de Lyapunov, pero incorpora el análisis de múltiples series temporales a la vez y distribuye el cómputo sobre la plataforma de tareas distribuidas T-arenal, con el objetivo de reducir los tiempos de cálculo.

Del proceso de análisis se genera un fichero con las dinámicas clasificadas respecto a su estabilidad. Teniendo en cuenta que el análisis puede haber incluido más de dos parámetros de control, el uso de la Minería de Datos, específicamente la técnica de los Árboles de Decisión, permite la creación y visualización de las reglas que llevan a comportamientos estables o inestables a partir de los resultados del análisis de estabilidad. La graficación en dos dimensiones es adicionalmente otra manera de interpretar los resultados que fue incluida en la metodología y muestra el comportamiento obtenido de un parámetro contra otro.

Se aplicó la metodología a un caso de estudio reportado en la literatura y se comprobó que se obtenía la misma clasificación de estabilidad o inestabilidad. Por otro lado al realizar el análisis en una cantidad de series temporales, el tiempo en que se obtuvo el resultado fue considerablemente pequeño, de acuerdo a su complejidad.

Abstract

The present work intends a methodology for the study of the stability in n-dimensional dynamic systems. The same one is based on the analysis of stability according to the spectrum of the exponents of Lyapunov, but it incorporates the analysis of multiple time series at the same time and it distributes the computation on the platform of tasks distributed T-arenal, with the objective of reducing the times of calculation.

Of the analysis process is generated in a file with the dynamics classified according to its stability. Keeping in mind that the analysis can have included more than two control parameters, the use of the Data Mining, specifically the technique of the Trees of Decision, allows the creation and visualization of the rules that take to stable or unstable behaviors starting from the results of the analysis of stability. The graphics in two dimensions are another way to interpret the results additionally that it was included in the methodology and sample the obtained behavior of a parameter against another.

The methodology was applied to a case of study reported in the literature and it was proven that it was obtained the same classification of stability or uncertainty. On the other hand when carrying out the analysis in a quantity of time series, the time in that the result was obtained was considerably small, according to its complexity.

Palabras claves: series temporales, estabilidad, exponente de Lyapunov.

Keywords: time series, stability, exponent of Lyapunov.

Índice

Introducción	1
Capítulo 1: Revisión Bibliográfica	6
1.1. Los Sistemas Dinámicos	6
1.2. Modelación matemática.....	7
1.2.1. Condiciones iniciales y parámetros	8
1.3. Series Temporales	9
1.4. Análisis de series temporales	10
1.4.1 Análisis de Estabilidad.....	10
1.4.2 Selección del método para el análisis de Estabilidad.....	14
1.5. Software para el análisis de series temporales	16
1.5.1. Copasi	16
1.5.2. TISEAN	17
1.5.3. Matlab.....	17
Conclusión Parcial:.....	18
1.6. Sistema de cómputo distribuido.....	19
1.6.1. Computación voluntaria.....	19
1.6.2. Plataformas existentes	20
1.7. Minería de Datos	22
1.8. Minería de datos en el estudio de sistemas dinámicos	23
Capítulo 2: Materiales y Métodos	25
2.1 Materiales.....	25
2.1.1 Lenguaje Java	25
2.1.2. Netbeans.....	26
2.1.3. BioSyS versión 1.0	26
2.1.4. Usando Matlab en BioSyS versión 1.0	27
2.1.5. Plataforma de Tareas Distribuidas T-arenal.....	28
2.1.6. Weka	28
2.2 Métodos.....	30
2.2.1. Espectro de Lyapunov.....	30
2.2.2. Árboles de decisión. Algoritmo J-48	31
2.3. Modelos matemáticos utilizados.....	33
2.3.1. Modelo de Lorenz.....	33

2.3.2. Modelo Lotka-Volterra (Presa-Depredador)	33
2.3.3. Dinámica de las células T CD4 + y el virus HIV_1	34
2.3.4. ICMODEL	35
2.3.5. Modelo de la Interleuquina (IL-2).....	36
Capítulo 3: Resultados y Discusión	38
3.1. Algoritmo distribuido para el cálculo de los exponentes de Lyapunov para múltiples series temporales n-dimensionales.....	38
3.1.1. Algoritmo distribuido para el cálculo del espectro de los exponentes de Lyapunov.....	40
3.1.2. Tiempos de procesamiento	41
3.2. Minería de Datos para el análisis de estabilidad	42
3.2.1. Algoritmo para el análisis de los resultados de la estabilidad mediante técnicas de minería de datos	43
3.2.2. Aplicación del algoritmo de Minería de Datos a un caso de estudio	44
3.3. Análisis del comportamiento de la estabilidad a partir de dos parámetros.....	45
3. 4. Metodología propuesta.....	46
3.5. Aplicación de la metodología propuesta en un caso de estudio ..	49
3.5.1. Prueba No.1 realizada a la metodología	49
3.5.2. Prueba No.2 realizada a la metodología	50
3.6. Metodología para el análisis de estabilidad de Sistema de Ecuaciones Diferenciales n-dimensionales incorporada dentro del software BioSyS 1.0	52
Conclusiones	54
Recomendaciones	55
Bibliografía.....	56
Glosario	62
Anexos.....	63
Anexo 1	63
Anexo 2	72
Anexo 3	74

Introducción

En los últimos cincuenta años el estudio de los sistemas dinámicos ha aportado valioso conocimiento en ciencias como la Biología, la Química y la Física, por solo citar algunas [1]. De manera general un sistema dinámico es aquel que su estado varía a medida que transcurre el tiempo, es decir, evolucionan en el tiempo a partir de un grupo de reglas que para algunos fenómenos suelen ser no lineales.

Comúnmente la forma de representar un sistema dinámico continuo es mediante un Sistema de Ecuaciones Diferenciales (SED), estas ecuaciones muy pocas veces se pueden resolver analíticamente, por ello se utilizan métodos numéricos, los cuales son muy conocidos y se encuentran implementados en numerosos software. Como resultado de la resolución del SED, se obtiene una simulación que constituye un conjunto de datos numéricos medidos en el tiempo. A partir del comportamiento de dichas soluciones se puede extraer información cualitativa sobre el comportamiento del sistema.

En un sistema dinámico si las funciones no son constantes, las soluciones del sistema definen una curva en el plano llamada trayectoria del sistema. Por otra parte existen puntos donde la solución es constante y a estos se le conoce como puntos críticos o de equilibrio del sistema.

Resulta de interés para los investigadores, el estudio de la disposición de las trayectorias alrededor de los puntos críticos, para el reconocimiento de posibles estados estacionarios. De manera que si se realiza una pequeña perturbación en el sistema y en este se produce un alejamiento del punto crítico se estará en presencia de un equilibrio inestable o, por el contrario, si el sistema se acerca al punto crítico estamos en presencia de un estado de equilibrio estable.

Identificar aquellos parámetros o valores de los mismos que llevan al sistema a comportamientos estables o inestables es imprescindible para arribar a conclusiones de valor para los investigadores, puesto que la estabilidad representa la habilidad para retornar al estado de equilibrio después de los cambios o perturbaciones temporales, según factores externos e internos, en muchas ocasiones impredecibles. Los conceptos referidos a la Teoría de la

Estabilidad fueron planteadas inicialmente por el matemático Aleksandr Mijáilovich Lyapunov.

Se ha podido identificar que la sensibilidad de un sistema ante pequeños cambios en los valores de los parámetros y/o de los estados iniciales, se expresa en la variación que sufren las variables de estado en el tiempo.

Existen técnicas para el análisis de las series temporales con el fin de reconocer patrones de comportamientos según la evolución de las variables y parámetros dentro de un sistema y predecir posibles valores futuros.

Muchas son las aplicaciones prácticas donde resulta interesante el estudio de la estabilidad de las soluciones, para ello se han desarrollado algunos métodos que permiten cuantificar la estabilidad del sistema dinámico. En este caso están: el Exponente de Lyapunov, la Dimensión de Correlación y el Ploteo Recurrente [2], [3].

Uno de los métodos más usados es la cuantificación de la estabilidad mediante el exponente de Lyapunov, el cual define que al aparecer un exponente positivo las órbitas cercanas divergen exponencialmente en una determinada dirección. Se puede garantizar así que el sistema es inestable, lo cual se relaciona con la imposibilidad o extrema dificultad para poder predecir el comportamiento futuro del sistema a partir del conocimiento del estado actual del mismo, lo que constituye una señal de un comportamiento caótico. De no existir un exponente positivo se estará en presencia de un caso estable.

En sistemas dinámicos de pocas dimensiones las combinaciones de signos permiten hacer una clasificación del comportamiento, pero a partir de más de tres dimensiones resulta difícil poder especificar un comportamiento [4]. Esto está dado precisamente, porque en dimensiones mayores son más variables que estarán definiendo uno u otro comportamiento y por otro lado por lo general son sistemas que presentan más de dos parámetros de control y resultan ser además, altamente no lineales. La teoría existente sólo permite la interpretación de resultados cuando se trabaja con sistemas de pocas dimensiones, por lo que aunque no se pueda caracterizar el sistema según los comportamientos distintivos planteados anteriormente, sí se necesita poder llegar a establecer si los sistemas de alta dimensionalidad son estables o inestables, pues por lo general son los que encontramos en la naturaleza.

En la actualidad se han desarrollado diferentes software tales como: Copasi, TISEAN y Matlab que implementan los métodos de análisis de series temporales y específicamente los de análisis de estabilidad. En todos los casos la limitante principal es que hacen el análisis de una serie temporal en particular y dejan el peso de las conclusiones del estudio en manos de los investigadores. Esto último resulta muy complicado de hacer de forma manual, pese a ello no se han encontrado referencias en la literatura revisada sobre alguna herramienta computacional que automatice el proceso.

Debido a que los software antes mencionados, realizan el análisis de la estabilidad de una sola serie temporal a la vez, y que de hacerlo para varias, aumentaría el tiempo de obtención del resultado, dada la complejidad embebida en el algoritmo, sería factible hacer el estudio a más de una serie y disminuir el tiempo de cómputo que conllevaría el mismo.

Dada la situación problemática se plantea como **problema científico**:

¿Cómo realizar el estudio de la estabilidad e interpretación de los resultados en Sistemas de Ecuaciones Diferenciales n-dimensionales al variar los parámetros?

Para dar respuesta a este problema se plantea la siguiente **hipótesis**: Con el desarrollo de una metodología que integre el análisis de estabilidad de las series temporales, el procesamiento de los resultados y la interpretación de los mismos, se podrá obtener un estudio de estabilidad de los Sistemas de Ecuaciones Diferenciales n-dimensionales al variar algunos de sus parámetros.

Objeto de estudio: Análisis de las series temporales.

Campo de acción: Análisis de la estabilidad de las series temporales.

Objetivo general: Desarrollar una metodología que permita estudiar la estabilidad de los Sistemas de Ecuaciones Diferenciales n-dimensionales, respecto a la variación de los parámetros del mismo e interpretar los resultados obtenidos.

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

1. Definir algoritmos para realizar el estudio de estabilidad en un conjunto de series temporales a partir de la variación de los parámetros.
2. Definir algoritmos para interpretar los resultados de los estudios de estabilidad para múltiples series temporales utilizando Minería de Datos.

3. Proponer una metodología para el estudio de estabilidad de SED n-dimensionales basada en los algoritmos implementados en los objetivos uno y dos.
4. Incorporar los algoritmos propuestos al software BioSyS en su versión 1.0.
5. Verificar los resultados obtenidos con los reportados en la literatura sobre el análisis de la estabilidad.

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

1. Realización de simulaciones distribuidas.
2. Estudio del análisis de estabilidad a partir del exponente de Lyapunov.
3. Análisis de estabilidad de varias series temporales a partir del espectro de los exponentes de Lyapunov.
4. Selección de una técnica de Minería de Datos para el estudio de los resultados del análisis de la estabilidad.
5. Implementación de la funcionalidad que permita representar o graficar el resultado del análisis de la Minería de Datos.
6. Aplicación de la metodología al modelo de Lorenz.

Estructura del documento:

El presente documento está estructurado en 3 capítulos. El **Capítulo1:**Revisión Bibliográfica, brinda una amplia reseña del estado del arte a nivel mundial, abarcando temas como los sistemas dinámicos, la modelación matemáticas, la repercusión de las variaciones en las condiciones iniciales y los parámetros dentro de los modelos. Por otra parte se trata el análisis de las series temporales, fundamentalmente las técnicas y software que permiten este tipo de estudio y se hace énfasis en una de las más utilizadas que es el exponente de Lyapunov. También se precisan sobre los temas del análisis distribuido y la Minería de Datos. El **Capítulo2:** Materiales y Métodos se describen las herramientas computacionales y los procedimientos utilizados en este trabajo. El **Capítulo3:** Resultados y Discusión se presentan de manera concreta los principales resultados obtenidos, se discuten y se comparan con otras investigaciones. Finalmente se procede a las conclusiones y recomendaciones que se sugiere la autora. Al final se incluyen las **Referencias Bibliográficas** y un conjunto de **Anexos** que brindan información complementaria al lector.

Capítulo 1: Revisión Bibliográfica

En este capítulo se abordan los principales conceptos referentes a los sistemas dinámicos, así como los métodos que se utilizan para el análisis de estabilidad que fueron consultados en la literatura. También se analizaron algunas herramientas informáticas diseñadas para este fin.

1.1. Los Sistemas Dinámicos

Los sistemas dinámicos constituyen un área de investigación de las matemáticas relativamente joven, pues hace alrededor de cuarenta años ha tomado mayor relevancia a partir de los trabajos realizados por los matemáticos e ingenieros S. Smale, V. Arnold y Lyapunov, pero sus inicios se remontan a los trabajos realizados por parte de Newton y Henri Poincaré con sus estudios en Mecánica Celeste y las ecuaciones diferenciales respectivamente.

Eduard Groller plantea que un sistema dinámico es aquel cuya evolución temporal de un estado inicial es establecida por un juego de reglas. Se encuentran sistemas dinámicos en muchas áreas de investigación y aplicación. Ejemplos son los procesos económicos, la Físicas, la Medicina y modelos de crecimiento de población [5].

Formalmente Rosales define que un sistema dinámico es un modo de describir el recorrido a lo largo del tiempo de todos los puntos de un espacio S dado. El espacio S puede imaginarse, por ejemplo, como el espacio de estados de cierto sistema físico. Matemáticamente, S puede ser un espacio euclidiano o un subconjunto abierto de un espacio euclidean. Un sistema dinámico para S nos dice para cada $x \in S$, donde estará x una unidad de tiempo más tarde, dos unidades de tiempo más tarde, y así sucesivamente [6].

De manera general un sistema dinámico es el sistema que experimenta variaciones en el tiempo y puede ser definido matemáticamente, como una herramienta para describir la evolución de su estado a través del tiempo. Modelar a través de la matemática a un sistema dinámico persigue dos objetivos principales: comprender un fenómeno y realizar alguna predicción sobre su comportamiento futuro. No son solamente los modelos matemáticos los que permiten el estudio de los sistemas dinámicos, también son analizados

a partir de las mediciones experimentales que se registran en un tiempo específico o de interés sobre un fenómeno real.

1.2. Modelación matemática

Como se decía anteriormente uno de los enfoques desarrollados para el estudio de los sistemas dinámicos es la modelación matemática que es una herramienta que permite describir una dinámica a través de las ecuaciones que rigen la misma y su objetivo es explicar o comprender los procesos naturales.

“Un modelo matemático es una representación para un objeto o proceso, en el cual quedan plasmadas sus principales características, a partir de conjeturas o suposiciones iniciales, estableciendo un compromiso entre complejidad y exactitud; en esta idealización es posible utilizar entes, estructuras y leyes de la matemática; permitiendo interpretar los resultados del modelo en términos del objeto o proceso estudiado [7]”.

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

Para la modelación se puede hacer uso de distintos tipos de modelos como: Autómatas Celulares, Sistemas de Ecuaciones Diferenciales, Sistemas de Ecuaciones Diferenciales en Derivadas Parciales y Ecuaciones en Diferencias o mapas. La forma más común de representar un sistema dinámico es mediante un sistema de ecuaciones diferenciales, de la forma:

$$\frac{d}{dt} = f(x, t, \mu)$$

El conjunto de variables del sistema está representado por x que es un punto de un conjunto abierto en R^n . El tiempo se representa como $t \in R$, aunque se puede considerar como la variable encargada de parametrizar las trayectorias o soluciones, descritas por el vector x . El vector μ representa una colección de p parámetros, $\mu \in R^p$ que son fijados en cada evolución del sistema [8].

1.2.1. Condiciones iniciales y parámetros

Las soluciones a un sistema de ecuaciones diferenciales se obtienen a partir de la definición de un conjunto de condiciones iniciales y un conjunto de parámetros, o sea, resolver el conocido problema de Cauchy. Cuando se está en presencia de un sistema autónomo (no dependiente del tiempo) las soluciones obtenidas no se cortan entre sí, es decir, un sistema determinista y autónomo debía ser predecible.

Sin embargo, existe una teoría sobre la multiplicidad de las soluciones de ecuaciones de evolución no lineal, las cuales fueron introducidas dentro de la rama de la matemática relacionada con la teoría de las bifurcaciones. Esta última ha permitido comprender cómo el número y la estabilidad de dichas soluciones cambian cuando se realiza una perturbación infinitesimal de algún parámetro. Los trabajos de Yorke y May en la década del 70 conllevaron a la idea de que los sistemas determinísticos aunque simples podían generar trayectorias de dinámicas muy influenciadas por la sensibilidad a las condiciones iniciales, lo cual indica que a pesar de conocer las reglas que definen la evolución del sistema si este es caótico no se podrá realizar una predicción a largo plazo por la divergencia exponencial que la dinámica experimenta ante pequeñas perturbaciones [9].

Estudios sobre los sistemas dinámicos no lineales han propiciado llegar a la conclusión de su sensibilidad frente a pequeños cambios en las condiciones iniciales. Matemáticamente, todos los sistemas no lineales con más de dos grados de libertad pueden llegar a mostrar caos y por lo tanto ser impredecibles. De igual manera en sistemas con esta característica también pueden estar incluidos varios parámetros, con los cuales generalmente se realizan variaciones para comprobar si se generan cambios de estado, dígame por ejemplo, dónde pueden aparecer o desaparecer estados de equilibrio [10], [11].

Tanto las variaciones en los valores de las condiciones iniciales como de los parámetros pueden llevar a comportamientos divergentes del sistema por lo que para comprender a fondo determinado modelo se hace necesario realizar exploraciones intensivas de los mismos, acompañadas preferiblemente de estudios de estabilidad o algún otro tipo de análisis que arroje información importante y concluyente sobre el mismo.

