

# DESARROLLO DE UNA APLICACIÓN WEB PARA LA GRAFICACIÓN DE ATRACTORES CAÓTICOS UTILIZANDO LA METODOLOGÍA SCRUM

## Development of a Web Application for Graphing Chaotic Attractors Using the SCRUM Methodology

Thalía Zárate Mora <sup>1*</sup>	talia.zarate@esPOCH.edu.ec
Raúl Rosero <sup>1, †</sup>	raul.rosero@esPOCH.edu.ec
Danilo Pastor <sup>1, ‡</sup>	danilo.pastor@esPOCH.edu.ec
Maricela Jiménez Rodríguez <sup>2, ¶</sup>	maricelajrodriguez@academicos.udg.mx
Omar S. Gómez	ogomez@esPOCH.edu.ec

<sup>1</sup> Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo (ESPOCH), 060155, Riobamba, Ecuador,

<sup>2</sup> Profesora-Investigadora en el Centro Universitario de la Ciénega, Universidad de Guadalajara

<sup>3</sup> Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador,

### RESUMEN

El trabajo se enfoca en optimizar el proceso actual de la graficación de atractores en el Grupo de Investigación en Ingeniería de Software (GrIISoft) de la FIE-ESPOCH mediante el desarrollo de una aplicación web que automatiza dicho procedimiento con la metodología Scrum. La aplicación abarca atractores como Lorenz, Rossler, Sprott y Chen, incorporando funciones como gestión de usuarios y visualización de gráficos. A través de cuatro sprints, se implementaron 17 historias de usuarios y 8 historias técnicas. La evaluación comparativa entre el procedimiento actual de GrIISoft y la aplicación web reveló una reducción significativa del 98.9% en el tiempo de respuesta promedio, disminuyendo de 10501,67 ms a 116,57 ms. La aplicación demostró eficiencia en el uso de memoria y CPU en navegadores como Edge y Opera, con consumos más bajos, y exhibió la capacidad de manejar hasta 207 usuarios simultáneamente. Los resultados respaldan la mejora en los tiempos de respuesta.

**Palabras Clave:** Desarrollo de Aplicaciones Web, Metodología SCRUM, Atractores Caóticos, Modelo Vista Controlador, Norma ISO/IEC 25010, Eficiencia.

### ABSTRACT

The work focuses on optimizing the current process of plotting attractors at the Software Engineering Research Group (GrIISoft) of FIE-ESPOCH by developing a web application that automates this procedure using the Scrum methodology. The application covers attractors such as Lorenz, Rossler, Sprott, and Chen, incorporating features such as user management and graph visualization. Through four sprints, 17 user stories and 8 technical stories were implemented. Comparative evaluation between GrIISoft's current procedure and the web application revealed a significant 98.9% reduction in average response time, decreasing from 10501.67 ms to 116.57 ms. The application demonstrated efficiency in memory and CPU usage on browsers like Edge and Opera, with lower consumptions, and exhibited the ability to handle up to 207 users simultaneously. The results support the improvement in response times.

**Palabras Clave:** Web Application Development, SCRUM Methodology, Chaotic Attractors, Model-View-Controller, ISO/IEC 25010 Standard, Efficiency.

## ► I. Introducción

En el ámbito de la visualización y representación gráfica de sistemas caóticos, el desarrollo de aplicaciones web modernas ha ganado relevancia como medio para mejorar la accesibilidad y eficiencia en la generación de atractores caóticos. El presente trabajo se centra en la creación de una aplicación web destinada a tal fin, específicamente dirigida al Grupo de Investigación en Ingeniería de Software (GrIISof), perteneciente a la FIE-ESPOCH. Este grupo se encuentra desarrollando el proyecto titulado: Enfoque de cifrado de objetos JSON utilizando sincronización caótica a partir de análisis de un conjunto de atractores (IDIPI-283), en donde uno de los procesos necesarios es la graficación de atractores caóticos. El propósito principal de este proyecto es abordar la necesidad de optimizar y agilizar el proceso actual de generación de atractores, incorporando la metodología ágil Scrum para su desarrollo.

La aplicación web desarrollada se estructura en módulos clave que abarcan desde la gestión de usuarios hasta la visualización y consulta de gráficos. Para asegurar su viabilidad y eficiencia, se realizó un estudio de los atractores caóticos utilizados por GrIISof, enfocándose en los atractores Lorenz, Rossler, Sprott y Chen. Se describe el proceso de desarrollo, destacando la metodología Scrum como enfoque de gestión que permite iteraciones regulares y una comunicación eficiente entre el equipo de desarrollo y los interesados.

Para evaluar el rendimiento de la aplicación web desarrollada se compararon los tiempos de respuesta promedio con los del actual proceso de graficación de atractores de GrIISof. Los resultados revelan una mejora significativa en términos de velocidad de respuesta, lo que se traduce en una experiencia de usuario más fluida y efectiva. Además, se llevó a cabo un análisis del consumo de recursos de la aplicación web en distintos navegadores, lo que evidenció su eficacia en términos de utilización de memoria RAM, CPU y GPU. Asimismo, se evaluó la capacidad de la aplicación para gestionar múltiples usuarios simultáneos, demostrando de manera concluyente

su escalabilidad y adaptabilidad. Estos hallazgos contribuyen a la comprensión integral del rendimiento de la aplicación en comparación con el proceso existente de graficación de GrIISof, subrayando sus beneficios tanto en términos de velocidad como de eficiencia en el manejo de recursos.

El análisis estadístico ANOVA aplicado a los tiempos de respuesta respalda las mejoras observadas, validando la significativa diferencia entre los períodos pretest y postest. En conjunto, este trabajo presenta una solución efectiva para la generación de atractores caóticos a través de una aplicación web, incorporando los principios ágiles de Scrum para su desarrollo y garantizando mejoras notables en términos de rendimiento y eficiencia en comparación con el proceso previo.

## ► II. Marco teórico

### A. Caos.

El caos se refiere a un estado de desorden absoluto en sistemas dinámicos, donde la predicción precisa de su comportamiento a largo plazo se vuelve imposible debido a la sensibilidad extrema a pequeñas alteraciones iniciales, conocido como el "efecto mariposa". Aunque estos sistemas muestran regularidades a corto plazo, su comportamiento se vuelve cada vez más impredecible a medida que pasa el tiempo, debido a la amplificación de errores iniciales. La teoría del caos surge para comprender este fenómeno, pero se destaca que, a pesar de utilizar características estadísticas para hacer predicciones, las leyes fundamentales de la física siguen siendo aplicables [1].

