VOLUMEN 6, N°1 / ENERO - JUNIO 2024 / e - ISSN: 2661

CIFRADO DE TEXTO MEDIANTE ATRACTORES CAÓTICOS: CRYPTOGUARD

Text encryption using chaotic attractors: Cryptoguard

Jemmy Anahí Puzma Granda [1]	jemmy.puzma@espoch.edu.ec
Danilo Mauricio Pástor Ramírez [2]	danilo.pastor@espoch.edu.ec
Raúl Hernán Rosero Miranda [3]	raul.rosero@espoch.edu.ec
Maricela Jiménez Rodríguez [4]	maricela.jrodriguez@academicos.udg.mx
Omar S. Gómez [5]	ogomez@espoch.edu.ec

[1,2,3,5] Facultad de Informática y Electrónica, Escuela Superior Politécnica del Chimborazo (ESPOCH), Riobamba, Ecuador.
[4] Profesora-investigadora en el Centro Universitario de la Ciénega, Universidad de Guadalajara.

RESUMEN

El presente estudio se embarca en la exploración de la criptografía basada en atractores caóticos, desarrollando una aplicación web destinada al cifrado y descifrado de cadenas de texto utilizando la sincronización de estos atractores. El primer paso de este estudio involucró una revisión detallada de cuatro sistemas caóticos para comprender a fondo las fórmulas de cada atractor. A través de una combinación de análisis matemático y programación, se implementó la sincronización de cada atractor en la aplicación web, utilizando la metodología SCRUMBAN, una combinación de los marcos de trabajo ágiles Scrum y Kanban. Las pruebas de Kruskal-Wallis, una prueba estadística no paramétrica utilizada para comparar tres o más grupos independientes de datos, revelaron diferencias significativas en los tiempos de sincronización, cifrado y descifrado entre los cuatro atractores. En términos concretos, estos resultados sugieren que el atractor de Lorenz es el más rápido para realizar la sincronización, cifrado y descifrado de cadenas de texto.

Palabras Clave: Cifrado de texto, Atractores caóticos, Seguridad web, Cifrado simétrico, Sistemas dinámicos caóticos.

ABSTRACT

The present study embarks on the exploration of cryptography based on chaotic attractors, we develop a web application intended for the encryption and decryption of text strings using the synchronization of these attractors. The first step of this study involved a detailed review of four chaotic systems to fully understand the formulas of each attractor. Through a combination of mathematical analysis and programming, the synchronization of each attractor was implemented in the web application, using the SCRUMBAN methodology, a combination of the agile frameworks Scrum and Kanban. Kruskal-Wallis tests, a nonparametric statistical test was used to compare three or more independent sets of data, revealed significant differences in synchronization, encryption, and decryption times between the four attractors. In concrete terms, these results suggest that the Lorenz attractor is the fastest to perform synchronization, encryption and decryption of text strings.

Palabras Clave: Text encryption, Chaotic attractors, Web security, Symmetric encryption, Chaotic dynamic systems.

I. Introducción

En la era digital actual, la seguridad de los datos se ha convertido en una prioridad crítica. La criptografía, el arte y la ciencia de cifrar información, es fundamental para proteger la comunicación en la extensa red de Internet. Con la evolución constante de las amenazas cibernéticas,

Fecha de Recepción: 05/dic/2023. **Fecha de Aceptación:** 17/ene/2024 **DOI:** https://doi.org/10.47187/perspectivas.6.1.213

se requieren métodos de cifrado avanzados que no solo sean robustos sino también adaptables. En este contexto, la criptografía basada en atractores caóticos emerge como una alternativa prometedora, aprovechando la imprevisibilidad inherente a los sistemas caóticos para fortalecer la seguridad del cifrado.

Este artículo presenta el desarrollo de una aplicación web innovadora diseñada para el cifrado y descifrado de cadenas de texto, utilizando atractores caóticos como su piedra angular. Los atractores caóticos, derivados de sistemas dinámicos no lineales, ofrecen propiedades únicas como la sensibilidad a las condiciones iniciales y la mezcla topológica, las cuales son explotadas en este estudio para mejorar la seguridad del cifrado.

La contribución principal de este estudio es la implementación de un algoritmo de cifrado basado en atractores caóticos dentro de una interfaz de aplicación web accesible y fácil de usar. Se discutirá los atractores caóticos utilizados, desarrollo del algoritmo, la arquitectura de la aplicación y la evaluación de la seguridad del sistema propuesto. Además, se examinará la eficacia del cifrado en términos de comportamiento temporal.

II. Marco teórico

A. Trabajos relacionados

Córdova Ramírez [1] tuvo como objetivo el desarrollo de un sistema en la cual se propone sistematizar el proceso de la redacción y generación de historiales médicos para reducir el tiempo de firma y aprobación de estos. Se utilizó el sistema RSA (Rivest, Shamir y Adleman) y la función hash SHA-256 para crear una firma digital.

Sheikholeslam utilizó sistemas dinámicos con atractores caóticos en el cifrado. Se basó en el sistema Encryption Dynamical para generar una clave de sincronización, y conseguir que el descifrado pueda actualizarse a las condiciones iniciales antes de generar el bloque. Como resultado se consiguió realizar una modificación discreta del sistema de Lorenz [2].

Gómez, Rosero, Estrada y Jiménez agregaron un mecanismo de seguridad a los objetos JSON mediante el uso de sincronización caótica. Y el resultado fue que este enfoque se puede aplicar como JSON Web Encryption (JWE) [3].

Montalván desarrolló el mecanismo de cifrado basado en el algoritmo criptográfico simétrico AES (MECIB-AES) para comparar la seguridad que brinda este a la información cifrada. Como resultado se obtuvo que la implementación de las modificaciones Mix-Shift, Mix-Key y Move-C ayudó a realizar diferentes pruebas donde se aceptó la hipótesis nula la cual midió la entropía, con un nivel de confiabilidad del 95% y un error del 5%, el análisis de frecuencias presentó variaciones en cada prueba realizada, la autocorrelación dió como resultado una mayor similitud de secuencias a favor del MECIB-AES, aunque puede tomarse como desventaja que los valores no son grandes por lo cual se consideró viable el algoritmo [4].

A pesar de que en el trabajo de Gómez et al. Agrega un mecanismo de seguridad a los objetos JSON mediante la sincronización caótica, no utiliza cadenas de texto para observar el cifrado y descifrado de la misma. De igual manera, en los trabajos revisados se utilizan sistemas de cifrado como RSA, SHA-256 y AES sin embargo, ninguno implementa cifrados con sistemas caóticos.