Resumiendo se puede decir que se necesitan tres elementos para reconocer el comportamiento de un sistema, ellos son: las ecuaciones, los parámetros y las condiciones iniciales [8].

1.3. Series Temporales

En todas las ramas de la ciencia existen datos que pueden ser considerados como series de tiempo, tales como: en la Economía con las series de precios de un artículo, tasas de desempleo, tasa de inflación, índice de precios; en la Física con las series de Meteorología, cantidad de agua caída y temperatura máxima diaria; o en la Telecomunicación con las series de Análisis de señales por sólo citar algunas.

Estos fenómenos en su mayoría suelen describirse a partir de sistemas de ecuaciones diferenciales que muy pocas veces se pueden resolver analíticamente por lo que se utilizan los métodos numéricos [12].

Si de estos fenómenos se cuenta con las ecuaciones, los parámetros y las condiciones iniciales, al resolver el sistema se obtiene un conjunto de simulaciones que al estar medidas durante un intervalo de tiempo, constituyen lo que se conoce como series temporales.

Según se plantea “una serie temporal es una secuencia cronológica ordenada de valores de medición sobre el estado de una variable cuantitativa de un proceso, obtenida a partir de la observación de su comportamiento en el transcurso del tiempo [13]”.

Igualmente se afirma que una serie temporal es un conjunto de mediciones de cierto fenómeno o experimento registradas secuencialmente en el tiempo. Estas observaciones serán denotadas por:

$\{x(t_1), x(t_2), \dots, x(n)\} = \{x(t): t \in T \subseteq R \text{ con } x(t_i)\}$ el valor de la variable x en el instante t_i . Si $T = Z$ se dice que la serie de tiempo es discreta y si $T = R$ se dice que la serie de tiempo es continua [14].

El análisis más sencillo de una serie temporal es representarla en un gráfico donde se muestre la evolución de la variable a lo largo del tiempo, como se ve en la **Fig. 1.1**.

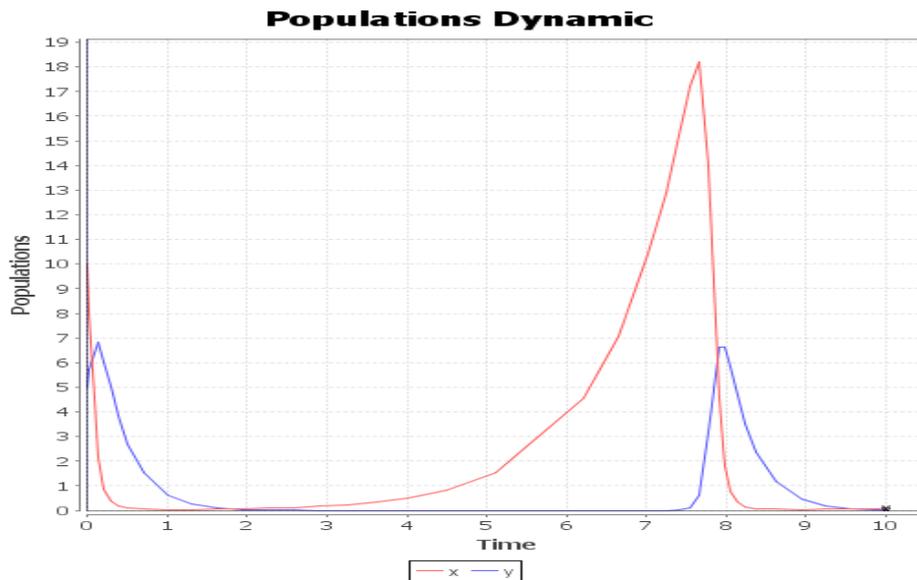


Figura 1.1 Serie temporal, medición de las variables x y y en el tiempo.

El estudio de las series temporales persigue los siguientes objetivos: conocer adecuadamente la evolución de las variables y parámetros, predecir posibles valores futuros y reconocer patrones de comportamientos. Para ello existen algunos tipos de análisis que ayudan en su investigación [15].

1.4. Análisis de series temporales

El análisis de series temporales incluye métodos que permiten interpretar este tipo de datos, extrayendo información representativa referente a los orígenes o relaciones y dando la posibilidad de pronosticar comportamiento futuro, este último constituye uno de los usos más habituales de las series temporales.

Muchos sistemas dinámicos no tienen un comportamiento regular en el espacio, como se dijo anteriormente, a estos se les conoce como sistemas caóticos pues evolucionan en el tiempo de manera inesperada. Cualquier pequeña perturbación puede cambiar el sistema entero, al análisis para poder identificar esta característica de los sistemas dinámicos se le conoce como Análisis de Estabilidad.

1.4.1 Análisis de Estabilidad

La teoría de la estabilidad juega un papel importante en la teoría de sistemas e ingeniería, estudia el comportamiento estable de las soluciones de las ecuaciones diferenciales en sistemas dinámicos, es decir, se verifica cómo difieren las soluciones bajo pequeñas modificaciones de las condiciones

iniciales. Como se dijo anteriormente para muchas disciplinas es necesario conocer la estabilidad de algunos sistemas pues comúnmente en la práctica las condiciones iniciales nunca se conocen con toda precisión y se hace necesaria de antemano identificar esta cualidad para evitar que pequeñas perturbaciones no provoquen comportamientos cualitativos diferentes.

Según definió el físico Lyapunov en 1892, la estabilidad de un sistema está relacionada con el comportamiento de sus trayectorias cuando su estado inicial se encuentra cerca de un equilibrio, siendo notable la idea de que las perturbaciones que afectan a un sistema tienden a separarlo del equilibrio.

Existen algunos tipos de análisis de series temporales específicamente para reconocer la estabilidad, ellos son: Dimensión de Correlación, Gráficos de Recurrencia o Ploteo Recurrente y Exponente de Lyapunov.

1.4.1.1. Dimensión de Correlación

La dimensión de correlación se refiere al número mínimo de dimensiones necesario para caracterizar al sistema. La baja dimensionalidad indica la existencia de un modelo simple que está generando y rigiendo el sistema en estudio.

Para el cálculo de la dimensión de correlación se han definido algunas alternativas, pero el método propuesto por Grassberger y Procaccia en 1983 es el más utilizado puesto que no precisa de series temporales tan largas como el método originariamente propuesto por Takens en 1981. Para su cálculo se elige al azar un punto perteneciente a la dinámica y sobre él se marca una esfera de radio r , luego se cuenta la cantidad de puntos dentro de la esfera y se estudia cómo varía esa cantidad en función del radio r de la esfera. Por ejemplo para una distribución uniforme de puntos en un volumen (objeto de tres dimensiones), el número de puntos varía como r^3 . Los exponentes de r son los valores de las dimensiones necesarias para describir la dinámica. Para dinámicas caóticas se encuentran formulaciones del tipo r^n , donde n es un número no entero [3].

1.4.1.2. Ploteo Recurrente

Los gráficos de recurrencia representan la dinámica de una serie temporal en un espacio bidimensional. En este tipo de gráficos, las regularidades o irregularidades que se observen, serán muestra de esta misma

característica en la dinámica del sistema. Este procedimiento se basa en buscar órbitas regulares (recurrencias), estables o inestables, de una forma gráfica. Además de ser una técnica muy utilizada para reconocer dinámicas deterministas no lineales o caóticas, tiene la ventaja de que pueden ser estudiadas en series cortas pues lo único que se busca es la presencia de alguna de estas recurrencias indicativas de un orden complejo [3].

Matemáticamente el ploteo recurrente puede ser representado de la siguiente manera:

$$R_{i,j} = \theta(\varepsilon_i - \|x_i - x_j\|), x_i \in R^m, \quad i, j = 1 \dots N$$

Donde N es el número de estados considerados x_j , ε_i es la distancia umbral y $\theta(\cdot)$ es la función de Heaviside (Marwan). A continuación se muestra en la **Fig. 1.2** la gráfica de recurrencia del modelo de Lorenz:

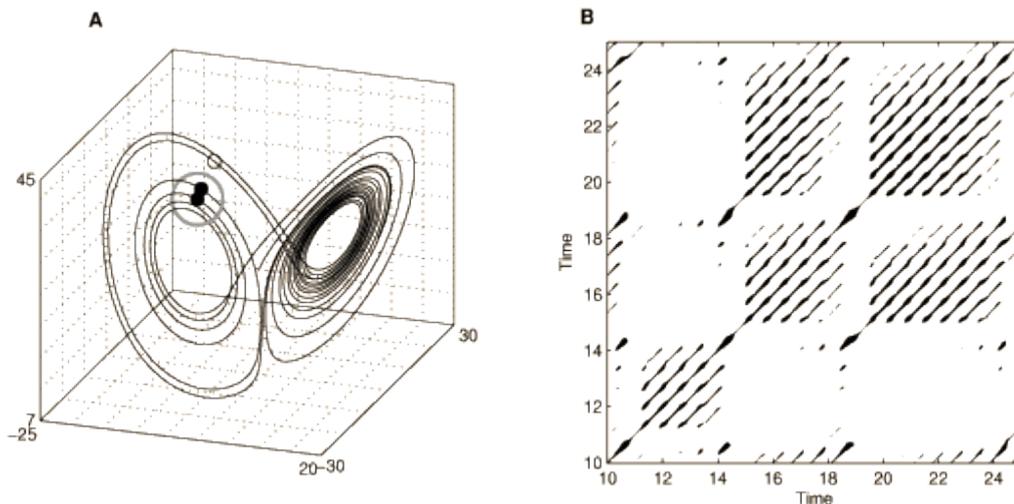


Figura 1.2 (A) Trayectoria del espacio de fase del modelo de Lorenz.
(B) Ploteo Recurrente.

1.4.1.3. Exponente de Lyapunov

El exponente de Lyapunov es una medida cuantitativa de la sensibilidad del sistema a las condiciones iniciales. Este método caracteriza la separación de trayectorias próximas y determina la condición de comportamiento caótico del sistema cuando toma valores positivos [16].

Existen dos alternativas para el reconocimiento de la estabilidad en un sistema a través del método del exponente de Lyapunov. El primero llamado indirecto, trabaja con las soluciones explícitas de las ecuaciones diferenciales que describen el sistema, mientras que el segundo, llamado directo, no

necesita de dichas soluciones pues utiliza el modelo matemático del sistema, por lo que resulta ser más general, además se encuentra mucho más extendido al ser más exacto, a diferencia al método a partir de las soluciones porque en ellas pueden haberse incluido errores, o ruido [17], [18]. En el artículo de Alan Wolf *et al.* [4] se considera que el exponente de Lyapunov siguiendo cualquiera de estas dos vías ha demostrado ser el método de diagnóstico más útil para el reconocimiento de los sistemas caóticos.

Los exponentes de Lyapunov proporcionan una caracterización cualitativa y cuantitativa de la convergencia exponencial de órbitas cercanas en el espacio de fase, teniendo en cuenta que órbitas cercanas pertenecerán a estados parecidos. Cualquier sistema con al menos un exponente positivo es suficiente para considerarlo caótico, de lo contrario tiene un comportamiento estable. En un sistema si no existen un punto fijo, al menos un exponente tendrá valor cero [4].

Dado un sistema dinámico continuo en un espacio de fase n -dimensional, se puede ver la evolución a largo plazo en una esfera infinitesimal formada por las condiciones iniciales. La esfera se volverá un elipsoide debido a la deformación que va ocurriendo ver **Fig. 1.3**. El exponente de Lyapunov en una dimensión se define entonces en términos de la longitud del eje principal del elipsoide. El exponente de Lyapunov (λ) está relacionado con la contracción o expansión de las diferentes direcciones que el sistema pueda tomar en el espacio de fase, el elipsoide cambia continuamente cuando evoluciona en direcciones asociadas con un exponente dado [4].

Este concepto es generalizado con el espectro de los exponentes de Lyapunov, λ_i ($i=1, 2, \dots, n$), siendo n el número de variables de estado del sistema [2].

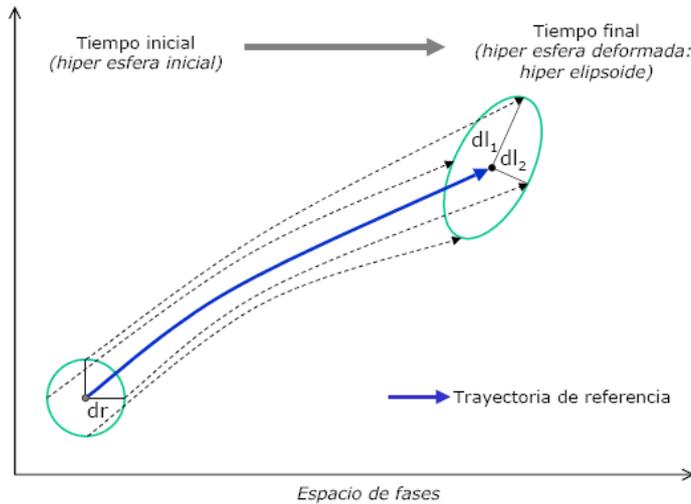


Figura 1.3 Representación en el espacio de fases de la evolución de un sistema a partir de condiciones iniciales ligeramente distintas.

Para el exponente de Lyapunov se define lo siguiente: en el caso más general de funciones $A(t)$, el exponente de Lyapunov asociado a la solución de $x(t; x_0)$ estará dado por la ecuación

$$\dot{x} = A(t)x, x(0) = x_0 \in R^d, t \in R$$

$$\text{siendo } \lambda(x_0) := \lim_{t \rightarrow \infty} \frac{1}{t} \log \| X(t; x_0) \|$$

En su trabajo Lyapunov demostró que para toda solución $x(t; x_0)$ con $x_0 \neq 0$ se cumple que $\lambda(x_0) < \infty$ [1].

Los signos de los exponentes de Lyapunov brindan una imagen cualitativa de la dinámica. “Si la perturbación inicial, en promedio, tiende a disminuir con el tiempo, entonces $\lambda(x_0)$ será negativo. En cambio, si la solución depende sensiblemente de las condiciones iniciales la perturbación crecerá sin límite y resultará en un $\lambda(x_0)$ positivo [19]”.

Por ejemplo en los sistemas dinámicos de hasta tres dimensiones, según las combinaciones de signos que se obtienen del exponente de Lyapunov, las clasificaciones más significativas son: atractor extraño (+, 0, -), dos toroides (0, 0, -), ciclo límite (0, -, -) y estado de equilibrio (-, -, -)[4].

1.4.2 Selección del método para el análisis de Estabilidad

Del análisis realizado a los tres métodos anteriores [20], [21] se puede decir que:

1. la Dimensión de Correlación y el Exponente de Lyapunov requieren para su correcta aplicación e interpretación que las series sean estacionarias.

2. muchas series experimentales no son estacionarias, es por ello que para su estudio se desarrolló la técnica conocida como Ploteo Recurrente propuesto por Eckmann, Kamphorst y Ruelle en 1987.

3. en los métodos Ploteo Recurrente, el Exponente de Lyapunov a partir de series temporales y la Dimensión de Correlación, para poder obtener las dinámicas que representen comportamientos como pudiera ser un atractor, un ciclo límite u otro, es necesario que la serie temporal cuente con una cantidad específica o que por lo general contenga un número grande de datos o puntos. Por ejemplo el atractor de Hénon es un sistema relativamente simple, pero se requieren 500 datos para poder reconstruirlo.

4. los métodos Ploteo Recurrente, el Exponente de Lyapunov a partir de series temporales y la Dimensión de Correlación, necesitan que el proceso de integración numérica sea lo más exacto posible dado que de por sí, al trabajar con series temporales existe el riesgo de introducir errores, ruido.

En la **Fig. 1.4** se muestra en la rama de la izquierda, este tipo de análisis.

Resumiendo, para resolver las limitantes de que unas técnicas solo analicen series estacionarias y que todas requieran de series largas de acuerdo al comportamiento que caracterizan, y que por otra parte se requiera de la mayor exactitud en el proceso de integración, es entonces la vía más adecuada, el Exponente de Lyapunov a través de la variante que realiza el estudio con el modelo matemático, pues aunque resulta mucho más costoso computacionalmente por la complejidad que tienen los algoritmos, es considerablemente más exacto, por lo que no se expone el estudio a la entrada de errores.

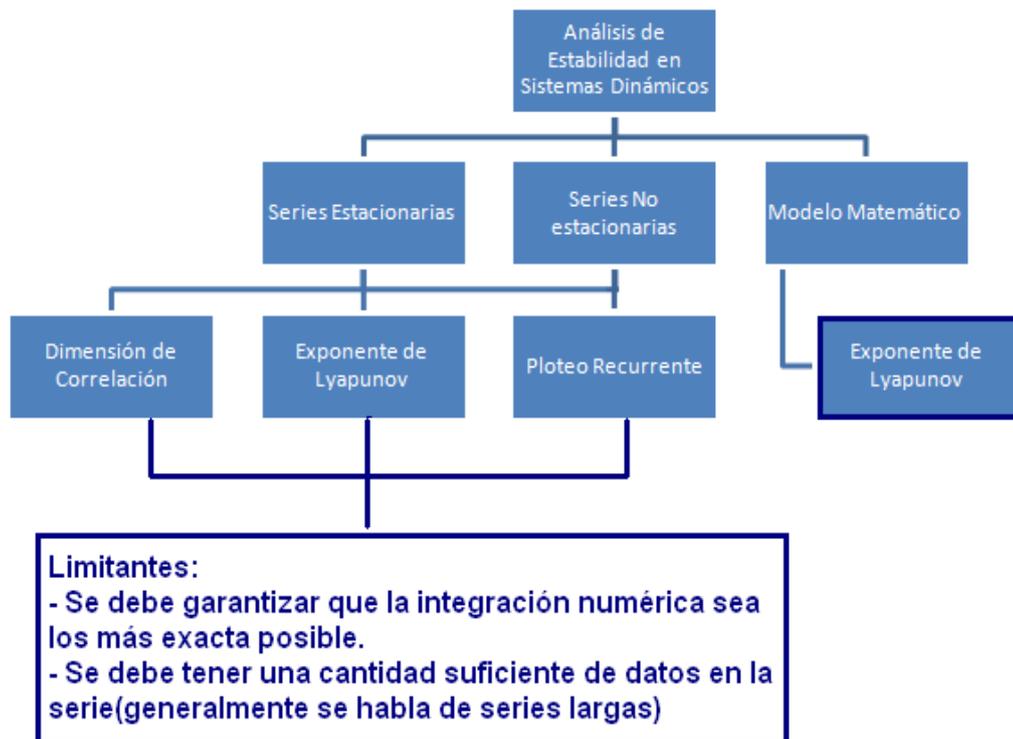


Figura 1.4 Análisis de las limitantes en el estudio de estabilidad en series temporales.