### B. Sistemas dinámicos

Los sistemas dinámicos, son modelos matemáticos que intentan prever el comportamiento de sistemas físicos a lo largo del tiempo. Estos modelos emplean ecuaciones diferenciales o en diferencias finitas según la naturaleza continua o discreta del sistema. Cuando un sistema no cumple con el principio de superposición, se considera no lineal, lo que conduce a un estudio cualitativo para comprender su dinámica, con la posibilidad

de comportamiento caótico. Se clasifican en sistemas discretos y continuos, donde los sistemas lineales muestran cambios proporcionales a las variaciones iniciales, mientras que los no lineales pueden generar cambios significativos [2].

En los sistemas lineales, la incertidumbre se mantiene constante, mientras que en los no lineales puede variar en el tiempo. Además, se destaca la diferencia en la resolución analítica entre sistemas lineales y no lineales, siendo estos últimos más complejos y con posibles propiedades de "mezclado topográfico", lo que dificulta la predicción a largo plazo [1], [3].

### 1) Sistema caótico

Los sistemas caóticos, son sistemas dinámicos especialmente sensibles a las condiciones iniciales. Incluso pequeñas variaciones en estas condiciones pueden llevar a resultados radicalmente diferentes con el tiempo, lo que hace difícil predecir su comportamiento a largo plazo. Estos sistemas muestran sensibilidad exponencial a las condiciones iniciales, lo que implica que pequeños cambios se amplifican enormemente a medida que el sistema evoluciona [4].

El matemático y meteorólogo Edward Lorenz fue pionero en el estudio de sistemas caóticos al descubrir la gran sensibilidad de modelos matemáticos simples de la atmósfera a las condiciones iniciales. Esto llevó al descubrimiento de atractores extraños en sistemas caóticos, conjuntos de puntos que el sistema visita repetidamente, pero sin repetirse exactamente en el mismo orden, y que tienen una estructura fractal, mostrando complejidad infinita a cualquier escala de observación [4], [5].

La teoría del caos y el estudio de sistemas caóticos tienen aplicaciones en campos como física, biología, economía y meteorología. Estos sistemas permiten comprender fenómenos complejos y no lineales que escapan a los modelos deterministas tradicionales. Además, el caos se aplica en áreas como criptografía y generación de números aleatorios [6].

### a) Ventajas y desventajas del sistema caótico

**Tabla I**  
Ventajas y desventajas del sistema caóticos

VENTAJAS	DESVENTAJAS
1. Capacidad para generar patrones complejos.	1. Falta de predicción precisa del comportamiento futuro.
2. Comportamiento no lineal y no predecible.	2. Dificultad para encontrar soluciones analíticas.
3. Adaptabilidad y capacidad de respuesta al cambio.	3. Dificultad para reproducir resultados.
4. Potencial para la exploración de nuevas soluciones.	4. Sensibilidad a pequeñas perturbaciones.

### 2) Atractor caótico

Un atractor se define como un conjunto de estados hacia los cuales un sistema tiende a evolucionar, abarcando diversas condiciones iniciales. Estos valores permanecen cercanos incluso ante pequeñas alteraciones. En sistemas de dimensiones limitadas, el atractor constituye una región en el espacio n-dimensional, siendo n el número de dimensiones, que pueden representar, por ejemplo, coordenadas de posición en entidades físicas o variables separadas en sistemas económicos [7].

La representación geométrica del atractor puede ser bidimensional o tridimensional, dependiendo de la variable en evolución. Puede tomar diversas formas, como un punto, conjunto finito de puntos, curva, variedad o incluso un conjunto complejo con estructura fractal, conocido como atractor extraño. En sistemas dinámicos caóticos, la descripción de los atractores ha sido un logro fundamental de la teoría del caos. Una trayectoria en el atractor no requiere restricciones particulares, excepto permanecer dentro del atractor con el tiempo, pudiendo ser periódica o caótica. Si un grupo de puntos muestra una pauta, pero se aleja con el tiempo, se denomina repelente, no un atractor [8].

El atractor es la representación geométrica de cómo un sistema evoluciona en el tiempo, identificable por su número de dimensiones. Una dimensión 0 indica un sistema estático, dimensión 1 denota sistema periódico, mientras que dimensión 2

o superior sugiere sistema cuasi-periódico. Un ejemplo es el péndulo oscilante, cuyo atractor puede guiar en oscilaciones regulares, pero variar debido a otros factores, resultando en trayectorias irregulares. La cuenca de un atractor es el conjunto de condiciones iniciales para un comportamiento determinado, y las trayectorias pueden ser periódicas, cuasi periódicas o caóticas [9].

- Atractor extraño

En el ámbito de los sistemas dinámicos no lineales y caóticos, los atractores extraños son regiones en el espacio de fases hacia las cuales convergen las dinámicas de sistemas que muestran comportamiento caótico. Estos atractores, representados por trayectorias curvas, describen la evolución de sistemas sometidos a movimientos caóticos. El atractor de Lorenz es un ejemplo clave de un atractor extraño tridimensional, reconocido por su complejidad y singularidad, adoptando una forma peculiar similar a las alas de una mariposa según un conjunto de ecuaciones diferenciales no lineales [10].

Estos atractores encapsulan dos características esenciales de los sistemas caóticos: determinismo e impredecibilidad. Geométricamente, los atractores extraños se caracterizan por tener una dimensión fractal, lo que refleja su compleja estructura y comportamiento no lineal. Su relevancia radica en su capacidad para ilustrar y comprender la dinámica de sistemas complejos que exhiben comportamientos caóticos impredecibles pero deterministas [11].