B. Caos

Ribero y Ramírez Proponen la manera en que un fenómeno presenta fluctuaciones en el tiempo es a menudo descrita por una ecuación diferencial. Por ejemplo, cuando una observación de un fenómeno en el periodo n+1 es una función de la observación del período n, que se puede expresar en general en la Ec. 1 [5].

$$X(n + 1) = F[X(n)] \quad (1)$$

Donde F[X] sea una ecuación diferencial no lineal y de primer orden.

Desde el punto de vista matemático, se trata de ecuaciones diferenciales ordinarias, esto quiere decir que, poseen una única variable independiente

que cumplen las condiciones necesarias para asegurar la existencia y la unicidad de sus soluciones para cada conjunto de valores de las variables dependientes [6].

C. Sincronización caótica

La sincronización caótica consiste en hacer coincidir y converger en la misma trayectoria varios sistemas caóticos después de un tiempo suficiente. La idea general de la sincronización caótica utilizada en comunicaciones seguras es la siguiente. Primero, el transmisor cifra la información mediante un sistema caótico. Después, la información cifrada es enviada a través de un canal para ser recibida por el receptor. Finalmente, el receptor utiliza la sincronización para recuperar el mensaje original de la información cifrada [7].

D. Atractores

El término atractor extraño se usa para describir una región o forma hacia la cual los puntos son llevados como resultado de cierto proceso que muestra una dependencia sensible de las condiciones iniciales (es decir, puntos que inicialmente están cerca del atractor se separa exponencialmente con el tiempo) [8].

1. Atractor de Rossler

El atractor de Rössler es un sistema de tres ecuaciones diferenciales ordinarias no lineales estudiadas por el autor. Estas ecuaciones diferenciales definen un sistema dinámico del tiempo continuo que muestra dinámicas caóticas asociadas con las propiedades fractales del atractor. Algunas propiedades pueden ser deducidas a través de métodos lineales como auto vectores, pero las principales características del sistema requieren métodos no lineales como Aplicaciones de Poincaré o diagramas de bifurcación [9].

Se considera al sistema de Rossler mediante las Ec. 2, Ec. 3, y Ec. 4. [3].

(2) $(\dot{x}1) = -y1 - z1$

(3) $(\dot{y}1) = x1 + ay1$

(4) $(\dot{z}1) = b + z1(x1 - c)$

Se sabe que en los parámetros $\left\{\begin{matrix} a = 0{,}2 \\ b = 0{,}2 \\ c = 5{,}7 \end{matrix}\right\}$ este sistema presenta comportamiento caótico (nótese que solo se cuenta con un término no lineal) [10].

Una imagen referencial de como se ve el atractor de Rossler es la que se muestra en la Fig:1.

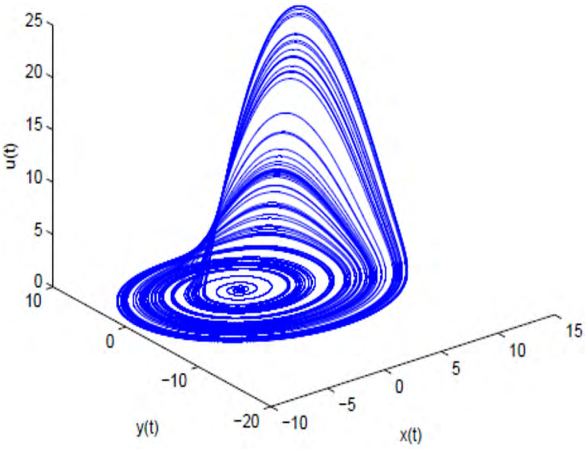

Fig. 1. *Atractor de Rossler*

2. Atractor de Lorenz

El atractor de Lorenz, es un sistema determinístico tridimensional derivado de las ecuaciones simplificadas de rollos de convección que se producen en las ecuaciones dinámicas de la atmósfera terrestre [9].

En el caso del sistema de Lorenz, el esquema maestro está representado por las Ecuaciones diferenciales ordinarias no lineales, ver en la Ec. 5, Ec. 6 y Ec.7 [3].

(5) $(\dot{x}1) = \sigma(y1 - x1)$

(6) $(\dot{y}1) = - x1z1 + \rho x1 - y1$

(7) $(\dot{z}1) = x1y1 - \beta z1$

donde $x1$, $y1$, $z1$ son las condiciones iniciales, y $\left\{\begin{matrix} a = 0{,}2 \\ b = 0{,}2 \\ c = 5{,}7 \end{matrix}\right\}$ son los parámetros del sistema.

Se visualiza el Atractor de Lorenz en la Fig. 2.

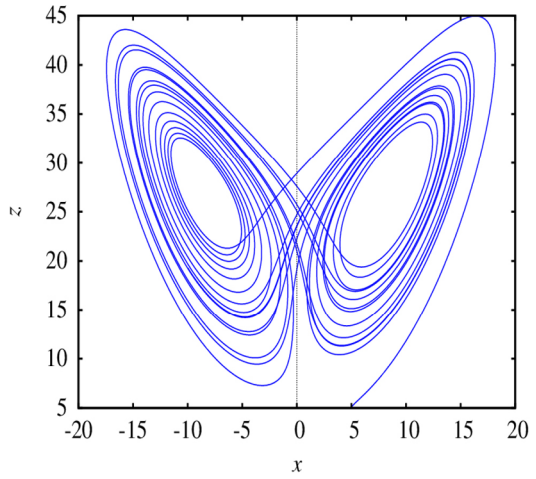

Fig. 2. *Atractor de Lorenz*

3. Atractor de Chen

Es un nuevo atractor caótico en un sistema autónomo tridimensional simple, que se asemeja a algunas características familiares de los atractores de Lorenz y Rossler [11].

Este sistema representado por la Ec. 8, Ec.9 y Ec. 10.

(8) $\dot{x} = a(y - x)$

(9) $\dot{y} = (c - a)x - xz + cy$

(10) $\dot{z} = xy - bz$

donde x, y, z son las condiciones iniciales del sistema y $\begin{cases} a = 35 \\ b = 3 \\ c = 28 \end{cases}$ son los parámetros del sistema.

Se observa el atractor en la Fig. 3.