1.5. Software para el análisis de series temporales

En la búsqueda de software que implementaran análisis el análisis de estabilidad haciendo uso de los exponentes de Lyapunov, se encontraron algunos de los cuales a continuación se hace una descripción.

1.5.1. Copasi

El software Copasi es una aplicación multiplataforma, que realiza múltiples análisis referentes al estudio de la Biología de Sistemas. Emplea métodos para garantizar soluciones con altos niveles de fiabilidad, entre sus funciones están: editar modelos desde la perspectiva matemática o bioquímica, hallar estados estacionarios, estimación y optimación de parámetros y realiza varios tipos de análisis como el del exponente de Lyapunov en una dinámica del sistema. Para el cálculo esencialmente utiliza un conjunto de librerías estándares las cuales implementan varios métodos numéricos. Se debe destacar que los procesos que se llevan a cabo para responder a los tipos de análisis son costosos computacionalmente por el gran número de operaciones que realizan internamente [22].

1.5.2.TISEAN

El software TISEAN es una herramienta de software libre específicamente para el estudio de series temporales que ofrece comandos y un conjunto de opciones para el trabajo con los diferentes tipos de análisis que tiene implementado. La interacción con el software a la hora de obtener los resultados no es de fácil entendimiento para investigadores poco acostumbrados al trabajo con comandos, pues la entrada al software es a través de ficheros. De igual forma se devuelven los resultados de los diferentes análisis. El TISEAN utiliza diversos métodos para realizar los diferentes análisis, por ejemplo para problemas de estabilidad se utilizan los métodos siguientes: Exponentes de Lyapunov y los métodos de dimensión de correlación y de entropías. Dentro del análisis respecto al exponente de Lyapunov hay implementadas dos variantes de análisis, una a partir del máximo exponente, contando para ello con dos posibles algoritmos y otra, obteniendo todo el espectro de los exponentes [23].

1.5.3. Matlab

Es un asistente matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio. Está disponible para las plataformas Unix, Windows y Apple Mac OS X. Está orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. Matlab integra análisis numérico, cálculo matricial, procesamiento de señales y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían tradicionalmente, sin necesidad de hacer uso de la programación habitual [24].

Matlab dispone de dos herramientas que expanden sus prestaciones, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario-GUI). Por otra parte tiene un amplio abanico de programas de apoyo especializados, Toolboxes, como se le denominan, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad, prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación destacando entre ellos el 'toolbox' de procesos de imágenes, señal, control y redes

neuronales. Este software muy usado en universidades y centros de investigación y desarrollo incluye funcionalidades para el análisis de estabilidad utilizando el cálculo del exponente de Lyapunov, específicamente valiéndose del espectro completo de los exponentes [25].

Conclusión Parcial:

1. Análisis de una sola serie temporal: los software estudiados implementan algoritmos para el análisis de series temporales y específicamente para el análisis de la estabilidad, pero se pudo apreciar que los estudios lo realizan a partir de una sola serie temporal a la vez y en algunos algoritmos se precisa de series largas para poder realizar el estudio.

2. Sistemas n-dimensionales: los procesos que son estudiados o de interés para entender los fenómenos que a diario ocurren en la naturaleza, en su mayoría están compuestos por más de tres ecuaciones diferenciales y por ello sus trayectorias no pueden ser clasificadas de acuerdo al tipo de comportamiento, por lo que resulta necesario, para contrarrestar esta dificultad, poder establecer si el sistema es estable o no.

3. Tiempos largos de análisis: debido a la complejidad matemática de estos métodos, ya sea en las tareas de integración numérica o en el trabajo con matrices, en general hacen que la obtención del análisis a partir de una sola serie temporal sea costosa computacionalmente.

Teniendo en cuenta que el cálculo de los exponentes de Lyapunov es costoso computacionalmente y que si se quiere hacer un estudio completo de la estabilidad de los sistemas respecto a la variación de variables y parámetros hay que repetir el cálculo de dicho exponente varias veces, se requiere del uso de alguna infraestructura computacional que acelere este proceso y ponga en manos de los investigadores los resultados de los estudios en el menor tiempo posible.

En los siguientes epígrafes se hará un estudio de diferentes infraestructuras que pueden ayudar en la solución de este problema y se escogerá la adecuada para este caso.

1.6. Sistema de cómputo distribuido

Un sistema distribuido es un conjunto de computadoras autónomas que aparecen integradas ante los usuarios como una única máquina, para resolver determinado problema. Los componentes del sistema no son más que hardware y software que se comunican entre sí a través de una red, preferiblemente canales de alta velocidad [26], [27], [28]. En el caso de un sistema de cómputo distribuido, el problema a resolver por lo general requiere grandes cálculos y lo que se trata es de reducir el tiempo en obtener la respuesta haciendo uso del procesamiento en paralelo, al distribuir los cálculos de forma transparente para el usuario entre varias computadoras.

Entre los sistemas de cómputo distribuido más reconocidos, se destacan algunos que han sido concebidos para dar solución a problemas específicos en distintas ramas de la ciencia, pero que no pueden ser reutilizados en otros problemas por su especificidad. Sin embargo, existen otros que han sido creados para un propósito general [28], [29], [30], [31] y sí pueden ser usados para dar solución a cualquier problema de intenso cómputo, que pueda ser dividido en subtareas más pequeñas para ser repartidas entre diversos procesadores, estos últimos son los que resultan interesantes para el presente trabajo.

1.6.1. Computación voluntaria

En la actualidad hay muchas PCs que en materia de cómputo están ociosas porque en su mayoría realizan tareas poco complejas, no utilizan todos los recursos del procesador, más aún teniendo en cuenta las excelentes características con las que cuentan las computadoras personales actuales, hay muchos recursos desaprovechados que podrían ayudar en un proyecto científico si fueran donando para su utilización.

Los proyectos que hoy se benefician de la computación voluntaria son de tipo académico o científico, ejemplos son: SETI@Home y ClimatePrediction.NET. El usuario que brinda su PC participa en los proyectos de forma anónima y gracias a su participación proyectos de bajo presupuesto pueden seguir adelante [31].

1.6.2. Plataformas existentes

Se mencionan a continuación algunos software que bajo la filosofía de computación voluntaria utilizan el tiempo ocioso de las computadoras disponibles que habitualmente funcionan como estaciones de trabajo:

1.6.2.1. Berkeley Open Infrastructure for Network Computing (BOINC)

BOINC es una infraestructura para la computación distribuida desarrollada originalmente para el proyecto SETI@home [28], [32], pero con vista a ser utilizada en otros proyectos. BOINC consiste en un cliente que es compartido entre todos los proyectos BOINC. Este sistema se caracteriza por utilizar una cola central de trabajos y de tener clientes conectados al planificador (scheduler).

Un usuario que desee donar su computadora para un proyecto BOINC, descarga la aplicación desde el sitio Web del proyecto y posteriormente instala el mismo. El usuario tiene un gran control sobre el cliente según sus preferencias, puede especificar cuándo el cliente puede ejecutar los trabajos, ya sea mientras el usuario está haciendo uso de la computadora o sólo usarla después de un período determinado de tiempo sin actividad. Como preferencia específica de un proyecto, el usuario puede establecer el mínimo o el máximo número de trabajos que el cliente puede tener en la computadora donada, para un proyecto dado. La cola local de trabajos en el cliente es atendida con un comportamiento FIFO (firts-in, firts-out). Los clientes realizan varios trabajos a la vez, según la cantidad de CPUs en el sistema. Los clientes se comunican con el servidor a través de http.

El servidor BOINC consiste en un sistema de base de datos para registrar el estado de las unidades de trabajo y sus resultados, un servidor Web de datos al servicio de los clientes y que permite acceder al planificador y cinco programas demonios ejecutándose periódicamente. La Base de Datos en MySQL almacena información importante del servidor BOINC. Esto incluye información acerca de los usuarios registrados y sus computadoras correspondientes, sobre las aplicaciones y sus versiones, sobre los clientes y sus versiones, y por supuesto sobre las unidades de trabajo y sus resultados asociados.

1.6.2.2. Condor

Condor es un sistema especializado para administrar trabajos de cómputo intenso, y realizar balance de carga. Condor es el resultado del proyecto de investigación Condor de la Universidad de Wisconsin-Madison. El proyecto se inició en 1988 y se basa en los resultados del proyecto Remote-Unix de la misma universidad [29].

El sistema Condor, consiste en un administrador central de recursos (central manager), una o más máquinas de envíos (submit machine), y una o más máquinas de ejecución (execution machine). El administrador central de recursos, maneja los detalles de vincular los trabajos con los recursos, y esto implica el conocimiento de ambos.

Condor es muy flexible, soporta muchas características y diferentes plataformas. Un concepto central en Condor es el denominado “*universos*” (universes), estos son los entornos en tiempo de ejecución que ofrecen distintas características para el usuario Condor, soportando los siguientes universos: standard, vanilla, PVM, MPI, globus, java y scheduler.

Condor también incluye soporte heterogéneo, lo cual permite que los trabajos puedan ser subidos y ejecutados en diferentes plataformas de computadoras, si los ejecutables para cada una de estas plataformas están disponibles. Si el trabajo no está vinculado a una determinada plataforma antes de su ejecución, significa que habrá más máquinas para elegir cuando se trata de buscar una máquina de ejecución. Por último, Condor hace seguro que el traslado de trabajos sea entre máquinas de la misma plataforma.

1.6.2.3. Plataforma de Tareas Distribuidas (T-arenal)

La Plataforma de Tareas Distribuidas es un sistema de cómputo distribuido programado en Java [33], basado en el software Java Based Heterogeneous Distributed Computing System [34]. Ha sido utilizado para dar solución a varios problemas de la Bioinformática y brinda un modelo de programación de alto nivel basado en el paradigma de la Programación Orientada a Objeto utilizando RMI para el paso de mensaje.

El sistema T-arenal, consiste en un servidor central (T-arenal server) y uno o más clientes (T-arenal client). El servidor central maneja toda la información concerniente al sistema, realiza la transferencia de ficheros vía

sockets TCP y planifica el orden en que los trabajos serán atendidos. El usuario que dona su computadora a T-arenal, sólo debe instalar la aplicación cliente y la estará donando al sistema, no a un proyecto en específico, por lo tanto, un cliente puede trabajar en uno o varios proyectos.

La aplicación cliente está formada además por dos hilos (threads) que se ejecutan periódicamente: el ClientController y el AlgorithmMonitorThread. El servidor T-arenal está formado por un sistema de base de datos para registrar información concerniente al sistema y un programa demonio (hilo de proceso que se ejecuta en segundo plano) ejecutándose periódicamente que es T-arenalServer.

Los sistemas distribuidos estudiados anteriormente implementan el modelo cliente–servidor, donde el sistema está compuesto por un servidor y varios clientes. De ellos T-arenal tiene la característica de poder ser ejecutado independientemente de la plataforma sin necesidad de recompilar el código fuente. A este tipo de aplicación se le conoce en la literatura como “write once and run anywhere”.

La aplicación de estas herramientas para distribuir las tareas de cómputo, ha posibilitado tener un cúmulo de información de la cual es necesario sacar conclusiones dentro de cualquier estudio, para esto se necesita de herramientas informáticas que faciliten la interpretación de dicha información y no lo deje solamente en manos de los investigadores.

1.7. Minería de Datos

Con el avance de la ciencia son muchos los datos que se recogen procedentes de diversos procesos. En buena medida es interesante tener inmensa cantidad de datos, pero por sí solos o solamente contando con ellos no ayuda a científicos o a cualquier otro implicado, a sacar la información necesaria para tomar determinada decisión o llegar a conclusiones.

Existe un área de descubrimiento del conocimiento conocido como Descubrimiento de Conocimiento en Bases de Datos (KDD, Knowledge Discovery in Databases) el cual se puede definir como “el proceso no trivial de identificar patrones válidos, novedosos y potencialmente útiles y en última instancia comprensible a partir de los datos [35]”. Dentro del mismo se encuentran algunas etapas con el objetivo de extraer información entre los

cuales se encuentran: selección, limpieza, integración, transformación, reducción, Minería de Datos, evaluación de patrones e interpretación de resultados [36].

Específicamente la Minería de Datos se puede definir como una etapa particular en el proceso de KDD, donde la Minería de Datos aplica algoritmos específicos o técnicas específicas para la extracción de patrones de los datos [35], diferenciándolo del proceso KDD antes mencionado. La Minería de Datos integra los procesos de aprendizaje y métodos estadísticos para la obtención de hipótesis de patrones y modelos. La Minería de Datos es capaz de procesar diversos datos, procedentes de diferentes fuentes como los distintos tipos de base de datos, datawarehouse y archivos [36].

1.8. Minería de datos en el estudio de sistemas dinámicos

El proceso de estudio de los sistemas dinámicos arroja grandes volúmenes de datos, sea desde un estudio experimental o más aún si se obtienen a partir de las simulaciones de un sistema. Es por esta razón que se necesitan de herramientas computacionales para el análisis de esos datos con el objetivo de convertirlos en información. Habitualmente es un experto quien mediante la visualización intenta descubrir patrones repetitivos y sus interrelaciones en el tiempo.

Dentro de la Minería de Datos existen técnicas para la extracción de patrones, las mismas serán en dependencia del conjunto de datos y del tipo de patrón que se quiere encontrar. Estas técnicas se dividen es descriptivas y predictivas, las primeras caracterizan las propiedades generales de los datos y por el contrario la predictiva realiza inferencias para poder realizar predicciones [36], [37].

Una técnica predictiva ampliamente utilizada son los árboles de decisión que definen un conjunto de clases, asignadas a cada dato de entrada una clase y determina la probabilidad de que ese registro pertenezca a la clase. En el árbol de decisión cada registro a clasificar fluye por una rama del árbol. La rama a seguir es determinada por una serie de preguntas definidas por los nodos de la rama. Cuando el registro llega a un nodo de hoja, se le asigna a la clase del nodo hoja [36].

En resumen, los árboles de decisión son sistema de predicción basados en reglas que llevan a cabo tests complejos que se ramifican hasta llegar al resultado óptimo.

Conclusiones

En este capítulo se ha realizado un análisis de los métodos que posibilitan detectar comportamientos estables o inestables, de ellos se comprobó que para poder obtener las dinámicas que representen comportamientos como pudiera ser un atractor, un ciclo límite u otro, es necesario que la serie temporal cuente con una cantidad específica o que por lo general contenga un número grande de datos o puntos como es el caso del atractor de Hénon que se reconstruya a partir de 500 datos. Por ello el método más apropiado es el Exponente de Lyapunov el cual se calcula no sólo a partir de la serie temporal, sino también del modelo, con lo que se obtiene un resultado mucho más exacto que con la serie temporal.

También se estudiaron herramientas informáticas que implementan algunos de los métodos planteados y otras que disminuyen el tiempo de cómputo para la obtención de los resultados.

Capítulo 2: Materiales y Métodos

En este capítulo se describen aquellos materiales, métodos y modelos que fueron utilizados a lo largo de este trabajo, mencionando los aspectos principales que los caracterizan.

2.1 Materiales

2.1.1 Lenguaje Java

Java ofrece todas las funcionalidades de un lenguaje potente, se propone su utilización como lenguaje para la implementación de las funcionalidades dada las siguientes características:

- **Distribuido:** pues Java puede invocar métodos en un equipo remoto de manera sencilla y transparente, utilizando protocolos comunes como CORBA (Arquitectura intermediaria para solicitar objetos comunes) y RMI (Invocación de métodos remotos), además de los servicios Web, como Servlets y Portlets.
- **Interpretado:** los programas en Java son interpretados, en lugar de ser compilados a código de máquina, el código de Java es traducido a código de bytes que permite que el código sea ejecutable en cualquier plataforma que tenga una Máquina Virtual de Java (JVM) sin necesidad de ser recompilado.
- **Robusto:** Java contiene varias funcionalidades que le permiten mejorar la fiabilidad de las aplicaciones como son el no tener punteros, lo que evita que un usuario pueda manipular segmentos de la memoria. Por otra parte tiene un recolector de basura que elimina de la memoria, de manera automática, elementos que no están siendo utilizados, evitando que los programadores olviden accidentalmente liberar la memoria o tengan que preocuparse por saber que espacio de memoria liberar.
- **De arquitectura neutral:** se dice que Java es arquitectónicamente neutral por ser su código independiente de cualquier plataforma en la que se esté ejecutando. La plataforma que necesite ejecutar programas desarrollados en Java sólo necesita tener una JVM.
- **Portable:** puede ser ejecutado en varias plataformas sin necesidad de cambios en su código, garantizado por la máquina virtual.

- Multitarea: Java incluye muchos recursos que facilitan el proceso multitarea, garantizando la seguridad de los procesos en ejecución [24].

2.1.2. Netbeans

Para el desarrollo de las funcionalidades fue necesario escoger la herramienta de desarrollo a utilizar según el lenguaje de programación que se seleccionó, por lo tanto se hizo uso del IDE Netbeans que es una herramienta para el desarrollo de aplicaciones de escritorio usando Java. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las APIs de NetBeans y un archivo especial que lo identifica como tal.

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole otros nuevos y pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios. Además es un proyecto GNU (libre), tiene un excelente diseñador de interfaces integrado, es muy rápido y fácil de usar [38].

2.1.3. BioSyS versión 1.0

Las funcionalidades que se implementaron fueron integradas dentro de la primera versión de BioSyS (Biological System Simulator o Simulador de Sistemas Biológicos) que es un software desarrollado en el polo de Bioinformática de la Universidad de Ciencias Informáticas. Dicha integración es motivada por tener esta herramienta el objetivo de permitir hacer estudios de los sistemas dinámicos y entre ellos los biológicos, aunque puede ser visto de propósito general mientras sean procesos modelados a partir de ecuaciones diferenciales.

BioSyS es un software que integra algoritmos y herramientas necesarias para servir de apoyo en las investigaciones de aquellos científicos dedicados al estudio de la Biología de Sistemas tales como la modelación, la simulación de manera local (y utilizando computación GRID) y la realización de análisis de los resultados de las simulaciones usando técnicas de Inteligencia Artificial. Ha sido programado siguiendo un diseño modular a partir de un

modelo arquitectónico en pizarra. Como centro del sistema se tiene una Base de Datos (BD) relacional, sobre la cual se escriben y se leen todos los demás módulos. En la actualidad se trabaja en la versión 2.0 extendiendo sus prestaciones.