### 3) *Atractor de Lorenz*

El concepto del atractor de Lorenz, presentado por Lorenz en 1963, revolucionó la comprensión de los sistemas dinámicos no lineales. Lorenz demostró que las predicciones lineales para sistemas atmosféricos en miniatura no eran ni periódicas ni asintóticamente periódicas. Este descubrimiento reveló un comportamiento caótico en ciertos valores de parámetros, generando una complejidad extrema conocida como el atractor de Lorenz [4]. Este atractor consiste en un conjunto de puntos en el espacio de estado que siguen trayectorias

caóticas en un sistema dinámico descrito por las ecuaciones de Lorenz. Se puede observar en la Figura 1. su forma en "mariposa" se repite a diversas escalas y su sensibilidad a las condiciones iniciales implica que cambios mínimos en estas condiciones pueden provocar comportamientos radicalmente diferentes en el sistema a largo plazo. Este fenómeno desafía las predicciones lineales y destaca la naturaleza impredecible y compleja de los sistemas no lineales y caóticos [12].

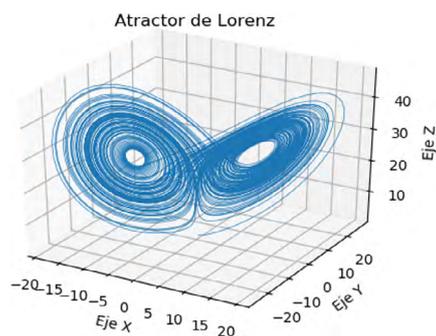


Fig. 1. *Atractor de Lorenz*

### 4) *Atractor de Rossler*

El artículo "An Equation for Continuous Chaos"[13] introduce el atractor de Rössler, un sistema dinámico no lineal definido por tres ecuaciones diferenciales ordinarias. Aunque tiene similitudes con el atractor de Lorenz, es más simple y muestra una estructura en forma de toro en su topología como se observa en la Figura 2. Este atractor exhibe dinámicas caóticas, caracterizadas por propiedades fractales, lo que significa que pequeñas variaciones en las condiciones iniciales pueden generar resultados radicalmente diferentes. Su comportamiento caótico lo hace sensible a las condiciones iniciales, lo que lo convierte en un fenómeno fascinante para el estudio de sistemas dinámicos y fractales [14].

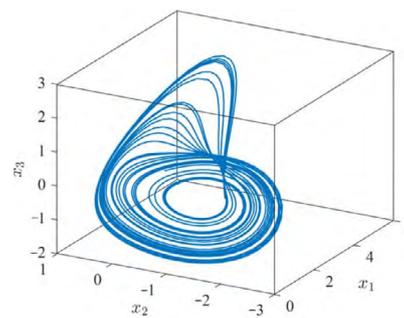


Fig. 2. *Atractor de Rossler*

### 5) *Atractor de Chen*

El descubrimiento de un atractor caótico por Guanrong Chen en 1999, que tiene similitudes, pero no es topológicamente equivalente al conocido Atractor de Lorenz [15]. Este nuevo atractor se encuentra en un sistema tridimensional autónomo de naturaleza simple. El atractor de Chen, también conocido como atractor de doble desplazamiento, presenta una estructura geométrica particular caracterizada por la presencia de un número infinito de capas fractales como se observa en la Figura 3. Cada sección del atractor muestra propiedades fractales a diferentes escalas, lo que implica que su complejidad se repite al observarlo con distintos niveles de magnificación. Este descubrimiento amplía la comprensión de los atractores caóticos al introducir un nuevo tipo con propiedades únicas y un comportamiento caótico distinto al Atractor de Lorenz [16].

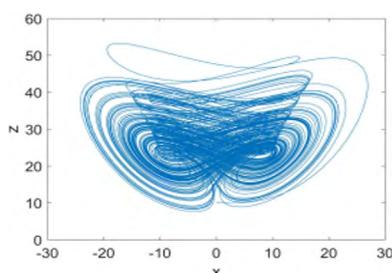


Fig. 3. *Atractor de Chen*

### 6) *Atractor de Sprott*

El atractor caótico de Sprott, es un fenómeno que muestra un comportamiento caótico a pesar de no tener puntos de equilibrio [17]. Se describe mediante una sola ecuación diferencial ordinaria de tercer orden y carece de parámetros específicos. En lugar de proporcionar condiciones iniciales, se presentan los exponentes de Lyapunov, que permiten obtener estas condiciones a partir de ellos [18]. Se puede observar la representación gráfica del atractor de Sprott en la Figura 4.

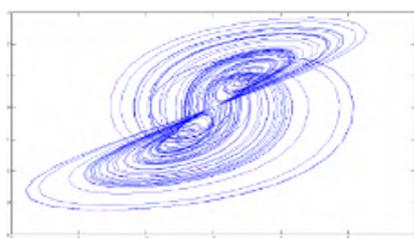


Fig. 4. *Atractor de Sprott*

## ► III. Marco metodológico

### A. *Aplicación de la Metodología SCRUM*

En este apartado se mencionan las fases de desarrollo de la aplicación web para la graficación de atractores caóticos utilizando la metodología ágil SCRUM y la metodología de evaluación de la eficiencia de acuerdo con la norma ISO/IEC 25010.

#### 1) *Product Backlog*

El product backlog registra todos los requisitos definidos por el propietario del producto. Estos requisitos se recopilan a través de entrevistas llevadas a cabo durante sesiones con el investigador Omar Gómez. Este se compone de historias de usuario y técnicas detalladas, fundamentales para la elaboración y desarrollo de la aplicación en cuestión.

#### 2) *Sprint Backlog*

El sprint backlog se emplea para gestionar el progreso del desarrollo de software de manera eficiente. Su finalidad principal radica en garantizar el logro de los objetivos planteados. En el contexto específico de esta aplicación, se han ejecutado cuatro Sprints, cada uno con una duración de dos semanas con 20 puntos de esfuerzo en cada uno. Al finalizar cada sprint el equipo de desarrollo evaluó el cumplimiento de tareas de cada sprint, tomando en cuenta los puntos de estimación designados inicialmente para cada sprint.

#### 3) *Desarrollo de la aplicación web de graficación de atractores*

Previamente a la presentación del Burndown Chart, resulta esencial destacar el desarrollo de la aplicación web dedicada a la graficación de atractores. Desarrollada como parte integral de este trabajo, dicha aplicación se ha diseñado no solo con el propósito de mejorar la eficiencia del proceso actual de graficación de atractores de GrIISof si no para facilitar la visualización y análisis de los atractores en el contexto de la investigación. La aplicación incluye módulos claves para la gestión de usuarios, autenticación,

almacenamiento y consulta de información mejorando potencialmente la experiencia de los investigadores que interactúen con la aplicación

El código fuente de la aplicación está disponible en el siguiente enlace <https://github.com/Thaferzzitha/griisoft-web-project>. Este enlace proporciona acceso al código fuente completo, permitiendo una revisión detallada del proyecto y la posibilidad de una futura contribución al desarrollo de mejoras en esta aplicación.