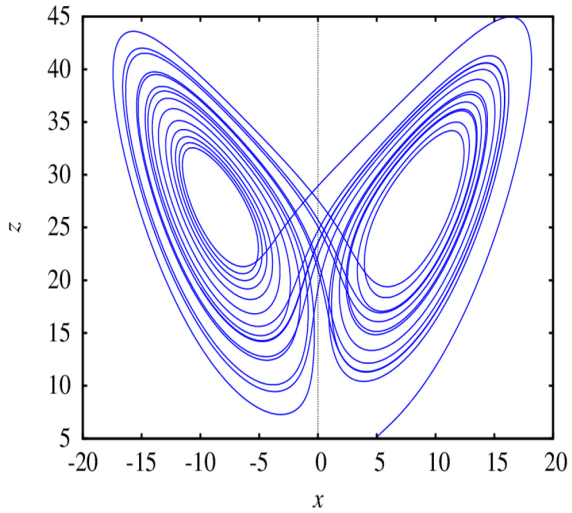

Fig. 3. *Atractor de Chen*

A finales del siglo pasado J. C. Sprott, introdujo una ecuación que producía caos en ciertos valores de parámetros, una característica importante de esta ecuación era la siguiente: Era una ecuación diferencial de tercer orden, que podría llevarse a tres de primer orden. Sprott construyó una serie de ecuaciones, que desde un punto de vista matemático eran muy sencillas, pero sus soluciones muestran estructuras muy complejas [12].

Está representado por las Ec. 11, Ec. 12 y Ec. 13 [13].

(11) $\dot{x} = a(y - x)$

(12) $\dot{y} = bxz$

(13) $\dot{z} = c - xy$

Con sus parámetros correspondientes a $\begin{cases} a = 5 \\ b = 2 \\ c = 1{,}6 \end{cases}$

El atractor se lo grafica de la siguiente manera, como se muestra en la Fig. 4.

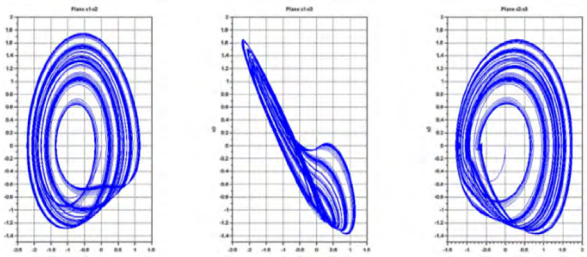

Fig. 4. *Atractor de Sprott a), en planos x1-x2, x1-x3 y x2-x3*

III. Desarrollo de la aplicación web

A. Elicitación de requerimientos

En este sistema, los usuarios solicitan la capacidad de seleccionar entre distintos atractores para el cifrado y descifrado de textos, enfatizando la necesidad de una interfaz que permita esta selección de manera sencilla. Además, requieren funcionalidades para registrarse e iniciar sesión, lo que subraya la importancia de un sistema de acceso seguro y eficiente. Los usuarios también piden poder ingresar cadenas de texto para cifrar o descifrar, y desean visualizar el resultado del cifrado junto con información detallada sobre los tiempos de procesamiento. Por otro lado, los administradores del sistema requieren habilidades similares para registrarse e iniciar sesión, pero con capacidades adicionales como la eliminación

de usuarios y datos de cifrado/descifrado para mantener la actualidad y relevancia de los datos. Además, necesitan generar informes estadísticos a partir de los datos de cifrado y descifrado para análisis y seguimiento, así como la capacidad de monitorear las IPs y ubicaciones de inicio de sesión de los usuarios para asegurar un uso adecuado del sistema. Tanto usuarios como administradores enfatizan la necesidad de una función de cierre de sesión para mantener la seguridad y la integridad del sistema.

B. Conceptualización del sistema para el administrador

En la Fig. 5 se observa que el administrador se tiene que autenticarse para ingresar a la página principal, una vez ingresado puede escoger entre las opciones: Inicio, Atractores, Usuarios, Inicios de Sesión e Informes. En la opción de Inicio podrá visualizar información del cifrado y los informes datos estadísticos, se realizará la petición en la base de datos.

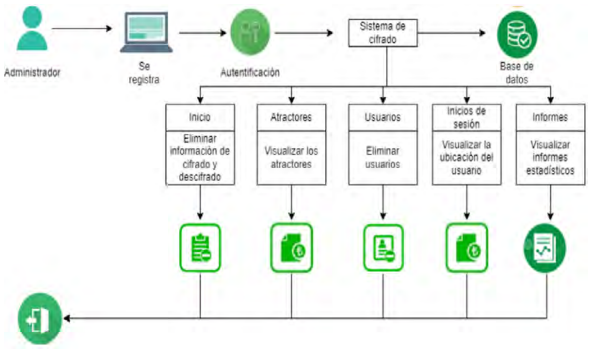

Fig. 5. *Conceptualización del sistema para el administrador.*

C. Conceptualización del sistema para el usuario

Se observa en la Fig. 6. que el usuario se autentica para ingresar al menú principal, donde constará de un inicio y opciones de cifrado, mediante Rossler, Lorenz, Chen y Sprott. En cada atractor se podrá ingresar una cadena de texto y al dar clic en el botón cifrar se mostrará información de tiempos de sincronización, cifrado, también tendrá la opción de descifrar en donde se ingresa la cadena de texto cifrada y mostrará la cadena descifrada, y el atractor utilizado con su tiempo. Se almacenará en la base de datos.

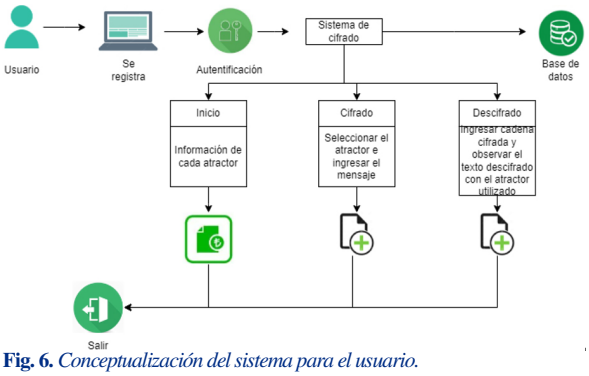

Fig. 6. *Conceptualización del sistema para el usuario.*

D. Diseño del algoritmo

El diseño del algoritmo es una fase crítica en la creación de una solución de cifrado efectiva y segura. Este proceso se dividió en dos etapas principales: la sincronización caótica y la construcción de las funciones de cifrado y descifrado. A continuación, se detallan los pasos y consideraciones metodológicas que guiaron el desarrollo del algoritmo.

1. Sincronización caótica

La sincronización caótica es el fundamento sobre el cual se construye la seguridad del algoritmo de cifrado. Para lograr una sincronización efectiva, se seleccionaron atractores caóticos basados en su complejidad dinámica y sensibilidad a las condiciones iniciales. Se implementó el método de sincronización mediante el vector de acoplamiento, asegurando que el emisor y el receptor pudieran generar secuencias caóticas idénticas en ausencia de diferencias en las condiciones iniciales. En la Fig. 7, Fig. 8, Fig. 9 y Fig. 10 se muestra como se sincronizan el sistema maestro y esclavo de cada uno de los atractores.