2.1.4. Usando Matlab en BioSyS versión 1.0

Específicamente los estudios dentro de BioSyS se basan en las simulaciones que se obtienen a partir de los modelos matemáticos utilizando métodos numéricos de resolución. Con este fin se ha incluido dentro del software BioSyS aquellas librerías del asistente matemático Matlab que posibilitan tener soluciones de ecuaciones diferenciales ordinarias, básicamente las que permiten trabajar con matrices mediante los métodos numéricos, pues Matlab tiene implementado varios de ellos como: Ode45, Ode23, Ode113, Ode15s, Ode23s, Ode23t, Ode23tb, Ode15i.

Por lo tanto en la metodología se incluye esta herramienta, pero además de tener el fin de poder obtener simulaciones al variar parámetros y condiciones iniciales, fundamentalmente se seleccionó porque tiene integrado entre sus muchas funcionalidades aquella que permite el análisis de la estabilidad.

Se determinó aplicar esta funcionalidad de Matlab y no de otra herramienta como pudo haber sido TISEAN porque a pesar de este ser un software específico para el análisis de series temporales y por tanto de sistemas dinámicos, el trabajo con el mismo no es sencillo o de fácil entendimiento además de que proporciona un resultado indirecto, es decir, que la solución que ofrece es necesario realizarle un procesamiento para obtener el resultado que realmente se quiere. Tal es el caso del cálculo del máximo exponente de Lyapunov que se puede calcular a través de dos comandos: `lyap_k` y `Lyap_r`, pero la salida para ambos es un fichero que contiene un conjunto de puntos con los que se conforman curvas y la más recta de ellas corresponde a la del máximo exponente, por esto si se quisiera conocer el valor de dicho exponente se debería realizar otro proceso adicional al que ofrece el software TISEAN.

2.1.5. Plataforma de Tareas Distribuidas T-arenal

El método de análisis de estabilidad a partir del espectro de Lyapunov está pensado para calcular sus valores teniendo en cuenta datos de parámetros fijos y con ello el resultado se obtiene en un tiempo considerable, en este sentido, se trata de poder disponer de un análisis más extendido lo cual se lograría al variar algunos parámetros y condiciones iniciales, pero esto traería como consecuencia que el tiempo de obtención del resultado se incrementaría exponencialmente con respecto al que se demora si solo se calcula el exponente de una dinámica. Para solucionar esta limitante se recurrió a la plataforma de cálculos distribuidos anteriormente descrita. Con la misma se podrá distribuir todo el cómputo resultante de realizar el cálculo del espectro para un modelo a partir de diferentes dinámicas. La plataforma se encargará de repartir las tareas de cálculo en aquellas máquinas que se encuentren disponibles y así poder soportar el procesamiento total, favoreciendo la reducción de los tiempos de respuesta de los resultados, aspecto relevante dentro de la presente investigación.

2.1.6. Weka

Con el propósito de brindar un resultado más específico, concreto y aprovechable para próximos estudios y posteriores tomas de decisiones, luego del análisis de la estabilidad obtenido de aplicar el espectro de Lyapunov, se empleó el software Weka para el trabajo con la Minería de Datos.

Weka es un software escrito en lenguaje Java 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. En los últimos años ha sido reconocido internacionalmente como uno de los software más utilizados en el área de la Minería de Datos, esto es debido a que incluye diversos estándares para la realización de tareas como: preprocesado de datos, clustering, clasificación, regresión, visualización y selección de características [39], [40].

Weka trabaja con una estructura de datos interna conocida como fichero *.arff*. La estructura de un fichero con formato *.arff* (Attribute-Relation File Format) es muy sencilla. Se divide en 3 partes: @relation, @attribute y @data [41].

- **@relation<relation-name>**: todo fichero .arff debe comenzar con esta declaración en su primera línea (no permite líneas en blanco al principio). La estructura **<relation-name>** será una cadena de caracteres y si contiene espacios se pone entre comillas.
- **@attribute<attribute-name><datatype>**: en esta sección se incluye una línea por cada atributo (o columna) que vayamos a incluir en el conjunto de datos, indicando su nombre y el tipo de dato. Con **<attribute-name>** se indica el nombre del atributo, que debe comenzar por una letra y si contiene espacios tendrá que estar entrecomillado. Con **<datatype>** se indica el tipo de dato para este atributo (o columna) que puede ser: **numeric** (numérico) **string** (texto) **date** [**<date-format>**] (fecha). En **<date-format>** se indica 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 el usuario propiamente y que pueden tomar una serie de valores que se indiquen.
- **@data** En esta sección se incluyen los datos propiamente dichos. Se separa cada columna por comas y todas las filas deberán tener el mismo número de columnas, número que coincide con el de declaraciones @attribute que se añaden en la sección anterior. Si no se dispone de algún dato, se coloca 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 [41], [42].