En la Figura 5. Podemos visualizar la interfaz del módulo de Graficación de atractores de la aplicación web desarrollada.

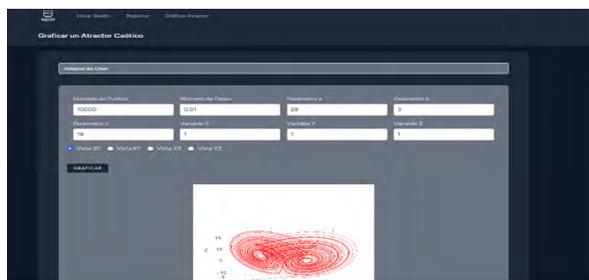


Fig. 5. Interfaz del módulo de graficación de la aplicación web

#### 4) *Burndown Chart*

La metodología ágil SCRUM recomienda el burndown chart como una herramienta gráfica para la gestión del proyecto, en la cual se muestra el progreso del proyecto a lo largo de los sprints, revelando la velocidad con la que se cumplieron. En la Figura 6. se observa el burndown chart resultante del presente proyecto, en el eje X se muestran los 4 sprints, y en el eje Y se muestran los puntos estimados de esfuerzo. Los puntos estimados de esfuerzo se representan en la línea azul y los puntos reales de esfuerzo se representan en la línea naranja.

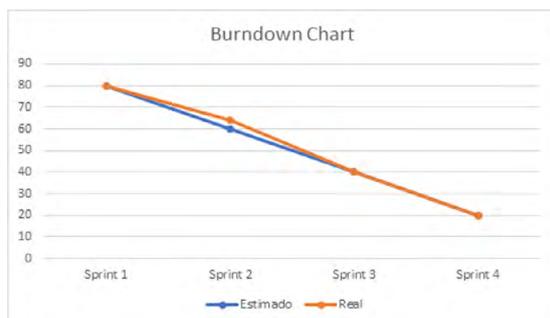


Fig. 6. Burndown Chart del Desarrollo de la aplicación web

Los sprints 1, 3 y 4 se desarrollaron a la par de los puntos estimados, sin embargo, en el sprint 2 existe un desfase respectivamente de los puntos estimados, esto debido a que en este sprint se desarrolló la funcionalidad de graficación de atractores caóticos, considerado por el equipo de desarrollo como una funcionalidad de dificultad alta.

#### B. Método utilizado para la evaluación de la Eficiencia en función a la norma la ISO/IEC 25010

Para poder evaluar la eficiencia de desempeño según la norma ISO/IEC 25010 se utilizó el método de observación, esto debido a que se observa el comportamiento, las acciones y eventos, en entornos naturales y controlados, de la aplicación web de graficación de atractores caóticos versus el proceso actual de graficación de atractores caóticos, para poder registrar los datos arrojados y analizarlos posteriormente.

##### 1) Población

Se emplearon enfoques específicos para medir las tres subcaracterísticas de la norma ISO/IEC 25010, por lo tanto, cada subcaracterística tiene una población propia descrita a continuación.

- Cinco participantes con experiencia en sistemas de visualización web y ejecución de scripts Python, quienes interactuaron con la aplicación web de graficación de atractores caóticos versus el script de Python de graficación de atractores caóticos de GRIISOFT para evaluar sus Comportamiento Temporales.
- Cuatro navegadores webs comunes (Chrome, Firefox, Edge y Opera) que fueron sometidos a pruebas de rendimiento para evaluar la Utilización de Recursos.
- Trescientos usuarios simulados en una prueba de estrés para evaluar la Capacidad del sistema bajo condiciones de alta demanda.

## 2) *Recopilación de datos*

La recopilación de datos por cada subcaracterística se describe a continuación.

### **Comportamiento temporal**

Las 5 personas seleccionadas realizaron interacciones con la aplicación y ejecutaron el script de Python tomado del proceso actual de GRIISOFT, generando gráficos del atractor caótico de Rossler. Cada individuo contribuyó con 6 tiempos de respuesta registrados durante estas interacciones, totalizando 60 registros en total.

### **Utilización de recursos**

Los navegadores Chrome, Firefox, Edge y Opera fueron sometidos a pruebas de rendimiento en las cuales se midió el consumo de CPU, RAM y GPU mientras se generaban los gráficos de atractores caóticos. Se realizaron 3 repeticiones de las pruebas para cada navegador, obteniendo un total de 12 registros de utilización de recursos.

### **Capacidad**

La evaluación de la Capacidad del sistema se llevó a cabo mediante una prueba de estrés diseñada en la herramienta JMeter para medir su rendimiento bajo cargas extremas. En esta prueba, se simularon 300 usuarios interactuando con la graficación de atractores caóticos, con un intervalo de 1 segundo entre cada interacción. Esta prueba permitió determinar la capacidad máxima del sistema para manejar usuarios simultáneos sin comprometer significativamente el rendimiento.

## ► **III. Marco de Resultados**

### *A. Evaluación de la eficiencia de desempeño según la norma ISO/IEC 25010*

En la evaluación de la eficiencia de desempeño se la lleva a cabo con el análisis de sus tres subcaracterísticas: Comportamiento temporal, Utilización de recursos y Capacidad.

## 1) *Comportamiento Temporal*

Para conocer el Comportamiento Temporal del proceso actual de graficación de atractores caóticos de GRIISOFT y del módulo de graficación de atractores caóticos de la aplicación web se tomaron los tiempos de respuesta que perciben los usuarios al ejecutar manualmente el script Python del proceso actual y utilizar la funcionalidad automatizada de la aplicación web, y una vez obtenidos estos resultados se pudo comparar ambos tiempos promedio y conocer si existe diferencia en los procesos de graficación de atractores caóticos.

### *a) Tiempo de respuesta*

Los tiempos de respuesta se pueden conocer de forma exacta debido a que se se agregaron líneas de código al inicio y fin de las funciones que grafican el atractor caótico en ambos procesos, estas líneas de código nos permiten contar el tiempo exacto que demoran en ejecutarse las funciones y como resultado mostrar el gráfico del atractor caótico. Cabe mencionar que con el fin de obtener el mismo resultado (grafico) se evaluó en base a la graficación del atractor caótico de Rossler con los siguientes datos.