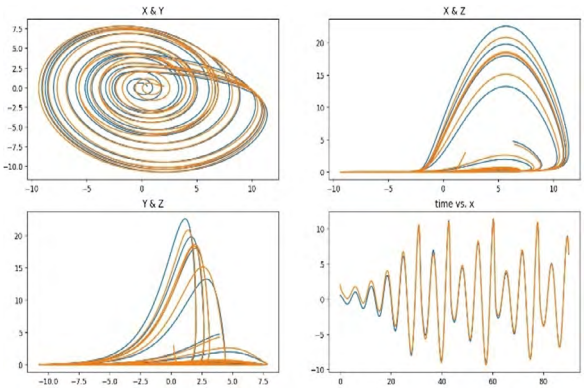

Fig. 7. *Sincronización del atractor de Rossler.*

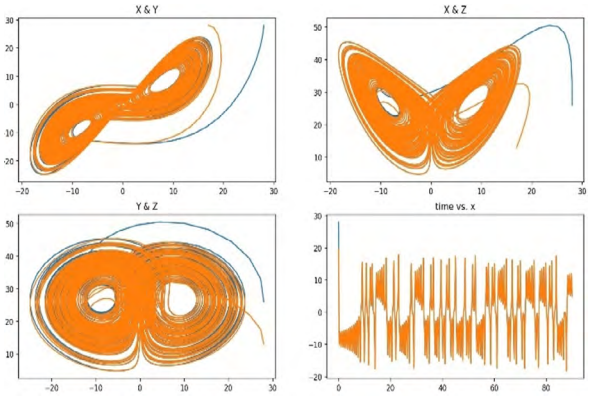

Fig. 8. *Sincronización del atractor de Lorenz.*

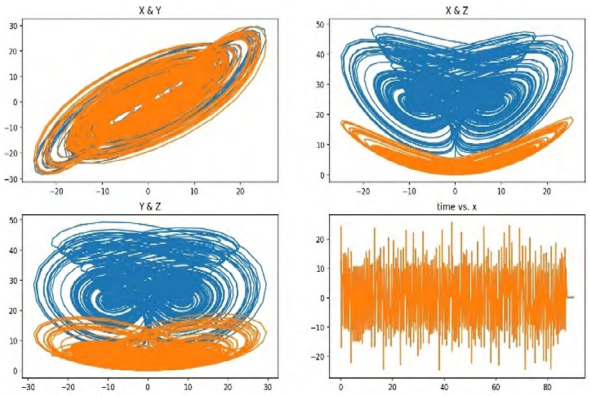

Fig. 9. *Sincronización del atractor de Chen.*

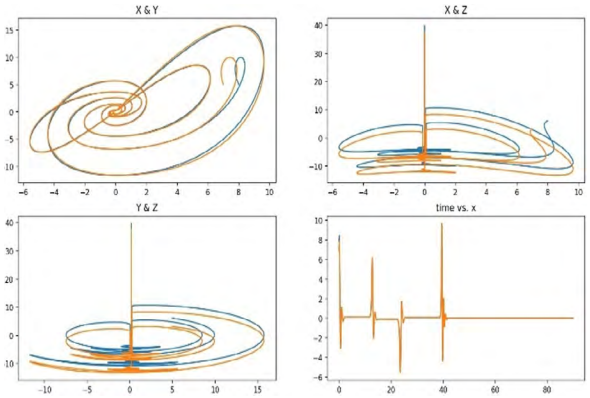

Fig. 10. *Sincronización del atractor de Sprott.*

1. Función de cifrado y descifrado

En base a la sincronización caótica, se creó las funciones de cifrado y descifrado que reciben como parámetros una cadena de texto y la opción dependiendo al atractor seleccionado, de acuerdo con esto se aplican las ecuaciones pertenecientes a cada uno de los atractores.

Esta función cifra un mensaje de texto utilizando la dinámica de sistemas caóticos sincronizados. Cada carácter del mensaje se integra en una trayectoria caótica, y el resultado se codifica en base64. La sincronización caótica entre dos sistemas (maestro y esclavo) es clave en este proceso, ya que asegura que solo quien conozca las condiciones iniciales y las ecuaciones del sistema pueda descifrar el mensaje correctamente.

2. Arquitectura MVC

Para documentar la arquitectura se utiliza el modelo 4+1 de krutchen, ver Fig. 11. Cada vista aborda un conjunto específico de preocupaciones de los diferentes interesados en el sistema.

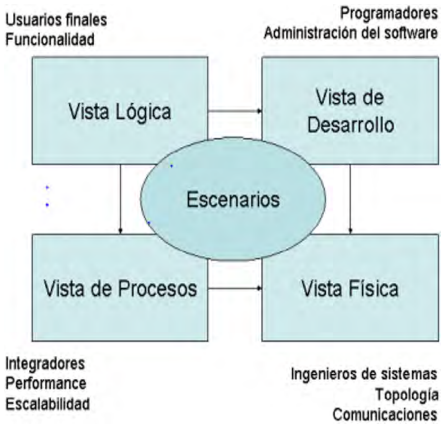

Fig. 11. *Vistas del modelo 4+1 de Krutchen*

1. Vista lógica

En la Fig. 12. de la vista lógica se observa el diagrama de clases que compone la aplicación web de cifrado. Esta vista es de gran interés para los desarrolladores de software y los analistas de sistemas, ya que se relaciona con la funcionalidad principal del sistema.

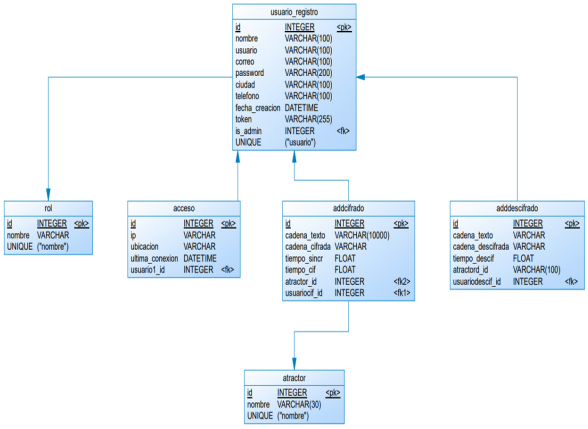

Fig. 12. *Diagrama de clases*

2. Vista de despliegue

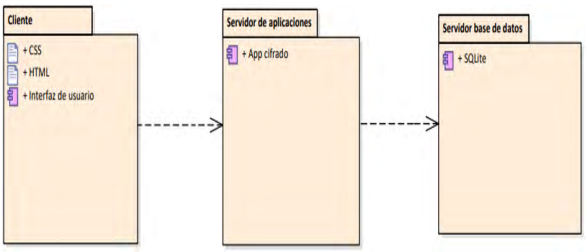

Fig. 13. *Diagrama de componentes.*

En esta vista se ha elegido representarla mediante el diagrama de componentes, que se visualiza en la Fig. 13. Incluye aspectos como la gestión de la configuración y la organización de software en unidades de implementación como archivos de código fuente, scripts, bibliotecas.