A continuación se muestra un ejemplo de un fichero *.arff, **Fig. 2.1**:

```
@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 2.1 Fichero de tipo *.arff.

2.2 Métodos

2.2.1. Espectro de Lyapunov

Entre los métodos analizados para el estudio de estabilidad se seleccionó el del Exponente de Lyapunov y no la Dimensión de Correlación o el Ploteo Recurrente pues a pesar de que en ocasiones no se cuenta con el modelo que rige al sistema y sí se tienen los datos, los dos últimos métodos anteriormente dichos, requieren que la serie temporal, para el estudio, sea lo suficientemente larga o contenga la cantidad de datos necesarios para poder realizar el análisis a la dinámica. Otra razón implicada es que al utilizar series temporales se pueden introducir errores, ruido [4].

Como se mencionó en el epígrafe 2.1.4, se recurrió al asistente Matlab que dispone de la función *lyapunov* implementada para el cálculo del espectro de Lyapunov a partir de ecuaciones diferenciales y que está basada en el método planteado por A. Wolf *et al.* [4]. Dicha función se encuentra dentro del toolbox para la investigación de sistemas dinámicos.

Se decidió determinar la estabilidad empleando el espectro de Lyapunov con vistas de poder tener un resultado más descriptivo. No se usó el máximo exponente, puesto que el espectro brinda una descripción más completa al mostrar en orden decreciente los valores de los exponentes y con ello un análisis cualitativo.

En el artículo de Wolf *et al.* explican que el espectro de Lyapunov proporciona una interpretación geométrica de la estabilidad. En otro artículo se plantea que “Si bien el mayor de los exponentes de Lyapunov es quien alerta sobre la sensibilidad a las condiciones iniciales, el espectro completo provee mucha más información acerca de la estructura local de la función a lo largo de la solución [43]”. En el mismo artículo se dice que “El espectro de Lyapunov brinda una descripción cualitativa del espacio de fase, ya que los espectros correspondientes a dos soluciones cualitativamente distintas poseen diferentes distribuciones de signos.”

Se define entonces el exponente en términos del tamaño del elipsoide principal como:

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \log_2 p_i(t)/p_i(0)$$

donde el espectro quedaría definido: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_n$, la magnitud del elipsoide crece como $2^{\lambda_i t}$, el área definida por el primero de los dos ejes principales es $2^{(\lambda_1+\lambda_2)t}$ y el volumen definido por el primero de los tres principales ejes es $2^{(\lambda_1+\lambda_2+\lambda_3)t}$ [4].

Por otra parte el espectro “verifica, además de los signos, propiedades cuantitativas que se aplican para corroborar los resultados de las simulaciones. Por ejemplo, en el caso de un límite estable, el exponente de valor cero refleja la dirección de estabilidad neutra, tangente a la órbita, mientras que el resto de los exponentes corresponden a direcciones de contracción transversales [43]”.

Para el cálculo del espectro de Lyapunov se debe integrar simultáneamente el sistema junto con n copias de $z(t) = \left(\frac{\partial f}{\partial x}\right)_{x(t)} z(t)$ que es la ecuación variacional que gobierna la dinámica de la perturbación y aplicar según el método de renormalización de Gram-Schmidt, lo siguiente:

$$w_i(t_k^+) = \sum_{j=i}^{i-1} [z_i(t_k^-) * z_j(t_k^+)] z_j(t_k^+)$$

$$a_i(t_k) = \| z_i(t_k^-) - w_i(t_k^+) \|, z_i(t_k^+) = \frac{z_i(t_k^-) - w_i(t_k^+)}{a_i(t_k)}$$

para $i = 1, \dots, m$ suficiente cantidad de veces. Luego deben integrarse un total de $n(n + 1)$ ecuaciones diferenciales de primer orden durante un intervalo de tiempo relativamente extenso [4], [44].

Las estimaciones del espectro de Lyapunov, dada la complejidad matemática es lenta, es decir, la integración y la descomposición matricial que se realizan, exigen que se implementen estas tareas con gran eficiencia, desarrollando métodos de mayor simplicidad de cálculo que requieran de menor tiempo de cómputo [43].

Se debe señalar que este análisis de estabilidad que realiza el método del espectro de Lyapunov o en general para encontrar el valor de un exponente, sólo es de un juego de valores de condiciones iniciales y parámetros, es decir, de únicamente una sola dinámica del sistema.

2.2.2. Árboles de decisión. Algoritmo J-48

Con la intención de poder utilizar los resultados del análisis de estabilidad en estudios futuros del mismo tema, pudiendo aplicar nuevas

estrategias, se pretende obtener modelos de clasificación que proporcionen una orientación que indiquen cierta determinación a seguir a partir del patrón de comportamiento que pueda ser identificado. Para ello se dispuso de la Minería de Datos, en específico de la técnica conocida como árboles de decisión.

Los árboles de decisión representan reglas donde atributos independientes determinan los valores finales. Esta técnica lleva a cabo un test a medida que este se recorre hacia las hojas para alcanzar así una decisión. El árbol de decisión suele contener nodos internos, nodos de probabilidad, nodos hojas y arcos. Un nodo interno contiene un test sobre algún valor de una de las propiedades. Un nodo de probabilidad indica que debe ocurrir un evento aleatorio de acuerdo a la naturaleza del problema, este tipo de nodos se representa con círculos, los demás con cuadrados. Un nodo hoja representa el valor que devolverá el árbol de decisión y finalmente las ramas brindan los posibles caminos que se tienen de acuerdo a la decisión tomada [45].

El factor decisivo para la aplicación de esta técnica de clasificación es su capacidad de construir modelos interpretables. También se tuvo en cuenta las siguientes características: es método robusto a datos ruidosos (con errores), capaz de aprender expresiones disyuntivas y es muy utilizado y práctico para inferencia inductiva [46].

La técnica Árboles de Decisión se encuentra implementada dentro de las herramientas de clasificación que están incluidas en el software Weka que como se dijo anteriormente fue utilizado. Para la construcción de los árboles de decisión se seleccionó el algoritmo J-48 igualmente implementado en dicha herramienta.

El algoritmo J-48 es una implementación dentro del software Weka del algoritmo C4.5, este último a su vez es una versión actualizada del algoritmo ID3 [47] que fue propuesto por [48]. El J-48 construye árboles de decisión permitiendo podar antes de llegar a las hojas de cada subárbol una vez que éste ha sido inducido con lo que se puede evitar el overfitting de atributos nominales con valores únicos [39]. La poda está basada en la aplicación de un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no una determinada rama. El algoritmo J-48 se basa en la utilización del criterio ratio de ganancia (gain ratio). De esta manera se consigue evitar

que las variables con mayor número de posibles valores salgan beneficiadas en la selección.

2.3. Modelos matemáticos utilizados

Durante el desarrollo de este trabajo de tesis se utilizaron algunos modelos los cuales serán descritos a continuación.

2.3.1. Modelo de Lorenz

El meteorólogo Edward Lorenz diseñó un modelo matemático simplificado basado en tres ecuaciones diferenciales bien conocidas en el ámbito de la física de fluidos [49], [50]:

$$\frac{dx}{dt} = \delta(y - x)$$

$$\frac{dy}{dt} = rx - y - xz$$

$$\frac{dz}{dt} = xy - bz$$

Este modelo consiste en un conjunto de tres ecuaciones diferenciales en la que cada uno de los términos dx/dt ; dy/dt ; dz/dt indican lo que varía cada una de *variables* como consecuencia de las relaciones que se dan entre ellas y los *parámetros del sistema* (δ, r, b), siendo:

x la velocidad de rotación de un cilindro de masa gaseosa,

y la diferencia de temperatura en los extremos del cilindro,

z la desviación de la temperatura del sistema,

δ el parámetro relacionado con la viscosidad y la conductividad térmica de la masa de aire,

r el parámetro relacionado con la diferencia de temperatura entre la parte superior e inferior de la columna,

b el parámetro relacionado con la altura y anchura de la misma.

2.3.2. Modelo Lotka-Volterra (Presa-Depredador)

El modelo representa la interdependencia entre dos especies que conviven, y una de ellas es depredadora de la otra:

$$\frac{dx}{dt} = ax - bxy$$

$$\frac{dy}{dt} = -cy + dxy$$

Siendo:

x el número de presas,

y el número de depredadores,

dx/dt y dy/dt crecimiento de las dos poblaciones en el tiempo,

a la tasa de crecimiento de las presas(o tasa de natalidad),

b la tasa de eliminación de las presas por parte de los depredadores (o productividad de los depredadores o eficiente de captura),

c la tasa de eliminación natural de los depredadores (o tasa de mortalidad de los predadores),

d la tasa de crecimiento de los predadores como resultado del consumo de presas (o tasa de transferencia energética de las presas a los predadores) [51].

2.3.3. Dinámica de las células T CD4 + y el virus HIV_1

El modelo fundamentalmente describe la dinámica que existe entre las células CD4+ y el virus HIV_1 mediado por antígenos. Con este modelo se verifica la acción de una replicación viral residual bajo una terapia retroviral activa [52]:

$$\frac{dx(\varepsilon)}{dt} = \varepsilon\mu x(\varepsilon) \left[1 - \frac{n(\varepsilon)}{f(\varepsilon)} - \Lambda(v) \right]$$

$$\frac{dy(\varepsilon)}{dt} = p\varepsilon\mu x(\varepsilon)\Lambda(v) + \varepsilon\mu y(\varepsilon) \left[(2p - 1) - \frac{n(\varepsilon)}{f(\varepsilon)} \right]$$

$$\frac{dy_a}{dt} = (1 - p)\mu \int \varepsilon[\Lambda(v)x(\varepsilon) + y(\varepsilon)d\varepsilon - \delta y_a]$$

$$\frac{dv}{dt} = \lambda\delta y_a - cv - \int \varepsilon\mu x(\varepsilon)\lambda(v)d\varepsilon$$

2.3.4. ICMODEL

El siguiente modelo nombrado ICMODEL describe la cinética de procesos inmunológicos conocidos y un proceso patológico adicional sobre el daño del tejido del complejo inmune y la producción de antígenos, fue utilizado para:

1. comprar respuestas cinéticas simuladas con las reportadas
2. evaluar la relativa estabilidad del Sistema Inmune
3. determinar las condiciones requeridas para la producción de altos niveles del complejo inmune como los encontrados en pacientes con enfermedades de este tipo [53].

Donde:

$$\frac{dAg}{dt} = k_0 - k_1AgM - k_5AgAb + k_{19}IC$$

$$\frac{dAb}{dt} = k_4P - k_5AgAb - k_{17}Ab$$

$$\frac{dC}{dt} = k_{13} \left(\frac{1 + k_{20}IC}{1 - ICC} \right) - nk_6ICC - k_{18}C$$

$$\frac{dIC}{dt} = k_5AgAb - k_6IC C - k_9ICFc$$

$$\frac{dICC}{dt} = k_6ICC - nk_7ICC CR_1 - k_9ICC Fc$$

$$\frac{dICCc}{dt} = k_8CR_1ICC \frac{I}{n} - k_9ICCcFc - nk_{11}ICCcCR_2$$

$$\frac{dCR_1ICC}{dt} = nk_7ICC CR_1 - \frac{k_8}{n} - CR_1ICC I$$

$$\frac{dBFc}{dt} = k_9(IC + ICC + ICCc)Fc - k_{10}BFc$$

$$\frac{dCR_2ICCc}{dt} = k_{11}ICCc CR_2 - k_{12}CR_2ICCc$$

$$\frac{dTc}{dt} = k_1AgM + k_3AgTc - k_{15}Tc$$

$$\frac{dBc}{dt} = k_1AgM + k_2AgBc I_1 - k_{14}Bc$$

$$\frac{dP}{dt} = pk_1AgM + pk_2AgBc I_1 - k_{16}P$$

$$\frac{dI_1}{dt} = mk_1AgM + mk_3AgTc - k_2AgBc I_1$$

2.3.5. Modelo de la Interleuquina (IL-2)

Este modelo muestra el papel de la IL-2 en la interacción entre células CD4+ y las células T reguladoras [54].

Donde:

$$\frac{de_N}{dt} = \sigma_E - kE_A e_{B_N} \left(1 - \frac{rB_T}{S}\right)^{n(s-1)} + 2kE_P e_C + akE_S \left(\frac{\left(\frac{S_E}{riEA}\right)^h}{\left(\frac{S_E}{riEA}\right)^h + il2_B^h + ila^h} \right) e_A - eF_N$$

$$\frac{de_A}{dt} = kE_N \left(1 - \frac{rB_T}{S}\right)^{n(s-1)} - kE_S e_A$$

$$\frac{de_C}{dt} = kE_S \left(\frac{(il2_B)^h + ilail^h}{\left(\frac{S_E}{riEA}\right)^h + il2_B^h + ilail^h} \right) e_A - kE_P e_C$$

$$\frac{dr_N}{dx} = \sigma_R - kR_A r_{B_N} + 2kR_P r_C + akR_S \left(\frac{\left(\frac{S_R}{riRA}\right)^h}{\left(\frac{S_R}{riRA}\right)^h + il2_B^h} \right) r_A - k_d rF_N$$

$$\frac{dr_A}{dx} = kR_A r_{B_N} - kR_S r_A$$

$$\frac{dr_C}{dx} = kR_S \left(\frac{il2_B^h}{\left(\frac{S_R}{riRA}\right)^h + il2_B^h} \right) r_A - kR_P r_C$$

$$\frac{dil2}{dx} = \sigma_i + ki_P e_{B_N} \left(1 - \frac{rB_T}{S}\right)^{s-1} - ki_d il2_F - kE_{in} (riREAe_A + riECe_C) il2_B - kR_{in} (riRNr_N + riRAR_A + riRCr_C) il2_B$$

Conclusiones

En este capítulo se trataron aquellos métodos, herramientas y modelos utilizados en la realización de este trabajo, ellos son: lenguaje Java, herramienta de desarrollo Netbeans, asistente matemático Matlab, Weka como herramienta para la Minería de Datos (de ella se utilizó el algoritmo J-48 para la creación de árboles de decisión), T-arenal para la distribución de las tareas de cálculo y el método del espectro de los exponentes de Lyapunov para el análisis de estabilidad. Se emplearon los modelos matemáticos de Lorenz, Lotka-Volterra, ICMODEL, IL-2 y el de la dinámica de las células T CD4+ con el virus HIV_1.

Capítulo 3: Resultados y Discusión

En este capítulo se muestran los principales resultados obtenidos en el presente trabajo, los cuales dan respuesta a los objetivos planteados para la investigación. En el mismo se explica la implementación distribuida del algoritmo de Lyapunov seleccionado y se muestra cómo se reducen considerablemente los tiempos de cómputo cuando se hace uso del mismo. También se explica la utilización de la Minería de Datos para analizar la estabilidad de múltiples dinámicas y se presenta otro tipo de análisis que fue adicionado para mostrar gráficas con información valiosa al investigador. Particularmente se explica la metodología propuesta como centro de esta investigación, utilizando todos los resultados anteriormente dichos y la inclusión de la misma dentro del software BioSyS. Finalmente para comprobar importantes pasos de la metodología, esta fue aplicada a un caso de estudio.

3.1. Algoritmo distribuido para el cálculo de los exponentes de Lyapunov para múltiples series temporales n-dimensionales

Como se planteó en el capítulo anterior, se utilizó el espectro de los exponentes de Lyapunov calculado a partir del modelo matemático. Se pudo comprobar como se muestra en la **Tabla I**, que el tiempo de cómputo aumentaba al aumentar el número de ecuaciones, esto no se puede absolutizar puesto que para los casos en que el proceso de integración numérica es más complejo no se tendrá igual resultado, como es el caso del modelo de Lorenz en comparación con el del modelo de la dinámica de las células T CD4 +, donde en el primero se tienen tres ecuaciones y se demora 23 minutos, sin embargo en el segundo se tienen cuatro ecuaciones y se demora 15 minutos. En la misma figura se observa que el tiempo requerido para analizar un sistema de alta dimensionalidad, como es el ejemplo del modelo ICMODEL formado por 13 ecuaciones, fue de 260 minutos. También se debe tener en cuenta los intervalos de los tiempos inicial y final que hayan sido tomados para el estudio, debido a que de ser un rango de tiempo considerablemente grande, implicará un mayor tiempo de cálculo, pues serán más puntos a evaluar en el tiempo definido para el análisis.

Por otra parte se analizó que generalmente los software que implementan el análisis de estabilidad a partir del espectro de los exponentes de Lyapunov lo hacen utilizando una sola dinámica. Como una de las intenciones de este trabajo es poder realizar el estudio de estabilidad en más de una dinámica, se realizó el mismo para 20 series temporales obtenidas del atractor de Lorenz, usando como valores de condiciones iniciales y de parámetros los que se especifican en la sección 3.1.2, obteniéndose el resultado en un tiempo de 7.52 horas, usando una sola PC, **Tabla II**, Prueba No.1.

Por las dos razones anteriores se vio la necesidad de disminuir el tiempo causado por el número elevado de ecuaciones que generalmente se estudian y la cantidad de series que se requiera analizar, para ello se diseñó un algoritmo teniendo en cuenta la complejidad que encierra el cálculo para obtener el espectro de los exponentes de Lyapunov dado este cúmulo de datos y por lo tanto el objetivo de dicho algoritmo es distribuir las tareas de procesamiento.