- Número de puntos: 10000
- Número de pasos: 0,01
- Parámetro a: 0,2
- Parámetro b: 0,2
- Parámetro c: 5,7
- Variable x: 1
- Variable y: 1
- Variable z: 1

Dando como resultado resultado por cada uno de los procesos 30 tiempos, 6 por cada persona.

Los datos obtenidos del proceso actual de GRIISOFT se pueden observar en la TABLA II.

**Tabla II**

**TIEMPOS DE RESPUESTA DEL PROCESO ACTUAL DE GRIISOFT**

Tiempos (ms)	Persona 1	Persona 2	Persona 3	Persona 4	Persona 5
T1	2600	4750	1700	590	870
T2	2570	4520	1660	540	830
T3	4120	2300	1560	590	860
T4	2100	4150	1800	550	890
T5	3390	5120	1780	550	2860
T6	2150	3320	1690	540	2060
Total	16930	24160	10190	3360	8370
Promedio	2821,67	4026,67	1698,33	560	1395

Los datos obtenidos del proceso en la aplicación web se pueden observar en la TABLA III.

**Tabla III**

**Tiempos de respuesta del proceso en la aplicación web**

Tiempos (ms)	Persona 1	Persona 2	Persona 3	Persona 4	Persona 5
T1	51,69	163,89	227,73	102,8	78,34
T2	52,29	160,19	205,68	89,77	59,74
T3	56,34	138,94	234,71	64,01	95,83
T4	58,33	156,61	211,58	70,49	54,49
T5	52,92	162,07	227,1	66,04	53,21
T6	52,08	187,89	220,6	68,09	73,62
Total	323,65	969,59	1327,4	461,2	415,23
Prome-dios	53,94	161,60	221,23	76,87	69,21

### Comparación de resultados

En la TABLA IV se presentan y analizan los promedios de los datos obtenidos y descritos previamente de los procesos realizados actualmente por GRIISOFT y realizados por la aplicación web.

**Tabla IV**

**Comparación de tiempos de respuesta entre los dos procesos**

	Promedio del proceso actual	Promedio de la aplicación web	Reducción en el tiempo
Tiempos (ms)	10501,67	116,57	10385,10

Se puede notar una disminución de los tiempos de respuesta entre el proceso actual de GRIISOFT y la aplicación web, pues el proceso actual de GRIISOFT tiene un tiempo promedio de 10501,67 ms y el tiempo promedio de la aplicación web es del 116,57 ms, por lo que los investigadores

de GRIISOFT al usar la aplicación web estarían ahorrando hasta 10385,10 ms en la graficación de atractores caóticos, lo que convertido a minutos es 0,17. Al disminuir el tiempo de respuesta en cada iteración durante la graficación de atractores caóticos, se abre un margen significativo para la experimentación. Esto permite a los investigadores probar una variedad más amplia de enfoques, ajustar variables y profundizar en sus investigaciones en el mismo intervalo de tiempo.

### 2) Utilización de Recursos

Para evaluar la Utilización de Recursos de la aplicación web se monitoreó el uso de memoria RAM, el uso de CPU y el uso de GPU observando los datos obtenidos con el Administrador de Tareas del computador, los cuales fueron tabulados y se obtuvieron los promedios del uso de cada uno de los recursos mencionados.

#### a) Uso de memoria RAM

El análisis del uso de memoria RAM es esencial para comprender cómo la aplicación web de graficación de atractores caóticos maneja los recursos disponibles. Se llevó a cabo una evaluación del uso de memoria RAM, planteando como escenario, la graficación del atractor caótico de Rossler, el mismo que se lo realizó 3 veces en los navegadores: Chrome, Firefox, Edge y Opera, capturando los datos que se obtiene con el administrador de tareas. Una vez monitoreado el uso de memoria RAM en Chrome, Firefox, Edge y Opera, se procedió a sintetizar los resultados en la TABLA V.

**Tabla V**

**Resultado del uso de memoria RAM**

RAM (MB)	Chrome	Firefox	Edge	Opera
Prueba 1	342,6	384,6	121,8	96,2
Prueba 2	331,2	457,1	123	97,4
Prueba 3	398,1	467,9	121,3	76,6
Total	1071,9	1309,6	366,1	270,2
Promedio	357,3	436,5	122,03	90,1

Se puede observar que cada navegador tiene un consumo promedio de memoria RAM distinto,

siendo Firefox el de consumo más alto con 436,5 MB y Opera el de consumo más bajo con 90,1

### b) *Uso de CPU*

Es posible identificar patrones de comportamiento mediante el análisis de la utilización de la unidad central de procesamiento (CPU) durante la ejecución de la aplicación web destinada a la graficación de atractores. Por lo tanto, para comprender cómo cada navegador maneja los recursos de CPU, se llevó a cabo una evaluación centrada en la graficación del atractor caótico de Rossler. Varios navegadores, incluidos Chrome, Firefox, Edge y Opera, repitieron este proceso tres veces. Los datos de los administradores de tareas de cada navegador se registraron, lo que da una imagen precisa de la utilización de recursos de CPU en tiempo real.

Una vez monitoreado el uso de CPU en Chrome, Firefox, Edge y Opera, se procedió a sintetizar los resultados en la TABLA VI.

**Tabla VI**  
Resultado del uso de CPU

CPU (%)	Chrome	Firefox	Edge	Opera
<b>Prueba 1</b>	1,7	3,5	2,2	1,6
<b>Prueba 2</b>	1,1	3,9	2,3	2,5
<b>Prueba 3</b>	1,1	3,4	2,1	2,1
<b>Total</b>	3,9	10,8	6,6	6,2
<b>Promedio</b>	1,3	3,6	2,20	2,1

Se puede observar que cada navegador tiene un consumo promedio de uso de CPU distinto durante las pruebas de generación del atractor caótico de Rossler, siendo Firefox el de consumo más alto con 3.6 de recursos de CPU y Chrome el consumo más bajo con 1.3 de recursos de CPU.