3. Vista de procesos

En la Fig. 14 se muestra el diagrama de actividades del cifrado y en el Fig.15 del descifrado. Se ocupa de los aspectos dinámicos del sistema, explicando cómo se ejecuta el sistema en términos de procesos o hilos.

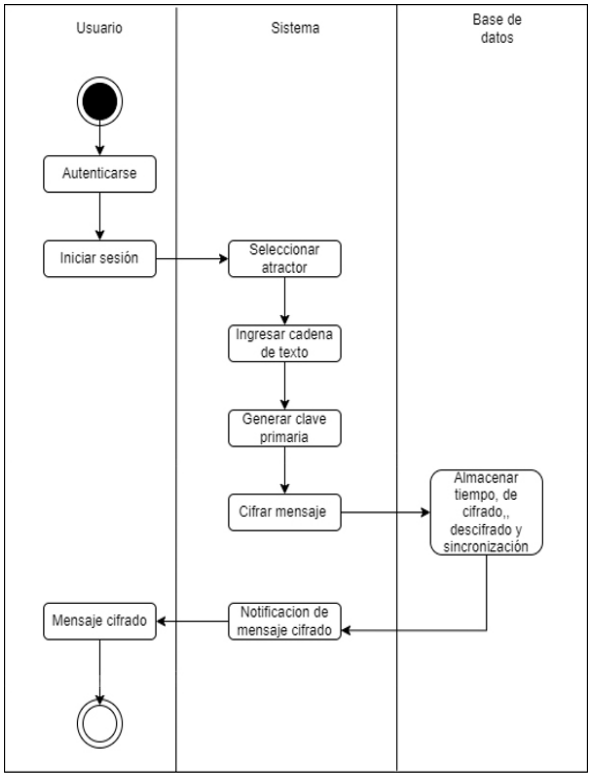

Fig. 14. *Diagrama de actividades del cifrado.*

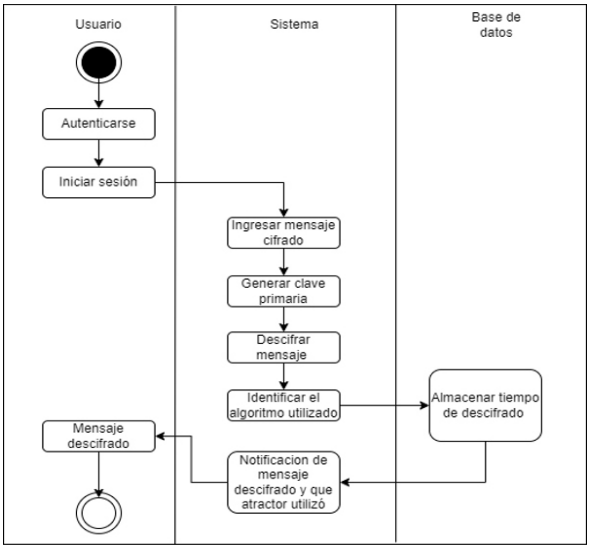

Fig. 15. *Diagrama de actividades del descifrado.*

4. Vista física

La siguiente Fig. 16. muestra el usuario que utiliza una computadora para ingresar al sistema, y solicitar las peticiones al servidor.

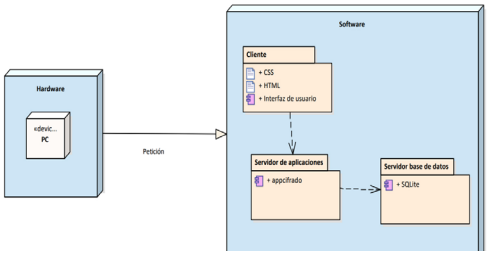

Fig. 16. *Diagrama de despliegue.*

5. Vista de escenario

Para la vista de escenario se muestra el diagrama de caso de uso, ver Fig. 17.

Este diagrama permite observar el funcionamiento del sistema completo con las interacciones de los usuarios y administradores.

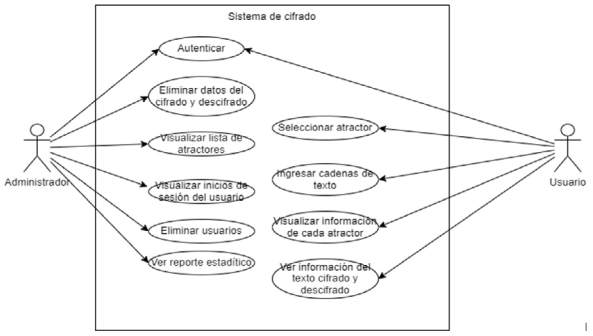

Fig. 17. *Diagrama de caso de uso*

F. Pruebas

En la Fig. 18 se muestra el menú del cifrado para que el usuario pueda seleccionar el atractor con el cual desea trabajar.

Fig. 18. *Aplicación web, menú cifrado.*

Para el descifrado se creó una sola entrada de texto cifrado, e internamente se identifica el atractor utilizado y se muestra en pantalla. Ver Fig. 19.

Fig. 19. *Aplicación web, descifrado*

IV. Análisis e interpretación de resultados

Se presenta los resultados obtenidos con el desarrollo de la aplicación web para cifrar y descifrar cadenas de texto mediante la selección de un atractor caótico. Estos resultados se obtuvieron mediante técnicas de medición del comportamiento temporal, y la confidencialidad, según los resultados obtenidos se aplicó test no paramétrico de Kruskal Wallis y diagramas de dispersión respectivamente.

A. Evaluación del comportamiento temporal

2. Proceso cifrado

Tiempo de sincronización

En la Tabla II se muestran los resultados de las 25 muestras por cada atractor, obteniendo el promedio de tiempo de sincronización en microsegundos.

Atractor	Promedio del tiempo de sincronización.
Chen	213428,44 µs
Lorenz	34650,96 µs
Rossler	314535,92 µs
Sprott	74541,6 µs

Para tener una mejor visualización de los resultados se muestra en la Fig. 20. barras generado desde la aplicación web de cifrado y descifrado.