Tabla I

Nombre del Modelo	Cantidad de Ecuaciones	$t_0 = t_f$	Tiempo de cálculo (minutos)
Lotka-Volterra	2	20 – 30	1.5
Lorenz	3	0 – 100	23
Dinámica de las células T CD4 + y el virus HIV_1	4	0 – 10	15
IL-2	7	0 – 10	54
ICMODEL	13	0 – 100	260

Teniendo como finalidad la distribución de las tareas para el cálculo del espectro de los exponentes de Lyapunov haciendo uso de la plataforma de tareas distribuidas T-arenal, se hace necesaria la implementación de dos clases en Java básicamente (BSDDataManager y BSAlgorithm), aunque estas

dos clases pueden hacer uso de funcionalidades incluidas en otras clases o paquetes. La primera de ellas, que es la que se ejecuta en el servidor es la encargada de distribuir las tareas a las máquinas clientes, integrar los resultados provenientes de cada cliente y procesar los mismos. En este caso esta clase es la que se encarga de recibir los datos necesarios para hacer las simulaciones, enviar a cada cliente la unidad de trabajo correspondiente y recibir de estos últimos el resultado de las simulaciones.

La clase BSAAlgorithm.class es la que se ejecuta en cada uno de los clientes y es la encargada de gestionar la conexión entre Java y el Matlab, manda a ejecutar las funciones de Matlab que contienen la implementación de los algoritmos para el cálculo de los exponentes de Lyapunov y recibe los resultados de este procesamiento, los cuales son enviados de vuelta al servidor.

A continuación se hace una explicación por pasos de cómo funciona el flujo de trabajo de esta implementación.

3.1.1. Algoritmo distribuido para el cálculo del espectro de los exponentes de Lyapunov

Paso 1. Preparación de los datos en el servidor:

a) Se cargan los valores de las condiciones iniciales y parámetros para el estudio.

b) se cargan las propiedades para el estudio: nombre del modelo matemático, método numérico, tiempo inicial, tiempo final, paso de integración, granularidad inicial (cantidad inicial de instancias a enviar a las PCs clientes), máximo de granularidad, identificador del modelo, IP del servidor de la BD, puerto de la BD, nombre de la BD, usuario de la BD, contraseña de la BD, gestor de la BD)

Paso 2. Distribución de las tareas: se verifica que la PC cliente que se reportó es una de las PCs que se usarán para realizar el estudio. Luego genera las unidades de trabajo que le serán enviadas.

Paso 3. Procesamiento de las tareas: se reciben las unidades de trabajo que le fueron enviadas, se descarga en el cliente la función de Matlab que será usada (BSlyapunov), se extrae de la BD el modelo matemático que será usado en el estudio, se ejecuta Matlab y se comienza a realiza el estudio. Se crea el

vector con los resultados de los análisis realizados que serán enviados al servidor.

Paso 4. Recepción de los resultados del análisis de estabilidad: se recibe el vector con los resultados de los análisis realizados por los clientes.

Ver en la **Fig. 3.1** los pasos del algoritmo descrito.

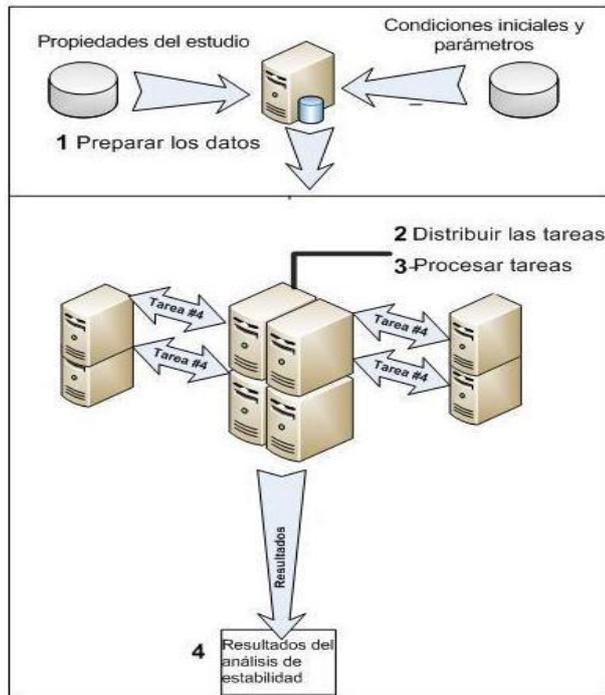


Figura 3.1 La imagen muestra los cinco pasos que se realizan para el análisis de estabilidad, a partir del algoritmo definido para distribuir las tareas en varias computadoras.

Para mayor profundización en el **Anexo 1** se puede ver el código del algoritmo descrito.

3.1.2. Tiempos de procesamiento

Con el fin de ilustrar cómo al distribuir las tareas para el cálculo del espectro de los exponentes de Lyapunov en varias PCs el tiempo de cómputo es menor que de hacerlo de manera secuencial en una sola computadora (Prueba No.1), se muestra a continuación en la **Tabla II**, el cálculo realizado (a partir del modelo de Lorenz) a la misma cantidad de combinaciones de valores y de parámetros (20 series), siguiendo el algoritmo distribuido propuesto, pero realizado en diferentes cantidades de computadoras.

Los valores usados fueron: $x_0 = 0$, $y_0 = 1$, $z_0 = 0$ y los valores de los parámetros: $s = 10$, $b = 8/3$. Se trabajó con los valores de r desde 20 a 30 y en

este rango se tomaron 20 puntos. El $t_0 = 0$, $t_f = 100$ y paso de integración 0.1.

Tabla II

Prueba No.	Cantidad de combinaciones*	Cantidad de PCs Utilizadas	Tiempo de cálculo (horas)
1	20	1	7.52
2	20	2	5.20
3	20	4	4.53
4	20	6	3.35
5	20	8	1.57

* Juego de valores de los parámetros y las condiciones iniciales.

Como se observa en la tabla en la medida que se aumenta el número de PCs disminuye el tiempo de procesamiento, yendo desde 7.52 horas en una sola PC hasta 1.57 horas en ocho PC. Haber disminuido el tiempo del estudio constituye un importante resultado, pues permite optimizar el tiempo dedicado a cada análisis, de manera que se puedan utilizar todas las PCs disponibles.

3.2. Minería de Datos para el análisis de estabilidad

La implementación del algoritmo del Lyapunov en ambiente distribuido brinda la posibilidad de realizar estudios complejos de la estabilidad de los sistemas dinámicos ante la variación de varios parámetros de control e incluso de las condiciones iniciales. Sin embargo se hace necesaria la implementación de técnicas que permitan interpretar los resultados obtenidos.

Si bien ya se explicaba que durante la investigación de este trabajo no se encontraron software que permitieran la realización de estudios de estabilidad haciendo uso del espectro de los exponentes de Lyapunov para más de una serie temporal, tampoco se hallaron implementaciones de técnicas de análisis para estos tipos de resultados.

Con el objetivo de resolver este problema se decidió incorporar una técnica de Minería de Datos que ayude a los investigadores a trazarse estrategias con respecto al análisis de múltiples dinámicas brindando así la posibilidad de obtener patrones de comportamiento. Como se dijo en el capítulo 2 de este trabajo, se utilizó el software Weka y con esta herramienta se implementó un algoritmo, dentro del software BioSyS (al cual se le aplicó la

metodología que se propondrá en este documento), para obtener información a partir del análisis de estabilidad realizado. Los pasos del algoritmo se explican en la siguiente sección.

3.2.1. Algoritmo para el análisis de los resultados de la estabilidad mediante técnicas de minería de datos

- **Paso 1.** Se carga el fichero *.arff obtenido del análisis de estabilidad.
- **Paso 2.** Se inicializa el algoritmo J48.
- **Paso 3.** Se clasifica el fichero con el algoritmo J48.
- **Paso 4.** Se visualiza el árbol de decisión.
- **Paso 5.** (Opcional). Se gráfica el comportamiento de la estabilidad de las dinámicas de un parámetro contra otro.

En la **Fig. 3.2** se pueden ver los pasos del algoritmo.

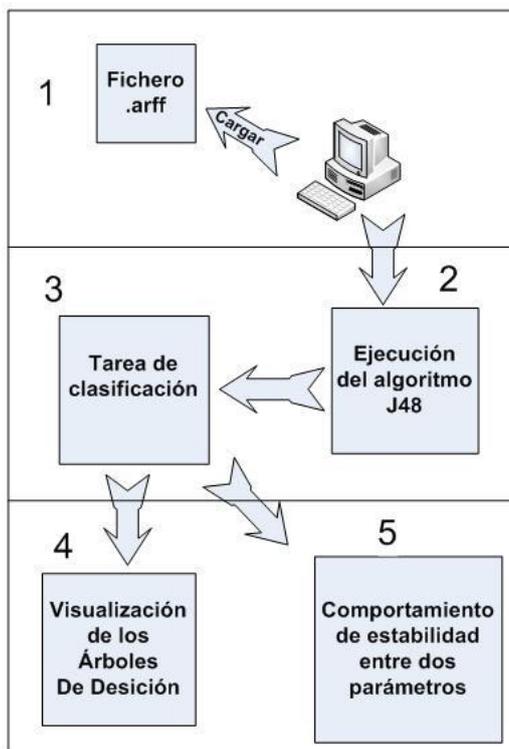


Figura 3.2 Se muestran los pasos del algoritmo de minería de datos implementado para la interpretación de los resultados obtenidos en el proceso de análisis de estabilidad realizado por el método del espectro de los exponentes de Lyapunov.

3.2.2. Aplicación del algoritmo de Minería de Datos a un caso de estudio

Para mostrar la utilidad de este algoritmo a la hora de visualizar una interpretación de los resultados del análisis de estabilidad, se tomó el modelo matemático Presa-Depredador, descrito en el capítulo 2. Se realizó un estudio de la estabilidad del mismo y se visualizó el árbol de decisión obtenido. Los valores utilizados fueron los siguientes:

$$x_0 = 1, y_0 = 2$$

b = variando desde 2 hasta 6 y en este rango se tomaron 5 puntos

c = variando desde 3 hasta 10 y en este rango se tomaron 10 puntos

$$a = 1$$

$$d = 4$$

$t_0 = 0$, $t_f = 5$ y paso de integración 0.1

Estos juegos de datos generan un total de 66 simulaciones, las cuales demoraron un tiempo 2.09 horas en una sola PC. Con el fichero generado, (ver **Anexo 2**), se procedió a crear el árbol de decisión correspondiente.

En la **Fig. 3.3** se muestra para cada hoja cuántos ejemplos de la evidencia fueron cubiertos y se obtuvo una precisión del modelo del 95%. El modelo de clasificación fue conformado por cuatro reglas. Para este caso se obtuvo:

1. si se tiene un rango de valores entre $c \leq 4.4$ y $b \leq 3.6$ se tendrán comportamientos inestables.
2. si se tiene un rango de valores entre $c \leq 4.4$ y $b > 3.6$ y $c \leq 3.7$ se tendrán comportamientos estables.
3. si se tiene un rango de valores entre $c \leq 4.4$ y $b > 3.6$ y $c > 3.7$ se tendrán comportamientos inestables.
4. para valores de $c > 4.4$ se tendrán comportamientos inestables.

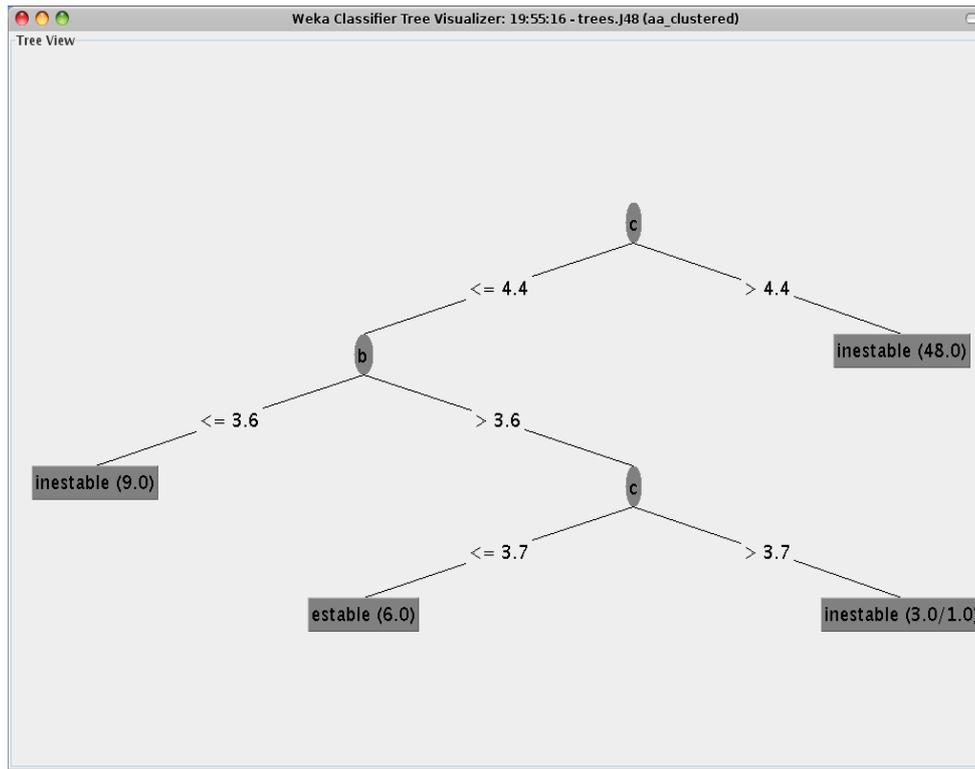


Figura 3.3 Árbol representado gráficamente a partir del modelo creado con las reglas obtenidas de los resultados del análisis de estabilidad.

La utilidad de esta técnica radica en que permite visualizar un estudio que tenga más de dos parámetros de control, debido a que generalmente un investigador únicamente puede graficar los valores de los parámetros de este tipo de análisis en gráficas de dos dimensiones, sin embargo con los árboles de decisión se puede visualizar un estudio que tenga implicados varios parámetros y extraer las reglas lógicas que rigen los diferentes comportamientos. Pero además con el modelo de clasificación generado se pueden clasificar otras dinámicas que no hayan sido clasificadas.

Este modelo, al crear rangos donde existirán zonas de inestabilidad o estabilidad, permite ahorrar tiempo de estudio de los investigadores al conocer para qué intervalos de los valores de los parámetros se define dicha clasificación.

3.3. Análisis del comportamiento de la estabilidad a partir de dos parámetros

Además de la visualización de árboles de decisión, en este trabajo se quiso dar otro tipo de análisis basado en los resultados del análisis de la estabilidad para así mostrar gráficas con información valiosa. Por ello se

desarrolló una funcionalidad que permite graficar el comportamiento de la estabilidad correspondiente a dos parámetros que sean seleccionados para este tipo de estudio. La gráfica de dos dimensiones que se visualiza se realiza usando los resultados del análisis de estabilidad almacenados en el fichero *.arff.

Para demostrar este resultado de la investigación se utilizó nuevamente el modelo Presa-Depredador y los valores usados en el estudio fueron los mismos del epígrafe 3.2.2.

Se seleccionaron los parámetros b y c para ser analizados y se obtuvo la siguiente gráfica, **Fig. 3.4**.

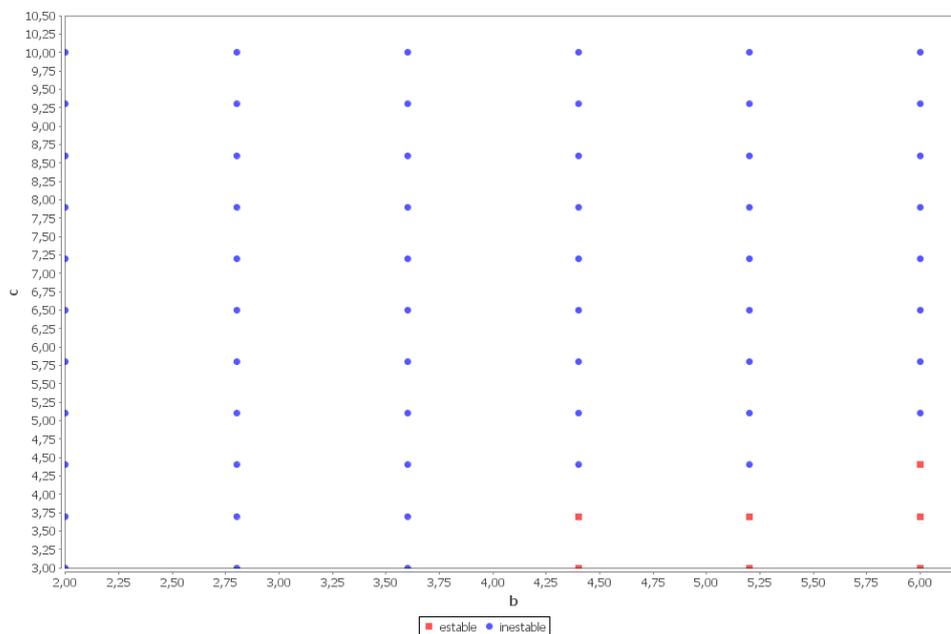


Figura 3.4 Como se muestra, del juego de datos utilizado se obtuvo una zona de estabilidad para los mayores valores del parámetro b y para los menores del parámetro c , este análisis visual bien pudiera guiar a los investigadores al enmarcar dichos comportamientos.

3. 4. Metodología propuesta

El análisis de series temporales, desde el punto de vista de su comportamiento, tanto pasado como futuro, requiere el uso de nuevas técnicas, pues las presentadas hasta el momento, aunque le son aplicables, no cubren las necesidades que surgen en el tratamiento de este tipo de datos, como por ejemplo realizar estudios de varias dinámicas a la vez al variar diferentes parámetros. Es por ello que a continuación se plantea la propuesta de una

metodología para poder estudiar dinámicas complejas. Dicha metodología fue aplicada al software BioSyS 1.0 por lo que muchas de sus interfaces fueron utilizadas, además le fueron añadidas al software aquellas funcionalidades que propician un estudio completo de la estabilidad, como se contempla en la metodología.

Descripción de la metodología:

Paso 1. Selección o inserción del modelo matemático: se selecciona de la BD de BioSyS el modelo matemático de interés para realizar el estudio.

Paso 2. Selección de los parámetros a utilizar en el estudio: se eligen aquellos parámetros que se quiere analizar para comprobar cuánto influyen en la estabilidad de la dinámica.

Paso 3. Selección del rango de variación de los parámetros: se introducen los valores con los que se desea realizar el estudio.

Paso 4. Cálculo del espectro de los exponentes de Lyapunov del modelo: se ejecuta la acción que permitirá hacer el cálculo del espectro.

Paso 5. Análisis de estabilidad a partir del espectro de los exponentes de Lyapunov obtenido: Si el máximo exponente del espectro de Lyapunov es positivo, el sistema es inestable y si es negativo, el sistema es estable. El sistema será tanto más estable conforme menor sea, el exponente de Lyapunov.

Paso 6. Creación de un fichero *.arff: se creará un fichero donde se guardarán los valores de los parámetros que fueron variados y los resultados del análisis de la estabilidad para el modelo.

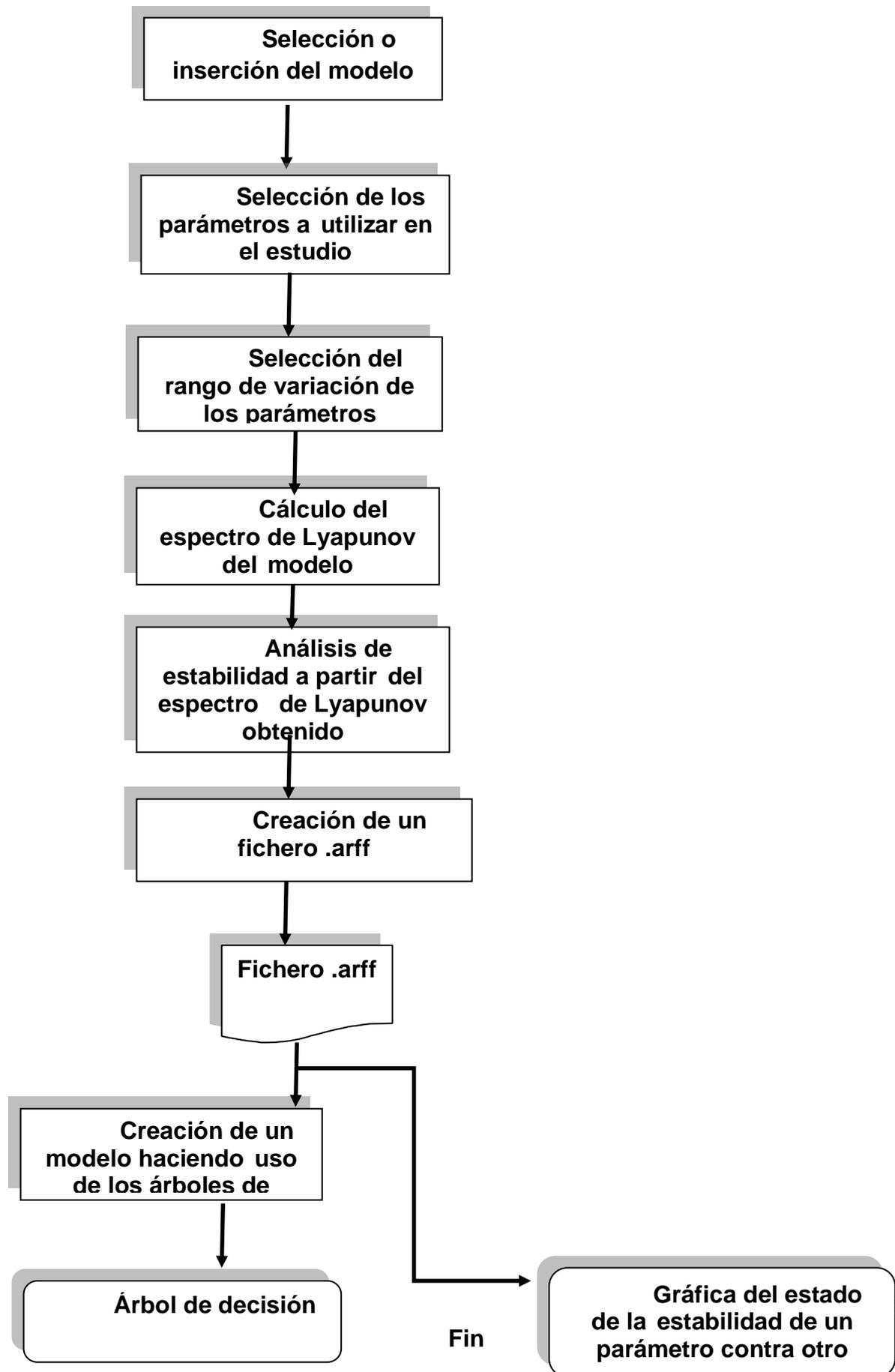
Paso 7. Creación de un modelo haciendo uso de los árboles de decisión: a partir del fichero *.arff se clasifica con el algoritmo J-48 implementado dentro del software Weka.

Paso 8. Árbol de decisión: se podrá visualizar el árbol a partir del modelo de clasificación obtenido.

También se podrá, teniendo el fichero *.arff creado en el paso 7, graficar el estado de la estabilidad de un parámetro contra otro seleccionando dos parámetros de interés dentro del modelo.

En la **Fig. 3.5** se esquematiza la metodología planteada:

Inicio



3.5. Aplicación de la metodología propuesta en un caso de estudio

3.5.1. Prueba No.1 realizada a la metodología

Para verificar los resultados obtenidos se acudió a aquellos reportados en la bibliografía a partir de las ecuaciones del modelo de Lorenz. Se utilizaron los valores registrados en el artículo [55], sin embargo no fue necesario usar el $t_f = 10000$, puesto que en el estudio realizado para un $t_f = 10$ se obtuvo la misma clasificación de estabilidad que se definió en dicho artículo.

El resto de los valores utilizados en la evaluación de la metodología fueron los siguientes:

$x_0 = 0$, $y_0 = 1$ y $z_0 = 0$ y los valores de los parámetros:

$$\delta = 10$$

$$b = 8/3$$

$r =$ desde 20 hasta 30 y en este rango se tomaron 10 puntos

En la **Fig. 3.6** No. Se muestra la interfaz de entrada de los datos en el software BioSyS.

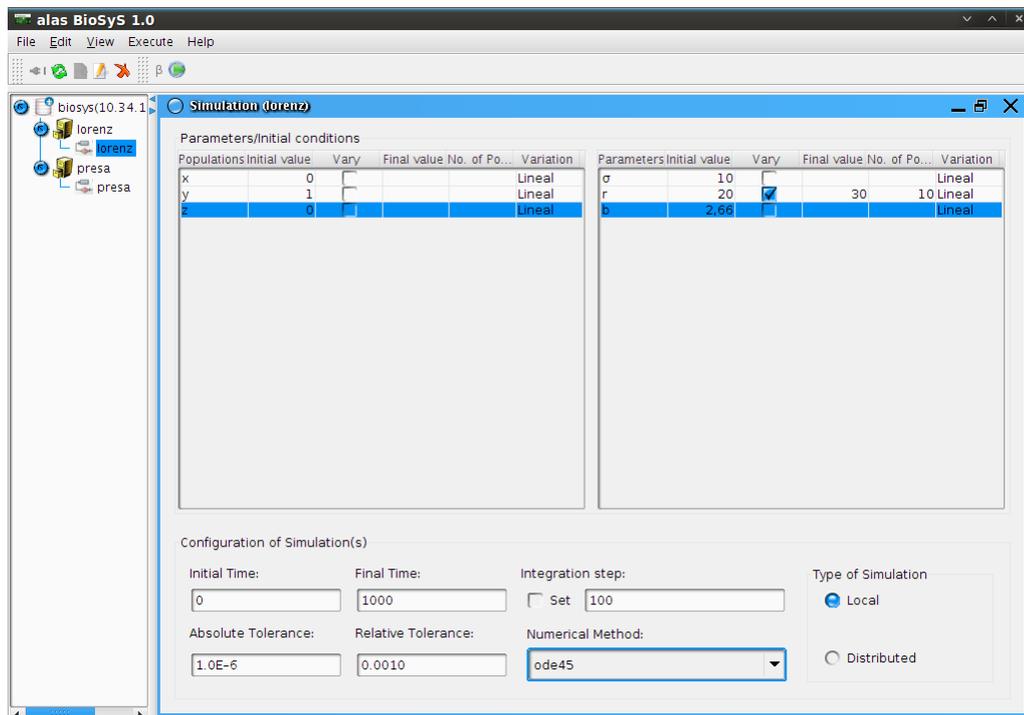


Figura 3.6. Se muestra en la Interfaz de BioSyS, donde fueron entrados los datos para el estudio.

Se obtuvo el espectro para cada una de las dinámicas y se guardaron los resultados en el fichero *.arff (ver **Anexo 3**). Luego se creó el modelo y se visualizó el árbol de decisión que se muestra debajo en la **Fig. 3.7**.

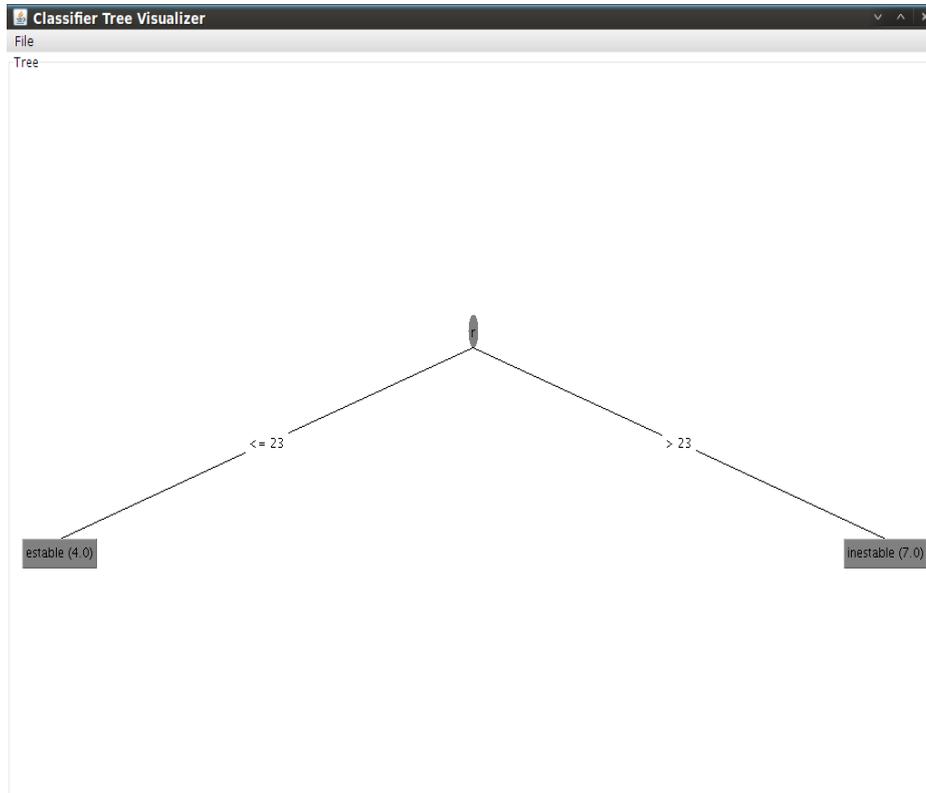


Figura 3.7 Se muestra en la imagen el árbol de decisión donde a partir de la variación del parámetro r se definió de acuerdo a los resultados de la estabilidad que valores de r menores e iguales que 23 las dinámicas se comportan estables, mientras que para valores mayores que 23 se comportan inestables.

Los resultados de la clasificación realizada por el modelo entran en los rangos de los valores de los dos tipos de comportamientos definidos en el artículo antes mencionado, lo que fue utilizado para verificar la metodología.

Para este caso de estudio no se puede mostrar la gráfica del comportamiento de la estabilidad de un parámetro contra otro, debido a que solamente se varió un parámetro.

3.5.2. Prueba No.2 realizada a la metodología

Adicionalmente se realizaron otras corridas para el mismo modelo, pero usando otros valores de variables y parámetros, de estos últimos se variaron dos parámetros, como se muestra a continuación:

Se seleccionaron los parámetros s y r a variar y se tomaron los valores iniciales: $x_0 = 1$, $y_0 = 0$ y $z_0 = 1$ y los valores de los parámetros, ver **Fig.3.8**:

δ = desde 5 hasta 10 y en este rango se tomaron 5 puntos

$b = 8/3$

r = desde 0 hasta 30 y en este rango se tomaron 10 puntos

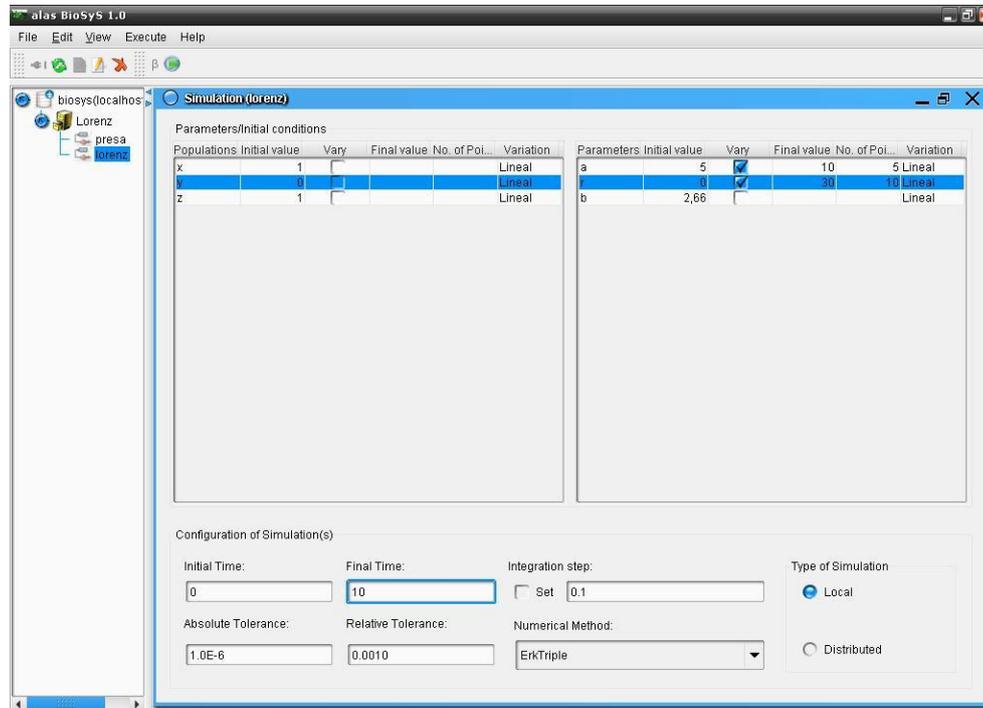