### c) *Uso de GPU*

El análisis de la utilización de GPU permite tener una visión profunda de cómo se gestionan los recursos disponibles para la renderización gráfica durante la ejecución de la aplicación web de graficación de atractores, para comprender cómo cada navegador distribuye y gestionan los recursos de GPU. Se utilizó la graficación del atractor caótico de Rossler como escenario para realizar

una evaluación del rendimiento de la unidad de procesamiento gráfico (GPU). En los navegadores Chrome, Firefox, Edge y Opera este proceso se repitió tres veces. Los datos proporcionados por el administrador de tareas de cada navegador durante estas pruebas se registraron y analizaron. Este método permitió obtener una comprensión completa de cómo se gestiona la carga gráfica intensiva en cada situación.

Una vez monitoreado el uso de GPU en Chrome, Firefox, Edge y Opera, se procedió a sintetizar los resultados en la TABLA VII.

**Tabla VII**  
Resultado del uso de GPU

GPU (%)	Chrome	Firefox	Edge	Opera
<b>Prueba 1</b>	12	25	29	13
<b>Prueba 2</b>	13	24	27	12
<b>Prueba 3</b>	14	18	31	13
<b>Total</b>	39	67	87	38
<b>Promedio</b>	13	22,3	29	12,7

Se puede observar que cada navegador tiene un consumo promedio de uso de GPU distinto durante las pruebas de generación del atractor caótico de Rossler, siendo Edge el de consumo más alto con 29 de recursos de GPU y Chrome el consumo más bajo con 12.7 de recursos de CPU.

### 3) *Capacidad*

La evaluación de la capacidad es esencial para comprender los límites operativos del sistema y asegurarse de que esté preparado para afrontar cargas máximas y situaciones de demanda intensa. En esta evaluación se llegó a determinar el límite de usuarios concurrentes que la aplicación web de graficación de atractores caóticos, específicamente del módulo de graficación, puede manejar antes de que se alcancen umbrales críticos de rendimiento, en un ambiente controlado, en un servidor local.

#### a) *Escenario de carga*

Se crearon tres escenarios de carga en JMeter, cada uno representando una cantidad de 100 usuarios concurrentes. Estos escenarios abarcaron

desde cargas bajas hasta cargas muy altas, con incrementos de 100 usuarios en cada escenario de carga, hasta encontrar en que cantidad de usuarios empieza a dar errores en la carga del módulo de graficación.

*b) Monitoreo y registro de datos*

Una vez ejecutada las 3 primeras pruebas con un incremento de 100 usuarios en cada prueba, se obtuvieron los datos que se observan en la TABLA VIII.

**Tabla VIII**  
Resultados de las tres primeras pruebas en JMeter.

Con 100 Usuarios						
Label	# Samples	Average	Min	Max	Std. Dev.	Error %
HTTP Request	100	8254	0	16214	4668.06	0.00%
TOTAL	100	8254	0	16214	4668.06	0.00%
Con 200 Usuarios						
HTTP Request	200	17182	0	33904	9657.52	0.00%
TOTAL	200	17182	0	33904	9657.52	0.00%
Con 300 Usuarios						
HTTP Request	300	12471	0	35946	12068.66	32.00%
TOTAL	300	12471	0	35946	12068.66	32.00%

Se puede observar que en las primera dos pruebas no se obtuvieron errores, sin embargo, en la tercera prueba con 300 usuarios se obtuvo un 32% de errores lo que es igual a 96 usuarios que el servidor no permitió el ingreso al módulo de graficación. También, se puede decir que, hasta 204 usuarios concurrentes el sistema no debería mostrar errores, por lo que se procedió a realizar una prueba con 204 usuarios concurrentes. Los resultados de la cuarta prueba se muestran en la TABLA IX.

**Tabla IX**  
Resultados de la cuarta prueba en JMeter

Label	# Samples	Average	Min	Max	Std. Dev.	Error %
HTTP Request	204	24425	0	43502	12594.97	0.00%
TOTAL	204	24425	0	43502	12594.97	0.00%

Se muestran resultados esperados al ejecutar la prueba con 204 usuarios, debido a que no

existieron errores, por lo que se procedió a encontrar cual es el límite de usuarios, para aquello se ejecutaron pruebas con un aumento de un usuario en cada prueba, hasta encontrar cual es el límite de usuarios en la que el servidor responde. En la TABLA X se observa la ejecución de las dos últimas pruebas, que contienen el límite de usuarios sin errores, y el número de usuarios en las que el servidor no responde.

**Tabla X**  
Resultados de las dos últimas pruebas en JMeter

Con 207 Usuarios						
Label	# Samples	Average	Min	Max	Std. Dev.	Error %
HTTP Request	207	17661	0	35057	10296.9	0.00%
TOTAL	207	17661	0	35057	10296.9	0.00%
Con 208 Usuarios						
HTTP Request	208	17288	0	34021	9843.26	0.48%
TOTAL	208	17288	0	34021	9843.26	0.48%

*B) Evaluación comparativa de tiempos de respuesta mediante un test estadístico*

A través del análisis estadístico, se busca determinar si existen diferencias significativas en los tiempos de respuesta del proceso actual de graficación de atractores de GRIISOFT versus la aplicación web desarrollada.

*1) Planteamiento de la hipótesis*

Hipótesis nula (H0): Los tiempos de respuesta del proceso actual de graficación de atractores de GRIISOFT y la aplicación web de graficación de atractores son iguales.

Hipótesis alternativa (Ha): Los tiempos de respuesta del proceso actual de graficación de atractores caóticos de GRIISOFT y la aplicación web de graficación de atractores caóticos son diferentes.

*2) Recopilación de datos*

Los tiempos de respuesta previamente registrados

en el comportamiento temporal se utilizarán como pretest correspondientes a los tiempos del proceso actual de graficación de atractores caóticos de GRIISOFT, que se muestran en la TABLA II, y como postest correspondientes a los tiempos obtenidos a través de la aplicación web de graficación de atractores caóticos, que se muestran en la TABLA III. Para su análisis posterior en R Studio con el lenguaje R, estos datos se han reunido en la TABLA XI. Es importante destacar que los tiempos de respuesta se han redondeado a números enteros para evitar posibles errores de procesamiento de datos en R.