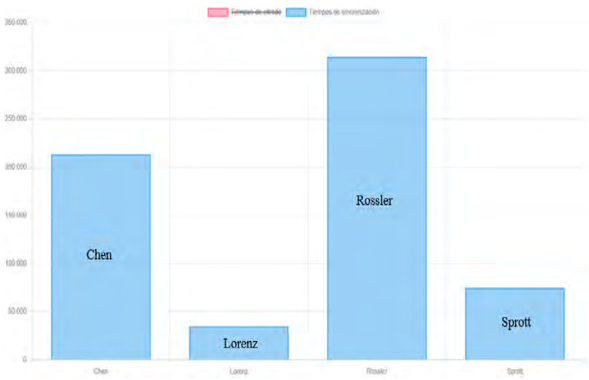

Fig. 20. *Barras del tiempo de sincronización.*

Se demuestra que el atractor de Lorenz es el más rápido en realizar la sincronización y el atractor de Rossler es el que tarda más.

Evaluación de supuestos del modelo

Se realiza una evaluación de supuestos del modelo para determinar si es factible realizar un análisis de varianza, o de lo contrario utilizar una prueba no paramétrica.

Para los análisis se utiliza un nivel de significancia de 0.05.

Igualdad de varianzas

Se utilizó levenetest para observar si existe una igualdad en las varianzas. Ver Fig. 21.

```
> leveneTest(tiempo_sincr ~ atractor_id, cifradotabla3)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value   Pr(>F)
group  3   4.791 0.003736 **
      96
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Fig. 21. *Test de levene para el tiempo de sincronización*

Dado que el p-valor (0.003736) es significativamente menor que 0.05, se puede rechazar la hipótesis nula (H0) de igualdad de varianzas entre los atractores. Esto significa que existe evidencia suficiente para afirmar que las varianzas de los tiempos de sincronización son diferentes entre al menos dos de los grupos definidos por el atractor_id.

Normalidad de los residuos del modelo

En la prueba de normalidad de Lilliefors (Kolmogorov-Smirnov) se obtuvo lo siguiente ver Fig. 22.

```
> lillie.test(standard_res)

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  standard_res
D = 0.31297, p-value < 2.2e-16
```

Fig. 22. *Prueba de normalidad de Lilliefors del tiempo de sincronización.*

La hipótesis nula (H0) de la prueba de Lilliefors es que los datos siguen una distribución normal. La hipótesis alternativa (H1) es que los datos no siguen una distribución normal.

Dado que el p-valor es extremadamente pequeño (< 2.2e-16) y es menor que el nivel de significancia de 0.05, se puede concluir que hay evidencia suficiente para rechazar la hipótesis nula (H0). Es decir, los datos de la variable "standard_res" no siguen una distribución normal.

Según los resultados observados se determinó que es necesario realizar una prueba no paramétrica, en este caso se utiliza la prueba de Kruskal Wallis. Fig. 24.

```
> kruskal.test(tiempo_sincr ~ atractor_id, cifradotabla3)

        Kruskal-Wallis rank sum test

data:  tiempo_sincr by atractor_id
Kruskal-Wallis chi-squared = 43.833, df = 3, p-value = 1.638e-09
```

Fig. 23. *Test de Kruskall Wallis para el tiempo de sincronización*

La hipótesis nula (H0) de la prueba de Kruskal-Wallis es que no hay diferencias entre las medianas de los grupos definidos por el atractor_id. La hipótesis alternativa (H1) es que al menos una mediana es diferente.

Dado que el valor p obtenido en la prueba (1.638e-09) es mucho menor que el nivel de significancia de 0.05, se puede concluir que hay evidencia suficiente para rechazar la hipótesis nula (H0). Esto significa que al menos una de las medianas del tiempo de sincronización en microsegundos difiere significativamente entre los grupos definidos por el atractor_id.

Tiempo de cifrado

En la Tabla III se muestran los resultados de las 25 muestras por cada atractor, obteniendo el promedio de tiempo de cifrado en microsegundos.

Tabla 2.
Promedios del tiempo de cifrado

Atractor	Promedio del tiempo de cifrado.
Chen	200683,48 µs
Lorenz	46287,52 µs
Rossler	349716,92 µs
Sprott	74817,28 µs

Para tener una mejor visualización de los resultados se muestra un gráfico de barras generado desde la aplicación web. Fig. 24.

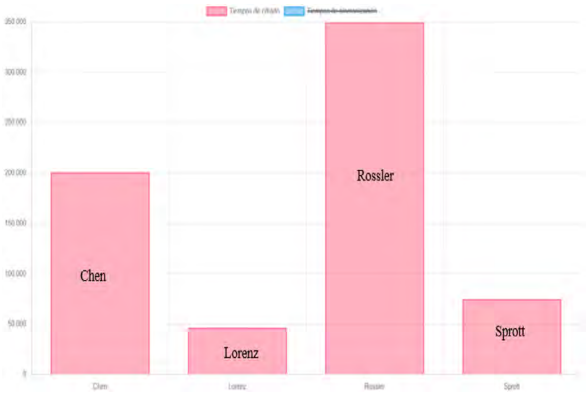

Fig. 24. *Barras del tiempo de cifrado.*

Se demuestra que el atractor de Lorenz es el más rápido para realizar el cifrado de cadenas de texto y Rossler es el más lento.

Evaluación de supuestos del modelo

Se realiza una evaluación de supuestos del modelo para determinar si es factible realizar un análisis de varianza, o de lo contrario utilizar una prueba no paramétrica.

Para los análisis se utiliza un nivel de significancia de 0.05.

Igualdad de varianzas

Se utilizó levenetest para observar si existe una igualdad en las varianzas. Ver Fig. 25.

```
> leveneTest(tiempo_cif ~ atractor_id, cifradotabla3)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value   Pr(>F)
group  3  4.4282 0.005836 **
      96
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Fig. 25. *Test de levene para el tiempo de cifrado.*

Los resultados del test de Levene indican que existe una diferencia significativa en la variabilidad del tiempo de cifrado entre los diferentes grupos de atractores. El valor p (0.005836) es menor que el nivel de significancia establecido (0.05), lo que significa que hay una fuerte evidencia para rechazar la hipótesis nula de que las varianzas de los tiempos de cifrado son iguales para todos los atractores.

Normalidad de los residuos del modelo

En la prueba de normalidad de Lilliefors (Kolmogorov-Smirnov) se obtuvo lo siguiente. Fig. 26.

```
> lillie.test(standard_res1)

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  standard_res1
D = 0.23238, p-value = 1.442e-14
```

Fig. 26. *Test de normalidad para el tiempo de cifrado*

El valor p en esta prueba es extremadamente bajo (1.442e-14, es decir, 0.00000000000001442). Un valor p bajo sugiere que se debe rechazar la hipótesis nula. En el caso de la prueba de Lilliefors, la hipótesis nula es que los datos siguen una distribución normal.