Figura 3.8. En esta imagen se muestra la interfaz de BioSys para la entrada de datos con los valores que fueron utilizados para este estudio y los parámetros variados.

El árbol de decisión del modelo generado a partir de los resultados de la estabilidad obtenido se muestra a continuación en la **Fig. 3.9**.

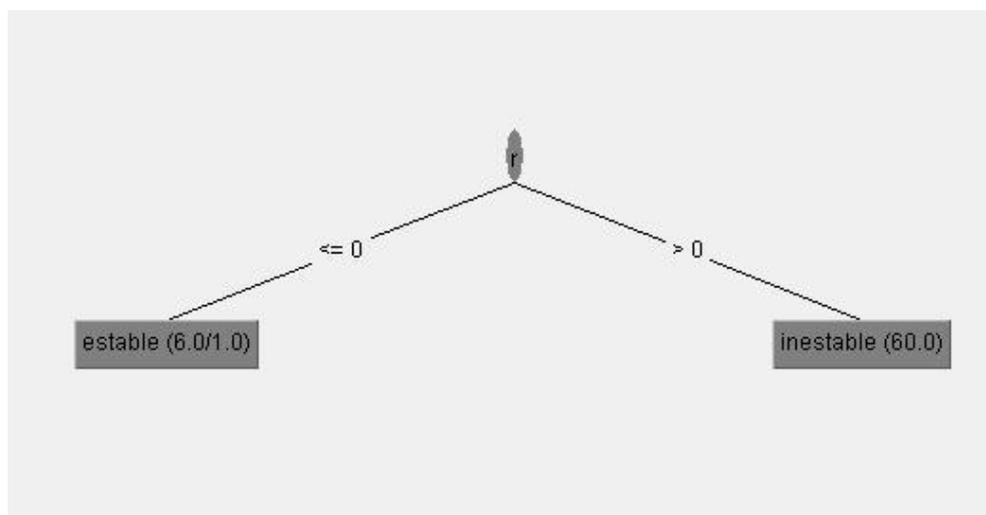


Figura 3.9 En esta imagen se puede observar que para el parámetro r cuando los valores estén por debajo de cero las dinámicas serán estables, mientras que para valores positivos de este parámetro las dinámicas se corresponderán a comportamientos inestables.

Aunque la ilustración de la gráfica del árbol de decisión es muy útil también se ofrece la oportunidad de poder graficar el comportamiento de la estabilidad al variar los parámetros a y r como se muestra en la **Fig. 3.10**.

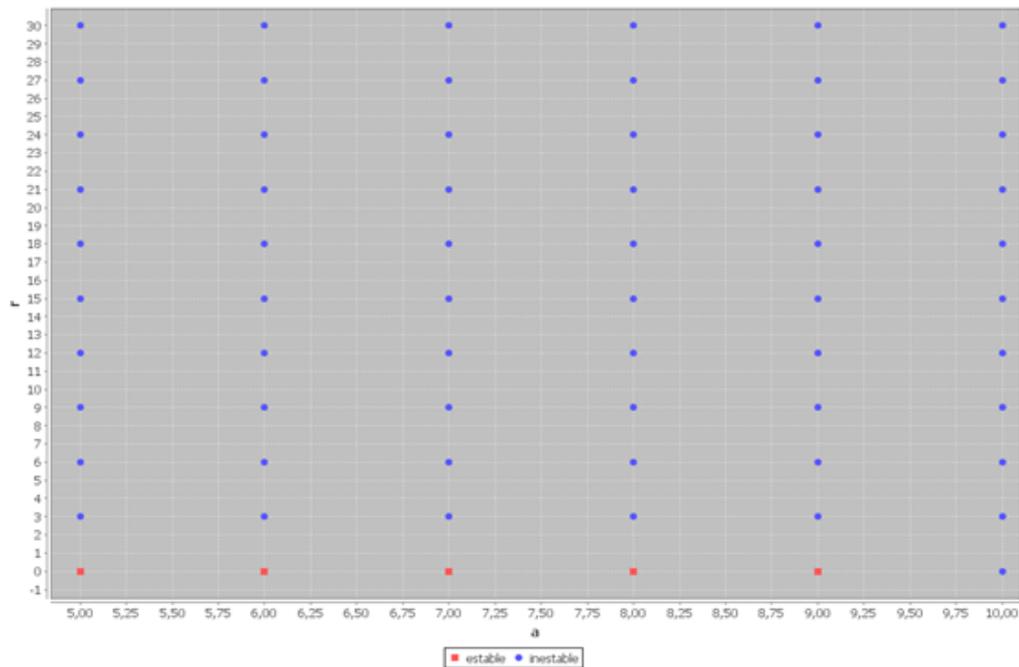


Figura 3.10 En esta imagen se muestra la estabilidad del parámetro r con respecto al parámetro a .

3.6. Metodología para el análisis de estabilidad de Sistema de Ecuaciones Diferenciales n-dimensionales incorporada dentro del software BioSyS 1.0

La metodología para el análisis de estabilidad de Sistema de Ecuaciones Diferenciales n-dimensionales definida en este trabajo, fue incluida dentro del software BioSyS 1.0 como se ha venido mostrando. Esto permitirá a los investigadores que utilicen el software, la posibilidad de realizar este tipo de estudio de una manera más organizada, siguiendo un conjunto de procedimientos que hacen uso de herramientas como el asistente matemático Matlab, el software de Minería de Datos Weka y de las funcionalidades necesarias que fueron

implementadas y agrupadas dentro de BioSyS, de manera que el investigador pudiese contar con la posibilidad de:

1. realizar análisis de estabilidad a partir de un modelo matemático en series de alta dimensionalidad y en muchas series a la vez.
2. reconocer qué parámetros definen determinada dinámica de un sistema.
3. definir rangos en los que los parámetros provocan estabilidad o inestabilidad.
4. permitir la clasificación de dinámicas aún sin clasificar.
5. visualización de los resultados del análisis de la estabilidad para su interpretación.
6. acortar tiempos de estudio.

Conclusiones

En este capítulo se mostraron los resultados de este trabajo partiendo del análisis de estabilidad. Se demostró que se acortó el tiempo de cómputo para dicho análisis al realizar el estudio tanto en series temporales de alta dimensionalidad como al hacerlo en más de una serie a la vez. Por otra parte se expusieron las dos salidas gráficas que ayudan a una mejor y más rápida interpretación de los resultados. Además se definió la metodología para el análisis de la estabilidad en series n-dimensionales que incorpora todas las funcionalidades anteriormente descritas, y finalmente se probó dicha metodología en un caso de estudio de la literatura, exhibiendo la misma, buenos resultados con respecto al artículo utilizado.

Conclusiones

- Se obtuvo una metodología para el análisis de estabilidad de Sistemas de Ecuaciones Diferenciales n-dimensionales.
- Se desarrollaron algoritmos para la estimación del espectro de los exponentes de Lyapunov para múltiples series temporales. Estos algoritmos se implementaron sobre la plataforma de tareas distribuidas T-arenal, logrando reducir considerablemente los tiempos de cómputo.
- Se definieron algoritmos de Minería de Datos basados en la técnica de los Árboles de Decisión, para interpretar los resultados de la clasificación de las múltiples series temporales según su estabilidad.
- Los algoritmos de la metodología propuesta fueron incorporados al software BioSyS lo que hará posible realizar el estudio de estabilidad en series temporales n-dimensionales.
- La metodología fue verificada con un caso de estudio de la literatura, obteniéndose resultados dentro de los mismos rangos de estabilidad e inestabilidad establecidos en el artículo utilizado.

Recomendaciones

- Adicionar otro algoritmo para el cálculo del Exponente de Lyapunov que resulte más eficiente computacionalmente.
- Incluir otra técnica que permita analizar la estabilidad en los casos en que no se cuente con el modelo del sistema, pero sí se tengan las series temporales.

Bibliografía

1. **Carbonell, Felix Miguel.** Tesis de doctorado: Cálculo numérico de exponentes de Lyapunov en ecuaciones diferenciales. Ciudad de la Habana : s.n., Enero de 2006.

2. **Michael T. Rosenstein, James J. Collins, Carlo J. De Luca.** A practical method for calculating largest Lyapunov. Boston University : s.n., 1992.

3. *Ponencia en Congreso Psicología Social. Dinámicas No Lineales: Algunas Técnicas de Análisis y Software Libre.* **Navarro, José.** Cádiz. España : s.n., 2007.

4. *Determining Lyapunov exponents from a time series.* **Alan Wolf, Jack B. Swift, Harry L. Swinney, John A. Vastano.** North-Holland, Amsterdam : Physica 16D, 1985, Vols. 16, pp. 285-317.

5. *Visualization of dynamical systems.* **Eduard Gröller, Helwig Löffelmann, Rainer Wegenkittl.** 1, The Netherlands : Elsevier Science Publishers B. V. Amsterdam, 1999, Future Generation Computer Systems, Vol. 15, págs. 75-86. 0167-739X .

6. *Sistemas Dinámicos Elementales.* **Rosales Ortega, José.** 2, Costa Rica : Revista Digital Matemática. Educación e Internet, 2006, Vol. 7. 1659 0643.

7. **Osorio, Karel.** Trabajo de diploma: Plataforma Computacional para el Desarrollo de la Biología de Sistemas. Ciudad de la Habana : s.n., 2004.

8. **Ricardo V. Solé, Susanna C. Manrubia.** *Orden y caos en sistemas complejos. Fundamentos.* Barcelona : Edicions UPC, 2001. 84-8301-430-0.

9. **González, María de las Nieves Lorenzo.** Tesis de doctorado: Influencia del ruido gaussiano correlacionado en la sincronización de sistemas caóticos . julio 2000.

10. *Modelos matemáticos de sistemas variables en el tiempo.* **Llavador, Fernando.** Valencia : s.n., 2005.

11. *A review of techniques for parameter sensitivity analysis of environmental models.* **Hamby, D.M.** 1994, Vols. 32, pp. 135-154.

12. **Rodríguez, Juan José.** Tesis de maestría: Un Estudio de Caso de Funciones Caóticas para el Análisis de la Encriptación de Imágenes Digitales. 2008.

13. **Aguirre, Armando.** *Introducción al tratamiento de series temporales. Aplicación a las ciencias de la Salud.* Madrid : Ediciones Díaz de Santos, 1994. 84-7978-153-X.
14. **Arellano, Mireya.** *Introducción al Análisis Clásico de Series de Tiempo.* [En línea] 2001. <http://www.5campus.com/leccion/seriest>.
15. *Utilización de redes neuronales en la caracterización, modelación y predicción de series temporales económicas en un entorno complejo.* **Elena Olmedo, Juan Manuel Valderas, Ruth Mateos, Ricardo Gimeno.** 23, s.l. : Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, 2004, Vol. 8, págs. 7-25. 1137-3601.
16. **Mouronte López, Mary Luz.** Tesis de doctorado: Caracterización y análisis de sistemas dinámicos no lineales mediante el estudio del mapa de frecuencias y espacio de fases. Madrid : s.n., 2004.
17. **M. García, A. Barreiro.** Análisis de la Estabilidad según Lyapunov de un Control Borroso en Tiempo Discreto. España : s.n.
18. **Lyapunov, A.M.** *Ph.D. Thesis: Problema General de la Estabilidad del movimiento.* s.l. : Karkov Mathematical Society, 1892.
19. *Implementación de un algoritmo para el cómputo de los exponentes de Lyapunov.* **Gabriel Agamennoni, Jorge L. Moiola.** s.l. : XII Reunión de Trabajo en Procesamiento de la Información y Control, 2007.
20. *El gráfico de recurrencia en el análisis de series temporales: el programa VRA (Visual Recurrence Analysis).* **López Jiménez, Ana María, y otros.** 1, pp.1-8, s.l. : Revista Electrónica de Metodología Aplicada, 2001, Vol. 6.
21. *Análisis fractal de series temporales.* **Sánchez Lozano, Miguel Angel.** 50, s.l. : Bolsa de Madrid, 1996.
22. *COPASI—a COmplex PAthway Simulator.* **Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jurgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, Ursula Kummer.** 24, pp. 3067–3074, s.l. : Bioinformatics, 2006, Vol. 22.
23. *Practical implementation of nonlinear time series methods: The TISEAN package.* **R. Hegger, H. Kantz, and T. Schreiber.** 2, pp. 413-435, s.l. : Chaos: An Interdisciplinary Journal of Nonlinear Science, 1999, Vol. 9.
24. **Arias, Gilberto.** Tesis de maestría: Editor de ecuaciones para la Plataforma de Simulación de Sistemas Biológicos. Ciudad de la Habana : s.n., 2008.

25. **The MathWorks.** MATLAB-The Language of Technical Computing. The MathWorks. [En línea] <http://www.mathworks.com/products/matlab/>.
26. **Tanenbaum A, Steen MV.** Distributed Systems: Principles and Paradigms. USA : Prentice Hall, Pearson Education , 2002.
27. **G Couloris JD, Kinberg T.** *Distributed Systems - Concepts and Design, 4th Edition.* s.l. : Addison-Wesley, Pearson Education, 2001.
28. **BOINC.** Berkeley Open Infrastructure for Network Computing. [En línea] National Science Foundation, 2009. <http://boinc.berkeley.edu>.
29. **Condor.** High Throughput Computing. [En línea] 2009. <http://www.cs.wisc.edu/condor/>.
30. **B.Sc, Thomas Keane.** Masters Thesis: A General-Purpose Heterogeneous Distributed Computing System. Department of Computer Science National University of Ireland Maynooth : s.n., 2004.
31. **CETA.** Centro Extremeño de Tecnologías Avanzadas. [En línea] 2006. <http://www.ceta-ciemat.es/es/?q=node/38>.
32. **SETI@home.** [En línea] 2009. <http://setiathome.berkeley.edu/index.php>.
33. *Tesis de maestría: Sistema de Cómputo Distribuido aplicado a la Bioinformática.* **Aguilera, Longendri.** Ciudad de la Habana : s.n., 2008.
34. *General-Purpose Heterogeneous Distributed Computing System.* **Keane, T. A.** National University of Ireland Maynooth : s.n., 2004.
35. **U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, P.Uthurusamy.** *Advances in knowledge discovery and data mining.* s.l. : AAAI/MIT Press, 1996.
36. **Escobar, Víctor Heughes.** Tesis de doctorado: Minería web de uso y perfiles de usuario: Aplicaciones con Lógica Difusa. 2007.
37. *Formas intermedias de representación en minería de texto. Memoria para el Diploma de Estudios Avanzados.* **Justicia, C.** Universidad de Granada : s.n., 2004.
38. **NetBeans.** NetBeans. [En línea] [Citado el: 20 de septiembre de 2009.] http://www.netbeans.org/index_es.html.
39. **Manuel Antolín Ayuso, Miguel Ángel Barcenilla Mancha.** Minería de Datos: Intrusiones de Red. [En línea] <http://www.it.uc3m.es/jvillena/irc/practicas/07-08/IntrusionesDeRed.pdf>.

40. **Sonia Collada, Rubén Gálvez.** Clasificación de e-mails: Detección de Spam. [En línea] <http://www.it.uc3m.es/jvillena/irc/practicass/07-08/DeteccionSpam.pdf>.

41. **Alonso, Carlos J.** *WEKA: Waitako Environment for Knowledge Analysis. Introducción básica.* Universidad de Valladolid : Departamento de Informática.

42. Minería de datos con Weka (ficheros ARFF). [En línea] 2007. <http://www.locualo.net/programacion/mineria-datos-weka-ficheros-arff/00000019.aspx>.

43. *Implementación de un Algoritmo para el Cómputo de los Exponentes de Lyapunov.* **Gabriel Agamennoni, Jorge L. Moiola.** s.l. : XII Reunión de Trabajo en Procesamiento de la Información y Control , 2007.

44. *Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems: A Method for Computing All of Them.* **G. Benettin, L. Galgani, A. Giorgilli and J.-M. Strelcyn.** 9, s.l. : Meccanica , 1980, Vol. 15 .

45. **Solís Angulo, Javier.** Tesis de maestría: Arquitectura de distribución adaptable para sistemas colaborativos implantados en la Web. México : s.n., 2009.

46. **SmartSoft.** SmartSoft. *Árboles de Decisión.* [En línea] 2009. <http://www.smartsoftint.com/esp/tecnologia/arboles-de-decision.html>.

47. **Julio Ortiz-Villajos, David Gutiérrez.** NPFINDER – Sistematización del Proceso de Reconocimiento de Sintagmas Nominales. [En línea] <http://www.esi.uem.es/~jmgomez/plenum/plenum4/06.pdf>.

48. **Quinlan.** *C4.5: Programs for Machine Learning Morgan Kaufmann .* 1993.

49. **Torres, Néstor V.** Caos en sistemas biológicos. Universidad de La Laguna, España : s.n.

50. *Solución numérica del atractor de Lorenz por el método de Runge Kutta-Fehlberg.* **Dosa Demartini, Melchor y Gómez Barria, Javier.** 1,2, s.l. : Revista de Investigación de Física, 2004, Vol. 7. 1605-7744.

51. **Monzó Marco, José.** Pensamiento Sistémico. Ideas y reflexiones sistémicas para un mundo complejo. 2008.

52. *Antigen-driven CD41 T cell and HIV-1 dynamics:Residual viral replication under highly active antiretroviral therapy.* **Ferguson, Neil M., y otros.** 1999.

53. *Mechanism and computer simulation of immune complex formation, opsonization, and clearance.* **Head, Martha, Maryhew, Nancy y Runquist, Olaf.** Minnesota : Journal Lab Clin Med, 1996, Vols. 128, pp. 61-74.

54. *Modeling the role of IL-2 in the interplay between CD4+ helper and regulatory T cells: Assessing general dynamical properties.* **García-Martínez, Karina y León, Kalet.** s.l. : Journal of Theoretical Biology, 2010.

55. *Estimación de los exponentes de Lyapunov.* **Luis Lara, Cesar Stoico, Rodrigo Machado, Mario Castagnino.** Argentina : Mecánica Computacional, 2003, Vol. 22.

56. **plots, Recurrence plots and cross recurrence.** Recurrence Plots and Cross Recurrence Plots. *Recurrence Plots At A Glance.* [En línea] 2009. <http://www.recurrence-plot.tk/glance.php>. 3-00-012347-4.

57. *The general problem of the stability of motion.* **Lyapunov, A.M.** s.l. : International Journal of control: Lyapunov centenary, 1992.

58. SETI@HOME. Detección de señales de otras civilizaciones . [En línea] 2003. <http://seti.astroseti.org/setiathome/>.

59. The Great Internet Mersenne Prime Search. [En línea] 2009. <http://www.mersenne.org/>.

60. **Ajpi, Jhony Oscar Cosme.** Modelo presa-depredador aplicando Matlab. [En línea] Universidad Franz Tamayo, 2009. <http://prueba66.blogspot.es/>.

61. **Christian Espinosa, Franco Parisi, Antonino Parisi.** Munich Personal RePEc Archive:Evidence of chaotic behavior in american stock markets. [En línea] November de 2007.

62. *Cuadernos de Economía, convergencia y estabilidad de los tipos de cambio europeos: Una aplicación de exponentes de Lyapunov.* **Elena Olmedo, Ricardo Gimeno, Lorenzo Escot, Ruth Mateos.** 2007, Vols. 44, pp. 91-108.

63. *Sistemas Dinámicos Elementales.* **Rosales Ortega, José .2,**Costa Rica: Revista Digital Matemática Educación e Internet Vol.7.1659 0643.

64. *Modelos matemáticos en Medicina y Biología. Bases teóricas y fundamentos.* **Campollo, Octavio.** 4, México : Revista Invest Clin. Imbiomed, 1994, Vols. 46, pp. 307-321.

65. **Marwan, Norbert.** Ph.D. Thesis: Encounters With Neighbours - Current Developments Of Concepts Based On Recurrence Plots And Their Applications. University of Potsdam : s.n. 3-00-012347-4.

66. **Miguel Ángel Fuente Rodríguez, Pablo Galarza Heredero.**
Práctica 1– Inteligencia en redes de comunicaciones. Predicción meteorológica.
Minería de Datos. [En línea] <http://www.it.uc3m.es/jvillena/irc/practicas/05-06/13mem.pdf>.

Glosario

- *JVM (Java Virtual Machine en inglés)*: forma parte de una plataforma llamada Ambiente de Ejecución de Java (JRE en sus siglas en inglés): contiene un conjunto de bibliotecas que además de la JVM son necesarias para ejecutar programas en Java.
- *Demonio*: programa residente que se ejecuta siempre que permanezca la aplicación que lo creó.
- *Overfitting*: son ciertas regularidades que se pueden encontrar en un conjunto de datos, pero sin tener sentido.
- *Serie estacionaria*: una serie es estacionaria cuando sus propiedades no varían a lo largo del tiempo.

Anexos

Anexo 1

En este anexo se muestra el código referente a cada paso dentro del algoritmo distribuido para el cálculo del espectro de los exponentes de Lyapunov.

Paso 1. Preparación de los datos en el servidor.