**Tabla XI**  
Tiempos de respuesta agrupados por Pretest y Postest

persona	periodo	respuesta
1	pretest	2600
1	postest	52
2	pretest	4750
2	postest	164
3	pretest	1700
3	postest	228
4	pretest	590
4	postest	103
5	pretest	870
5	postest	78
1	pretest	2570
1	postest	52
2	pretest	4520
2	postest	160
3	pretest	1660
3	postest	206
4	pretest	540
4	postest	90
5	pretest	830
5	postest	60
1	pretest	4120
1	postest	56
t	pretest	2300
2	postest	139
3	pretest	1560
3	postest	235
4	pretest	590
4	postest	64
5	pretest	860
5	postest	96
1	pretest	2100

1	postest	58
2	pretest	4150
2	postest	157
3	pretest	1800
3	postest	212
4	pretest	550
4	postest	70
5	pretest	890
5	postest	54
1	pretest	3390
1	postest	53
2	pretest	5120
2	postest	162
3	pretest	1780
3	postest	227
4	pretest	550
4	postest	66
5	pretest	2860
5	postest	53
1	pretest	2150
1	postest	52
2	pretest	3320
2	postest	188
3	pretest	1690
3	postest	203
4	pretest	540
4	postest	68
5	pretest	2060
5	postest	74

### 3) Selección del Test estadístico

Se optó por realizar un análisis de varianza (ANOVA) en este contexto específico, donde se trabajó con dos grupos distintos (pretest y postest). El propósito de esta elección estadística es discernir si existen diferencias significativas en las medias de dichos grupos. La aplicación del ANOVA resulta pertinente en situaciones donde se busca determinar si al menos uno de los grupos difiere de manera estadísticamente significativa de los otros en términos de la variable de interés.

### 4) Aplicación del Test estadístico

El test estadístico se realizó usando la herramienta R Studio, en la Figura 7. y en la Figura 8. se detalla la ejecución:

```

1 library("psych")
2
3 datos = test_thalia_zarate
4
5
6 describeby(datos$respuesta, datos$periodo)
7
8 boxplot(respuesta ~ periodo, data=datos, main="Diagrama de Caja", xlab="Periodo", ylab="Duración(ms)")
9
0 model = aov(respuesta ~ persona + periodo + Error(persona/periodo), data = datos)
1
2 summary(model)
    
```

Fig. 7. Código en lenguaje R del test estadístico a ejecutar usando R Studio

En la Figura 8. se observan las estadísticas descriptivas agrupadas por el factor "periodo". Esto te muestra la media, la desviación estándar, la mediana y otros estadísticos para cada período (pretest y postest).

```

Descriptive statistics by group
group: postest
vars n mean sd median trimmed mad min max range skew kurtosis se
X1 1 30 116 65.89 84 109.75 46.7 52 235 183 0.55 -1.37 12.03
-----
group: pretest
vars n mean sd median trimmed mad min max range skew kurtosis se
X1 1 30 2100.33 1388.29 1790 1957.92 1401.06 540 5120 4580 0.65 -0.76 253.47
    
```

Fig. 8. Estadísticas descriptivas agrupadas por el factor periodo

En la Figura 9. Se puede observar de mejor manera la distribución de los tiempos de respuesta en los dos períodos en un diagrama de caja.

```

Descriptive statistics by group
group: postest
vars n mean sd median trimmed mad min max range skew kurtosis se
X1 1 30 116 65.89 84 109.75 46.7 52 235 183 0.55 -1.37 12.03
-----
group: pretest
vars n mean sd median trimmed mad min max range skew kurtosis se
X1 1 30 2100.33 1388.29 1790 1957.92 1401.06 540 5120 4580 0.65 -0.76 253.47
    
```

Fig. 9. Diagrama de caja con la distribución de los tiempos

### 5) Resultados del Test estadísticos

En la Figura 10. Se puede observar los resultados del Test estadístico ANOVA.

```

Error: persona
Df Sum Sq Mean Sq
persona 1 12189000 12189000

Error: persona:periodo
Df Sum Sq Mean Sq
periodo 1 30120591 30120591

Error: Within
Df Sum Sq Mean Sq F value Pr(>F)
periodo 1 40721291 40721291 71.15 1.5e-11 ***
Residuals 56 32051604 572350
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
    
```

Fig. 10. Resultados del Test estadístico ANOVA

Se detallan los resultados del Test a continuación.

El análisis compara la variabilidad entre grupos (personas, periodos) con la variabilidad dentro

de los grupos. La hipótesis nula es que no hay diferencias significativas entre los grupos, y la hipótesis alternativa es que los grupos son diferentes entre sí. El estadístico F se utiliza para evaluar si la variabilidad entre grupos es significativamente mayor que la variabilidad dentro de los grupos

Para determinar si las diferencias son significativas, el ANOVA calcula una estadística llamada F-value. Este valor se obtiene al dividir la variabilidad entre grupos por la variabilidad dentro de los grupos. Si el F-value es grande y la probabilidad asociada con él ( $Pr(>F)$ ) es pequeña, entonces se concluye que uno de los grupos es significativamente diferente del otro.

### 6) Interpretación de los resultados

El resumen del ANOVA proporciona los resultados de la prueba de hipótesis para cada uno de los términos en el modelo:

- El factor "periodo" muestra un valor de F significativo ( $Pr(>F) < 0.001$ ), lo que indica que existe una diferencia significativa entre los períodos (pretest y postest) en términos de tiempos de respuesta. En otras palabras, hay una diferencia estadísticamente significativa entre los tiempos de respuesta antes y después del proceso.
- La interacción "persona:periodo" no se muestra en el resumen, lo que podría indicar que no hay una diferencia significativa en cómo los tiempos de respuesta varían entre las personas en diferentes períodos. Esto no necesariamente significa que la interacción no es relevante, solo que no es significativa en este modelo.
- El valor de F para la variación dentro de "persona:periodo" (Within) también es significativo, lo que indica que hay diferencias significativas entre las respuestas de las personas en diferentes períodos, una vez que se ha tenido en cuenta la variación individual y la variación debida a la interacción.

### Conclusión del Test estadístico

Basándonos en los resultados del ANOVA, podemos concluir que existe una diferencia significativa en los tiempos de respuesta entre los períodos (pretest y posttest). Por lo tanto, podríamos rechazar la hipótesis nula ( $H_0$ ) y aceptar la hipótesis alternativa ( $H_a$ ). Esto significa que los tiempos de respuesta del proceso actual de graficación de atractores de GRIISOFT y la aplicación web de graficación de atractores son diferentes en términos de los dos períodos evaluados.