Según los resultados observados se determinó que es necesario realizar una prueba no paramétrica, en este caso se utiliza la prueba de Kruskal Wallis. Fig. 27.

```
> kruskal.test(tiempo_cif ~ atractor_id, cifradotabla3)

        Kruskal-Wallis rank sum test

data:  tiempo_cif by atractor_id
Kruskal-Wallis chi-squared = 39.107, df = 3, p-value = 1.647e-08
```

Fig. 27. *Test de Kruskal Wallis para el tiempo de cifrado.*

Dado este valor p extremadamente bajo, hay fuertes evidencias para rechazar la hipótesis de que los tiempos de cifrado son iguales para todos los atractores. Esto sugiere que al menos un atractor tiene un tiempo de cifrado mediano significativamente diferente a los demás.

2. Proceso descifrado

En la Tabla IV se muestran los resultados de las 25 muestras por cada atractor, obteniendo el promedio de tiempo de descifrado en microsegundos.

Tabla 3.

Resultados del proceso de descifrado

Atractor	Promedio del tiempo de descifrado.
Chen	387290,32 µs
Lorenz	287799,44 µs
Rossler	379987,72 µs
Sprott	301363,72 µs

Para tener una mejor visualización de los resultados se muestra un gráfico de barras generado desde la aplicación web. Fig. 28.

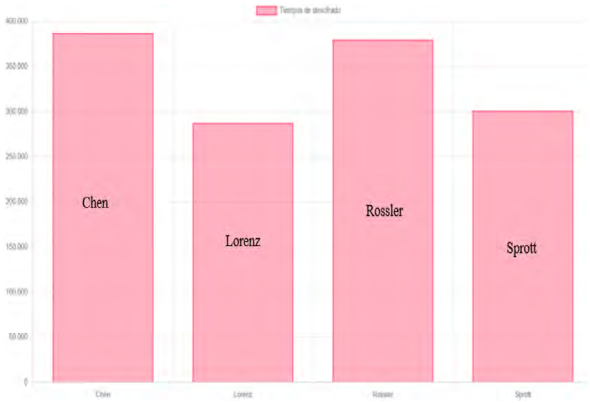

Fig. 28. *Barras del tiempo de descifrado.*

Se comprueba que existe una similitud entre los atractores, lo que indica que cualquier atractor es recomendado para realizar el descifrado.

Evaluación de supuestos del modelo

Se realiza una evaluación de supuestos del modelo para determinar si es factible realizar un análisis de varianza, o de lo contrario utilizar una prueba no paramétrica.

Para los análisis se utiliza un nivel de significancia de 0.05.

Igualdad de varianzas

Se utilizó levenetest para observar si existe una igualdad en las varianzas. Ver Fig. 29.

```
> leveneTest(tiempo_descif ~ atractord_id, descifradotabla3)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.8316 0.4797
```

Fig. 29. *Test de levene para el tiempo de descifrado*

Según la Prueba de Levene, no hay una diferencia significativa en la varianza del tiempo de descifrado entre los diferentes atractores en el conjunto de datos. Esto significa que la variabilidad del tiempo de descifrado es la misma para todos los atractores.

Normalidad de los residuos del modelo

En la prueba de normalidad de Lilliefors (Kolmogorov-Smirnov) se obtuvo lo siguiente Fig. 30.

```
> lillie.test(standard_res2)

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  standard_res2
D = 0.26212, p-value < 2.2e-16
```

Fig. 30. *Prueba de Lilliefors para el tiempo de descifrado.*

El valor p de esta prueba es extremadamente bajo, menos de 2.2e-16 (esto es prácticamente cero). Un valor p muy bajo sugiere que se puede rechazar la hipótesis nula. En este caso, la hipótesis nula es que los datos siguen una distribución normal.

Según los resultados observados se determinó que es necesario realizar una prueba no paramétrica, en este caso se utiliza la prueba de Kruskal Wallis. Fig. 31.

```
> kruskal.test(tiempo_descif ~ atractord_id, descifradotabla3)

        Kruskal-Wallis rank sum test

data:  tiempo_descif by atractord_id
Kruskal-Wallis chi-squared = 21.513, df = 3, p-value = 8.237e-05
```

Fig. 31. *Test de Kruskal Wallis para el tiempo de descifrado.*

Dado este valor p muy bajo (que es mucho menor que el nivel de significancia definido de 0.05), hay fuertes evidencias para rechazar la hipótesis de que los tiempos de descifrado son iguales para todos los atractores. Esto sugiere que al menos un atractor tiene un tiempo de descifrado mediano significativamente diferente a los demás.

3. Resultados finales

En conclusión, los resultados de los tests Kruskal-Wallis para el tiempo de sincronización, cifrado y descifrado demuestran que se debe rechazar la hipótesis nula (H0) en favor de la hipótesis alternativa (H1). Esto significa que al menos uno de los cuatro atractores difiere sustancialmente del resto en términos de tiempo de sincronización, cifrado y descifrado.

B. Evaluación de la confidencialidad

Los resultados que se obtuvieron se basaron en una cadena de texto de 118 caracteres.

En la Fig. 32 se muestra un gráfico de dispersión de la comparación de los valores del texto plano con el texto cifrado.

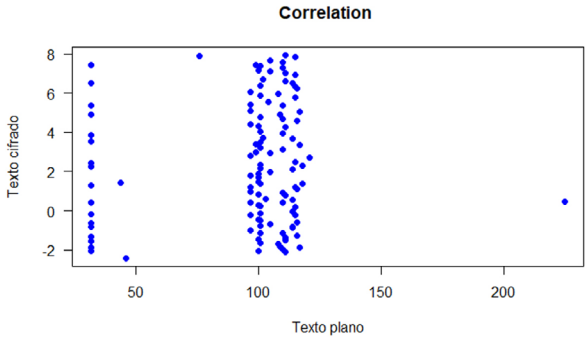

Fig. 32. *Correlación del texto plano con el cifrado del atractor Rossler.*

Se puede prestar atención que no tiene ninguna relación el texto plano y el texto cifrado, por lo que se determina que no existen patrones en los que se pueda lograr descifrar el mensaje sin conocer la llave.

El coeficiente de correlación de Pearson calculado fue de 0,08 con un valor p no significativo de 0,38, lo que sugiere que no hay evidencia suficiente para afirmar que existe una correlación lineal entre texto plano y texto cifrado.

A continuación, se muestra la Fig. 33 con la información del texto plano y el texto regenerado después del proceso de descifrado.

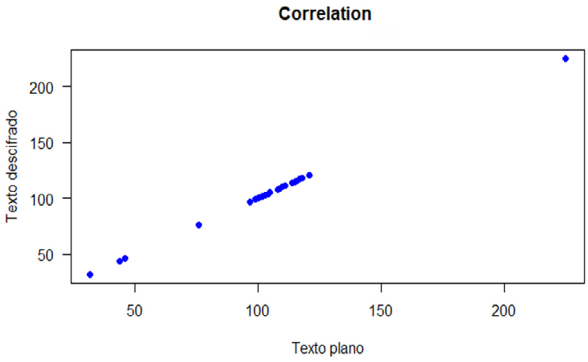

Fig. 33. *Correlación entre el texto plano y el regenerado después del descifrado del atractor Rossler.*

El texto se regenera por completo sin perder información después de descifrarse el mensaje. Se observó un coeficiente de correlación de 1, lo que indica que hay total similitud entre el mensaje original y el descifrado.