```
File f = new File(PROBLEMDIRECTORY, "sim.txt");  
ObjectInputStream in = new ObjectInputStream(new FileInputStream(f));  
cond = (ArrayList) in.readObject();  
in.close();
```

```
public void ReadProperties()  
    throws Throwable {  
    String parameter_list[] = {  
        "model.name", "metodo.name", "time.initial", "time.final", "eam", "erm", "step",  
"initial.granularity", "max.granularity", "idmodelo",  
        "servidorbd", "puertobd", "nombrebd", "usuariobd", "passbd", "gestorbd",  
"sistema"  
    };  
  
    BufferedReader inputs = new BufferedReader(new FileReader(new  
File(PROBLEMDIRECTORY, "input.txt")));  
    properties = new Properties();  
    for (String line = inputs.readLine(); line != null; line = inputs.readLine()) {  
        if (line.length() <= 0 || line.charAt(0) == '#') {  
            continue;  
        }  
        StringTokenizer stk = new StringTokenizer(line, "=");  
        if (stk.countTokens() == 2) {  
            String lhs = stk.nextToken().trim().toLowerCase();  
            String rhs = stk.nextToken().trim();  
            properties.setProperty(lhs, rhs);  
        }  
    }  
    inputs.close();
```

```
}
```

Inicializar el fichero .arff

```
private void inicializarArff() throws FileNotFoundException {  
    arffFile = new PrintStream(new FileOutputStream(new  
File(PROBLEMDIRECTORY, "arffFile.arff")));  
    arffFile.print("@relation aa_clustered\n\n@attribute Instance_number numeric\n");  
    posiciones = new ArrayList<Integer>();  
    simbolos = new ArrayList<String>();  
    int cantv = 0;  
    for (int i = 0; i < cond.size(); i++) {  
        if (cond.get(i) instanceof BSVariable) {  
            cantv++;  
            simbolos.add(cond.get(i).getSimbolo());  
        }  
        if (cond.get(i) instanceof BSParametro && cond.get(i).getValores().size() > 1) {  
            posiciones.add(i - cantv);  
            arffFile.print("@attribute " + cond.get(i).getSimbolo() + " numeric\n");  
        }  
    }  
    arffFile.print("@attribute Cluster {estable, inestable}\n\n@data\n");  
}
```

Paso 2. Distribución de las tareas.

```
public Vector generateWorkUnit(ClientInfo clientInfo)  
    throws Throwable {  
    String ip = clientInfo.getIP();  
    Vector output = new Vector();  
    Vector v = isAllowed(ip);  
    if (v != null && !isFallet(ip)) {  
        if (v.size() == 0) {  
            v = comandosEncontrados(ip);  
        }  
        if (simulacionesFallidas.size() != 0) {  
            output.add(ip);  
        }  
    }  
}
```

```

        output.add(properties);

        output.add(v);

        output.add(simulacionesFallidas.get(0));

        simulacionesFallidas.remove(0);

        delivery.println((new      StringBuilder()).append(((ArrayList)      ((ArrayList)
output.get(3)).get(0)).size()).append("\t").append(ip).toString());

        pendingUnits++;

        output.add(posiciones);

        output.add(simbolos);

        return output;
    }

    if (cont == cantsim) {

        return null;

    } else {

        delivery.println((new
StringBuilder()).append(granularity).append("\t").append(ip).toString());

        pendingUnits++;

        output.add(ip);

        output.add(properties);

        output.add(v);

        output.add(generarSimulaciones());

        output.add(posiciones);

        output.add(simbolos);

        return output;

    }

    } else {

        delivery.println((new      StringBuilder()).append("No      enviado      a:
\t").append(ip).toString());

        return null;

    }

    }
}

```

Paso 3. Procesamiento de las tareas.

```

public Vector processUnit(Vector workUnit)
    throws Throwable {

```

```

ArrayList<String> instancias = new ArrayList<String>();
ArrayList r = (ArrayList) workUnit.get(3);
Vector v = new Vector();
ArrayList<Integer> posiciones = (ArrayList) workUnit.get(4);
simbolos = (ArrayList<String>) workUnit.get(5);

PrintStream p = new PrintStream(new FileOutputStream(new
File(PROBLEMDIRECTORY, "Prueba.txt"), true), true);

properties = (Properties) workUnit.get(1);
downloadFile("BSToolbox.zip");
try {
    File compactado = getFile("BSToolbox.zip");

    File RESOURCE_FOLDER = new
File(PROBLEMDIRECTORY.getParentFile().getParentFile(), "resources");

    if (!RESOURCE_FOLDER.exists()) {
        boolean flag = RESOURCE_FOLDER.mkdirs();
        if (!flag) {
            throw new Exception("Unable to create resource folder");
        }
    }

    BSTOOLBOX_FOLDER = new File(RESOURCE_FOLDER, "BSToolbox");
    if (!BSTOOLBOX_FOLDER.exists()) {
        boolean flag = BSTOOLBOX_FOLDER.mkdirs();
        if (!flag) {
            throw new Exception("Unable to create BSToolbox folder");
        }
    }

    descompactar2(compactado.getAbsolutePath(),
BSTOOLBOX_FOLDER.getAbsolutePath());
} catch (Exception ex) {
    error = ex.getMessage();
    v.add(Integer.valueOf(-1));
    v.add(workUnit.get(0));
    v.add(ex.getMessage());
    v.add(r);
}

```

```

    p.close();
    return v;
}
crearMatlabModel();
try {
    Matlab m;
    if (System.getProperty("os.name").toLowerCase().contains("linux")) {
        if (((Vector) workUnit.get(2)).size() == 1) {
            m = new Matlab((String) ((Vector) workUnit.get(2)).get(0));
        } else {
            m = new Matlab(BSSimulacion.findTool("matlab"));
        }
    } else {
        m = new Matlab("matlab");
    }
    m.Ejecutar();
    m.eval("cd(" + BSTOOLBOX_FOLDER.getAbsolutePath() + ")");
    for (int i = 0; i < ((ArrayList) r.get(0)).size(); i++) {
        ArrayList vpopsim = (ArrayList) ((ArrayList) r.get(0)).get(i);
        String var = "[";
        for (int j = 0; j < vpopsim.size(); j++) {
            var += vpopsim.get(j);
            if (j != (vpopsim.size() - 1)) {
                var += ",";
            }
        }
        var += "]";
        ArrayList<Double> vparsim = (ArrayList) ((ArrayList) r.get(1)).get(i);
        String par = "[";
        for (int j = 0; j < vparsim.size(); j++) {
            par += vparsim.get(j);
            if (j != (vparsim.size() - 1)) {
                par += ",";
            }
        }
    }
}

```

```

    }
}
par += "]";
p.println(i + " calculando...");
m.eval("[T,Lexp] = BSlyapunov(" + vpopsim.size() +
",@Lorentz,@"+properties.getProperty("metodo.name")+"," + ((ArrayList) ((ArrayList)
r.get(3)).get(i)).get(0) + "," + properties.getProperty("step") + "," + ((ArrayList) ((ArrayList)
r.get(3)).get(i)).get(1) + "," + var + ",0," + par + ",[]);");
m.eval("x = max(max(Lexp));");
double val = (Double) m.save("x").getResultado(0, 0);
p.println(i + " termino...");
String instancia = "";
instancia += "," + Double.toString(vparsim.get(posiciones.get(0)));
for (int j = 1; j < posiciones.size(); j++) {
    instancia += "," + Double.toString(vparsim.get(posiciones.get(j)));
}
String estabilidad = "inestable";
if (val <= 0) {
    estabilidad = "estable";
}
instancia += "," + estabilidad + "\n";
p.println(i + instancia);
instancias.add(instancia);
}
v.add(Integer.valueOf(1));
v.add(workUnit.get(0));
m.Close();
if (m != null) {
    v.add(m.getCmd());
}
v.add(instancias);
} catch (Exception ex) {
    error = ex.getMessage();
    v.add(Integer.valueOf(-1));
}

```

```

        v.add(workUnit.get(0));

        v.add(ex.getMessage());

        v.add(r);

        p.close();

        return v;
    }

    p.close();

    return v;
}

private void crearMatlabModel() throws Exception {

    bsDao =
    BSDAOFactory.getDAOFactory(Integer.parseInt(properties.getProperty("gestorbd")),
    properties.getProperty("servidorbd"), properties.getProperty("puertobd"),
    properties.getProperty("nombredb"), properties.getProperty("usuariobd"),
    properties.getProperty("passbd"));

    persistentes.Modelo modelo =
    bsDao.getModeloDAO().getModelo(Integer.parseInt(properties.getProperty("idmodelo")));

    int cantv = modelo.getVariables().size();

    File Lorentz = new File(BSTOOLBOX_FOLDER, "Lorentz.m");

    BufferedWriter lo = new BufferedWriter(new FileWriter(Lorentz));

    lo.write("function r_r=Lorentz(varargin)\n\n" + "r_r=zeros(" + cantv * (cantv + 1) +
    ",1);\n");

    int posi = modelo.getMatlabmodel().indexOf("y_y");
    int posf = modelo.getMatlabmodel().indexOf("if") - 1;

    lo.write("\n" + modelo.getMatlabmodel().substring(posi, posf) + "\n" + "Y = zeros("
    + cantv + ");\n" + "n = " + (cantv + 1) + ";\nfor i = 1:" + cantv + "\n" + "for j = 1:" + cantv + "\n" +
    "Y(j,i) = y_y(n);\n" + "n=n+1;\n" + "end\n" + "end\n");

    String s = modelo.getMatlabmodel().substring(posi, posf);

    for (int i = 0; i < simbolos.size(); i++) {

        lo.write("syms " + simbolos.get(i) + ";\n");

        while (s.indexOf("y_y(" + (i + 1) + ")") != -1) {

            String s1 = s.substring(0, s.indexOf("y_y(" + (i + 1) + ")"));

            String s2 = s.substring(s.indexOf("y_y(" + (i + 1) + ")") + ("y_y(" + (i + 1) +
            ")").length(), s.length());

            s = s1 + simbolos.get(i) + s2;

        }

        while (s.indexOf("r_r(" + (i + 1) + ")") != -1) {

```

```

        String s1 = s.substring(0, s.indexOf("r_r(" + (i + 1) + ")"));
        String s2 = s.substring(s.indexOf("r_r(" + (i + 1) + ")") + ("r_r(" + (i + 1) +
        ")").length(), s.length());
        s = s1 + "g_g(" + (i + 1) + ")" + s2;
    }
}
lo.write(s + "\n");
lo.write("for i=1:" + cantv + "\n");
for (int i = 0; i < simbolos.size(); i++) {
    lo.write("J(i," + (i + 1) + ") = diff(g_g(i)," + simbolos.get(i) + ");\n");
}
lo.write("end\n");
for (int i = 0; i < simbolos.size(); i++) {
    lo.write(simbolos.get(i) + " = y_y(" + (i + 1) + ");\n");
}
lo.write("J = eval(J);\n" + "%Variational equation  \n" + "r_r(" + (cantv + 1) + "):" +
(cantv * (cantv + 1) + ")=J*Y;\n");
lo.close();
}

```

Paso 4. Recepción de los resultados del análisis de estabilidad.

```

public boolean processResults(Long aLong, Vector vector)
    throws Throwable {
    pendingUnits--;
    if (((Integer) vector.get(0)).intValue() == 1) {
        pcCorrectas.println(vector.get(1));
        if (vector.size() == 4) {
            Vector v = isAllowed((String) vector.get(1));
            if (v != null && v.size() == 0) {
                v = comandosEncontrados((String) vector.get(1));
                if (v.size() == 0) {
                    comandosEncontrados.println((new
                    StringBuilder()).append(vector.get(1)).append("\t").append(vector.get(2)).toString());
                }
            }
        }
    }
}

```

```

    }
    ArrayList<String> instancias = (ArrayList<String>) vector.get(vector.size() - 1);
    for (int i = 0; i < instancias.size(); i++) {
        arffFile.print(contadorInstancia + instancias.get(i));
        contadorInstancia++;
    }
    if (cont == cantsim && pendingUnits == 0 && simulacionesFallidas.size() == 0) {
        resultados.println("Registrando resultado: Se ha cumplido la condicion");
        resultados.close();
        arffFile.close();
        return true;
    }
} else if (((Integer) vector.get(0)).intValue() == -1) {
    pcFallida.println((new
StringBuilder()).append(vector.get(1)).append("\t").append(vector.get(2)).toString());
    simulacionesFallidas.add((ArrayList) vector.get(3));
}
return false;
}

```

Anexo 2

En este anexo se muestra el contenido de un fichero de tipo *.arff generado con los resultados del análisis de estabilidad, es a partir de este fichero que se puede realizar una interpretación de los resultados obtenidos.

```
@relation aa_clustered

@attribute Instance_number numeric
@attribute b numeric
@attribute c numeric
@attribute Cluster {estable, inestable}

@data
0,2.0,3.0,inestable
1,2.0,3.7,inestable
2,2.0,4.4,inestable
3,2.0,5.1000000000000005,inestable
4,2.0,5.8000000000000001,inestable
5,2.0,6.5000000000000001,inestable
6,2.0,7.2000000000000001,inestable
7,2.0,7.9000000000000001,inestable
8,2.0,8.6000000000000001,inestable
9,2.0,9.3,inestable
10,2.0,10.0,inestable
11,2.8,3.0,inestable
12,2.8,3.7,inestable
13,2.8,4.4,inestable
14,2.8,5.1000000000000005,inestable
15,2.8,5.8000000000000001,inestable
16,2.8,6.5000000000000001,inestable
17,2.8,7.2000000000000001,inestable
18,2.8,7.9000000000000001,inestable
19,2.8,8.6000000000000001,inestable
20,2.8,9.3,inestable
21,2.8,10.0,inestable
22,3.5999999999999996,3.0,inestable
23,3.5999999999999996,3.7,inestable
24,3.5999999999999996,4.4,inestable
25,3.5999999999999996,5.1000000000000005,inestable
26,3.5999999999999996,5.8000000000000001,inestable
27,3.5999999999999996,6.5000000000000001,inestable
28,3.5999999999999996,7.2000000000000001,inestable
29,3.5999999999999996,7.9000000000000001,inestable
30,3.5999999999999996,8.6000000000000001,inestable
31,3.5999999999999996,9.3,inestable
32,3.5999999999999996,10.0,inestable
33,4.3999999999999995,3.0,estable
34,4.3999999999999995,3.7,estable
35,4.3999999999999995,4.4,inestable
36,4.3999999999999995,5.1000000000000005,inestable
37,4.3999999999999995,5.8000000000000001,inestable
38,4.3999999999999995,6.5000000000000001,inestable
39,4.3999999999999995,7.2000000000000001,inestable
40,4.3999999999999995,7.9000000000000001,inestable
41,4.3999999999999995,8.6000000000000001,inestable
42,4.3999999999999995,9.3,inestable
43,4.3999999999999995,10.0,inestable
44,5.199999999999999,3.0,estable
45,5.199999999999999,3.7,estable
```

46,5.1999999999999999,4.4,instable
47,5.1999999999999999,5.10000000000000005,instable
48,5.1999999999999999,5.8000000000000001,instable
49,5.1999999999999999,6.5000000000000001,instable
50,5.1999999999999999,7.2000000000000001,instable
51,5.1999999999999999,7.9000000000000001,instable
52,5.1999999999999999,8.6000000000000001,instable
53,5.1999999999999999,9.3,instable
54,5.1999999999999999,10.0,instable
55,5.9999999999999999,3.0,estable
56,5.9999999999999999,3.7,estable
57,5.9999999999999999,4.4,estable
58,5.9999999999999999,5.10000000000000005,instable
59,5.9999999999999999,5.8000000000000001,instable
60,5.9999999999999999,6.5000000000000001,instable
61,5.9999999999999999,7.2000000000000001,instable
62,5.9999999999999999,7.9000000000000001,instable
63,5.9999999999999999,8.6000000000000001,instable
64,5.9999999999999999,9.3,instable
65,5.9999999999999999,10.0,instable

Anexo 3

En este anexo se muestra el contenido del fichero de tipo *.arff generado con los valores del parámetro r que fue variado y las correspondientes clasificaciones para estos valores.

```
@relation aa_clustered

@attribute Instance_number numeric
@attribute r numeric
@attribute Cluster {estable, inestable}

@data
0,20.0,estable
1,21.0,estable
2,22.0,estable
3,23.0,estable
4,24.0,inestable
5,25.0,inestable
6,26.0,inestable
7,27.0,inestable
8,28.0,inestable
9,29.0,inestable
10,30.0,inestable
```