### » V. Conclusiones

Con base en la observación y revisión documental, se realizó un análisis de los atractores caóticos empleados por GRIISOFT, que abarcan el atractor de Lorenz, Rossler, Chen y Sprott. Estos atractores presentan similitudes, tales como comportamientos no lineales y sensibilidad a condiciones iniciales, siendo todos tridimensionales y compartiendo el mismo conjunto de variables y parámetros. Estos hallazgos resultaron esenciales para el desarrollo del algoritmo destinado a la graficación los atractores en la aplicación web.

En la implementación de los módulos de gestión de usuarios, autenticación, información, creación y consulta de gráficos en la aplicación web, se adoptó la metodología Scrum. Esta metodología facilitó la agilidad y flexibilidad en el desarrollo al gestionar iteraciones, permitiendo así completar el producto dentro de los plazos establecidos de manera efectiva.

La eficiencia de desempeño del proceso actual de GRIISOFT y la aplicación web se evaluó utilizando la norma ISO/IEC 25010, específicamente en sus tres subcaracterísticas: Comportamiento temporal, Utilización de recursos y Capacidad. Se registraron los siguientes resultados: el tiempo de respuesta fue de 10501,67 ms para el proceso actual y 116,57 ms para la aplicación web al graficar un atractor caótico, lo que indica una reducción del 98,89% en el tiempo de respuesta entre ambos procesos. En cuanto al consumo de recursos, se observó que el navegador con menor consumo de memoria

RAM es Opera con 90,1 MB, el menor consumo de CPU se encontró en Chrome con un 1,3%, y el menor consumo de GPU también correspondió a Chrome, con un 13%. Además, se determinó que el módulo de graficación puede soportar hasta 207 usuarios concurrentes.

### » V. Referencias

- M. J. Sametband, "ENTRE EL ORDEN Y EL CAOS. La complejidad," 1999.
- [2] E. I. Amaya Barrera, C. A. Suárez Parra, R. E. Huérfano Ortiz, J. D. Moreno Posada, and F. A. Parra Fuentes, "Modelo de encriptación simétrica basada en atractores caóticos," *Ingeniería*, vol. 21, no. 3, 2016, doi: 10.14483/udistrital.jour.reving.2016.3.a08.
- [3] S. H. Strogatz, *Nonlinear Dynamics And Chaos*. 1994. Accessed: Mar. 04, 2023. [Online]. Available: <http://users.uoa.gr/~pjoannou/nonlin/Strogatz,%20S.%20H.%20-%20Nonlinear%20Dynamics%20And%20Chaos.pdf>
- [4] E. N. Lorenz, "Deterministic Nonperiodic Flow," *J Atmos Sci*, vol. 20, no. 2, pp. 130–141, 1963, Accessed: Jan. 21, 2023. [Online]. Available: [https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469\\_1963\\_020\\_0130\\_dnf\\_2\\_0\\_co\\_2.xml](https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml)
- [5] I. F. Rodríguez Rodríguez, E. I. Amaya Barrera, C. A. Suarez Parra, and J. D. Moreno Posada, "Algoritmo de Encriptación de Imágenes Utilizando el Atractor Caótico de Lorenz," *Ingeniería*, vol. 22, no. 3, 2017, doi: 10.14483/23448393.11976.
- [6] J. Ruitter, "Scholarship @ Claremont Practical Chaos: Using Dynamical Systems to Encrypt Audio and Visual Data," 2019. [Online]. Available: [https://scholarship.claremont.edu/scripps\\_theses/1389](https://scholarship.claremont.edu/scripps_theses/1389)
- [7] AcademiaLab, "Atractor," <https://academia-lab.com/enciclopedia/attractor/>.
- [8] M. Fuentes, "Dinámica científica y medidas de complejidad." Accessed: Aug. 14, 2023. [Online]. Available: [https://books.google.com.ec/s?id=Z8T0DwAAQBAJ&prints\\_ec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/s?id=Z8T0DwAAQBAJ&prints_ec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- [9] C. Andreu, J. Echave, and G. Buella, "Psicothema," *Psicothema* Vol. 10 (no 2). Accessed: Aug. 14, 2023. [Online]. Available:

- <https://www.psicothema.com/pi?pii=168>
- [10] E. Durán Ruiz, “Teoria del Caos\_2.” Accessed: Jul. 29, 2023. [Online]. Available: [http://recursostic.educacion.es/descartes/web/materiales\\_didacticos/Teoria\\_caos/teoria\\_del\\_caos\\_2.html](http://recursostic.educacion.es/descartes/web/materiales_didacticos/Teoria_caos/teoria_del_caos_2.html)
- [11] M. Á. Medina Torres, “La complejidad de la naturaleza: Fractales, caos y jugando a hacer montones de arena,” 2007.
- [12] S. M. Alwan, A. M. Al-Mahdi, and O. H. Odhah, “Optimal Control and Bifurcation Issues for Lorenz-Rössler Model,” *Open Journal of Optimization*, vol. 09, no. 03, pp. 71–85, 2020, doi: 10.4236/ojop.2020.93006.
- [13] E. Rossler, “AN EQUATION FOR CONTINUOUS CHAOS,” 1976.
- [14] C. Aguilar Ibáñez et al., “Identificación del sistema de Rössler: enfoque algebraico y algoritmos genéticos,” 2005.
- [15] G. Chen, “YET ANOTHER CHAOTIC ATTRACTOR,” 1999. [Online]. Available: [www.worldscientific.com](http://www.worldscientific.com)
- [16] P. Augustová and Z. Beran, “Characteristics of the Chen Attractor,” *Advances in Intelligent Systems and Computing*, vol. 210, pp. 305–312, 2013, doi: 10.1007/978-3-319-00542-3\_31.
- [17] Q. Lai and S. Chen, “Generating multiple chaotic attractors from Sprott B System,” *International Journal of Bifurcation and Chaos*, vol. 26, no. 11, Oct. 2016, doi: 10.1142/S0218127416501777.
- [18] E. León -GAMALIEL, S. Tellez -CANEK, S. Ibarra -YUMA, and N. Pérez -JOSE CRUZ, “Diseño de Sistemas Caóticos de Sprott usando los Exponentes de Lyapunov.”