Se mantiene el mismo comportamiento en todos los atractores.

V. Conclusiones

A lo largo del estudio, se efectuó un análisis minucioso de cuatro sistemas caóticos esenciales: Rossler, Lorenz, Chen y Sprott. Cada sistema, con su atractor característico y comportamiento dinámico, desempeña un papel vital en la comprensión del caos. Desde la estructura en espiral del Rossler, pasando por el icónico atractor con forma de "mariposa" de Lorenz, hasta las complejidades distintivas de Chen y Sprott, estos sistemas se han destacado por su naturaleza impredecible y altamente sensible a las condiciones iniciales. Esta singularidad y complejidad los convierten en herramientas poderosas, subrayando su significado y aplicabilidad en el mundo del cifrado y descifrado. La revisión de estos sistemas no solo ha permitido un entendimiento profundo del caos, sino que también ha establecido un fundamento sólido para futuras aplicaciones y estudios en el campo de la seguridad cibernética.

Más allá de una mera revisión teórica, el estudio se adentró en la práctica, llevando a cabo un proceso de sincronización caótica para cada uno de estos atractores. Los resultados revelaron detalles interesantes y esenciales para la implementación práctica de estos sistemas en la ciberseguridad. A través del uso del test Kruskal-Wallis, se pudo determinar diferencias significativas en los tiempos de sincronización entre los atractores. El atractor de Lorenz, con su complejo comportamiento y estructura, emergió como el más eficiente en términos de sincronización, lo que sugiere su gran potencial en aplicaciones de cifrado. Estos hallazgos no solo fortalecen la comprensión del comportamiento caótico, sino que también guían futuras investigaciones y aplicaciones en áreas críticas como la seguridad cibernética.

El desarrollo de la aplicación web representó un desafío técnico y metodológico, que se abordó con éxito mediante el empleo de SCRUMBAN como metodología de trabajo. SCRUMBAN combina elementos de SCRUM y Kanban, dos enfoques ágiles que promueven la flexibilidad, adaptabilidad y eficiencia en la producción.

Se llevó a cabo una evaluación rigurosa de la eficiencia y seguridad en el cifrado y descifrado en la aplicación desarrollada. Los análisis de tiempo revelaron diferencias significativas entre los atractores, subrayando la necesidad de equilibrar la velocidad y la seguridad en la elección de los métodos de cifrado. Quedando como el más eficiente en el cifrado y descifrado el atractor de Lorenz.

VI. Agradecimientos

Quisiera expresar mi más profundo agradecimiento al grupo de investigación GrIISoft de la Facultad de Informática y Electrónica (FIE) de la Escuela Superior Politécnica de Chimborazo (ESPOCH), por su invaluable apoyo y colaboración en el desarrollo del proyecto titulado "Enfoque de cifrado de objetos JSON utilizando sincronización caótica a partir del análisis de un conjunto de atractores".

La oportunidad de trabajar junto a un equipo tan experimentado y dedicado ha sido una experiencia enriquecedora y fundamental para el éxito de esta investigación. La orientación, el conocimiento y los recursos proporcionados por GrIISoft han sido elementos clave en la realización de este estudio, permitiéndome explorar nuevas fronteras en el campo del cifrado y la seguridad de la información.

VII. Referencias

[1] J. Cordova Ramirez, H. Vega Huerta, C. Rodriguez Rodriguez, y F. Escobedo Bailón, «Firma digital basada en criptografía asimétrica para generación de historial clínico», 3C Tecnol. Innov. Apl. Pyme, pp. 65-85, dic. 2020, doi: 10.17993/3ctecno/2020.v9n4e36.65-85.

[2] A. Sheikholeslam, «A chaos based encryption method using dynamical systems with strange attractors»:, en Proceedings of the International Conference on Security and Cryptography, Milan, Italy: SciTePress - Science and and Technology Publications, 2009, pp. 259-265. doi: 10.5220/0002105402590265.

[3] O. S. Gómez, R. Rosero Miranda, J. Estrada-Gutiérrez, y M. Jiménez-Rodríguez, «An Approach for Securing JSON Objects through Chaotic Synchronization», Cybern. Inf. Technol., vol. 22, pp. 23-34, dic. 2022, doi: 10.2478/cait-2022-0037.

[4] C. E. R. Montalván, «Desarrollo de un mecanismo de cifrado basado en el algoritmo criptográfico simétrico aes», p. 139, 2019.

[5] R. Ribero Medina y M. Ramirez Gómez, «Caos: Definición, Detección y Ejemplos», 1992, doi: https://revistas.uniandes.edu.co/doi/pdf/10.13043/dys.30.7.

[6] O. Lombardi, «La teoría del caos y el problema del determinismo», p. 22, 2020.

[7] J. Zaqueros-Martínez, G. Rodríguez-Gómez, E. Tlelo-Cuatle, y F. Orihuela-Espina, «Sincronización de sistemas caóticos fraccionarios», p. 73, 2020.

[8] J. C. P. Campuzano, «Strange Attractors», Strange Attractors. Accedido: 12 de enero de 2023. [En línea]. Disponible en: https://jcponce.github.io/

[9] E. Pacheco Cruz, «Atarctor de Lorenz y Rossler | PDF | Teoría del caos | Atractor», Scribd. Accedido: 30 de diciembre de 2022. [En línea]. Disponible en: https://es.scribd.com/document/398824115/Atarctor-de-Lorenz-y-Rossler

[10] C. A. Ibanez, «Identificacio´n del sistema de Ro¨ssler: enfoque algebraico y algoritmos gene´ticos», 2005.

[11] G. Chen y T. Ueta, «Yet Another Chaotic Attractor», Int. J. Bifurc. Chaos - IJBC, vol. 9, pp. 1465-1466, jul. 1999, doi: 10.1142/S0218127499001024.

[12] G. Paredes, «Los Flujos Caóticos Más Simples (FCMS) Un mito entre lo complejo y lo complicado». 2023. [En línea]. Disponible en: http://casanchi.org/mat/flujoscaoticos01.pdf

[13] Q. Lai y S. Chen, «Generating Multiple Chaotic Attractors from Sprott B System», Int. J. Bifurc. Chaos, vol. 26, p. 1650177, oct. 2016, doi: 10.1142/S0218127416501777.