Encriptación de audio caótica para comunicación PCM

Cristofer Martinez Garcia[a], Irving Alonso Mancillas Zamudio[b], M.C. Marco Antonio Pinto Ramos[c], Dr. Diego Armando Trujillo Toledo[d].

Facultad de Ciencias Químicas e Ingeniería. Universidad Autónoma de Baja California. C.P.22390 ·. Tijuana, B. C., México.
[a]cristofer.martinez@uabc.edu.mx
[b]irving.mancillas@uabc.edu.mx
[c]mpinto@uabc.edu.mx
[d]dtrujillotoledo@uabc.edu.mx

Resumen

La encriptación de audio caótica para áreas de comunicación como podría ser el método de Modulación por Codificación de Pulsos (PCM) tiene como objetivo desarrollar un sistema seguro de transmisión digital de audio. Mediante PCM y los mapas caóticos, el sistema garantiza la privacidad y seguridad de la información transmitida. Constando con cinco etapas totales; adquisición de la señal, encriptación caótica, modulación PCM, transmisión y la recepción-desencriptación.

La implementación del sistema se desarrolló mediante Matlab y Arduino, la cuál demostró que el método es altamente seguro y sensible a pequeñas variaciones en la clave. Sin embargo, se determinó áreas de oportunidad en aspectos como la sincronización entre el transmisor y el receptor, depuración de la eficiencia y estabilidad de la transmisión. Este trabajo es relevante en aplicaciones de telecomunicaciones y seguridad de datos, donde la protección de la información es prioritaria

Palabras clave—Arduino, Caos, Criptosistema, Encriptación, PCM.

Abstract

Chaotic audio encryption for communication areas, such as the Pulse Code Modulation (PCM) method, aims to develop a secure digital audio transmission system. By utilizing PCM and chaotic maps, the system ensures the privacy and security of the transmitted information. It consists of five main stages: signal acquisition, chaotic encryption, PCM modulation, transmission, and reception-decryption.

The implementation of the system in Matlab and Arduino demonstrated that the method is highly secure and sensitive to small variations in the key. However, areas of opportunity were identified in aspects such as synchronization between the transmitter and receiver, efficiency optimization, and transmission stability.

This work is highly relevant in telecommunications and data security applications, where information protection is a priority.

Keywords—Arduino, Chaos, Cryptosystem, Encryption, PCM.

1. Introducción

En la era actual de las comunicaciones digitales, la seguridad y la confidencialidad se han convertido en aspectos cruciales para garantizar la integridad de los datos transmitidos [1]. Como aplicación para comunicaciones complejas donde los problemas de seguridad violan los derechos de privacidad de las personas, este proyecto se centra el desarrollo de un sistema de transmisión digital seguro, diseñado para convertir señales analógicas a digitales utilizando la modulación por codificación de pulsos (PCM) agregando técnicas de encriptación caótica para su transmisión, de forma que garantice el contenido de la señal transmitida sea ilegible sin la clave de encriptación adecuada, incrementando así la seguridad y confidencialidad en el proceso de comunicación. Asegurando que la información permanezca protegida contra accesos no autorizados [2].

La implementación de este sistema busca no solo salvaguardar el contenido transmitido, haciéndolo ilegible para usuarios no autorizados, sino también demostrar la integración de tecnologías de conversión y encriptación en un único flujo de comunicación. Este enfoque resulta especialmente relevante en aplicaciones donde la seguridad de los datos es crítica, como en sistemas de telecomunicaciones, transmisión de audio y control remoto o sistemas de monitoreo a tiempo real [3].

2. Antecedentes

En el contexto de la comunicación digital y la necesidad de proteger la información, el uso de la Modulación por Código

de Pulsos ha sido una técnica ampliamente utilizada para convertir señales analógicas a digitales. Además, la encriptación se ha convertido en una herramienta esencial para garantizar la privacidad y seguridad en la transmisión de datos. La combinación de PCM y encriptación es relevante en áreas como las telecomunicaciones, donde se necesita transmitir información digital de forma segura, evitando que sea interceptada o alterada sin autorización. Este proyecto se basa en esta necesidad, buscando generar un sistema de transmisión PCM encriptado, asegurando que la señal transmitida sólo pueda ser decodificada por el receptor autorizado.

La encriptación a través de mapas caóticos no es algo nuevo y sirven como referente para el propósito, un par de ejemplos son, el modelo caótico de Lorenz y el modelo de Chua que se han utilizado para encriptación debido a su comportamiento caótico. La señal original se combina con una señal caótica generada por las ecuaciones de Lorenz o Chua con ayuda de la operación XOR [4].

Ecuaciones de Lorenz:

$$\frac{dx}{dt} = \sigma(y - x)\rho\beta \qquad \text{ec.(1)}$$

$$\frac{dy}{dt} = \rho x - y - xz \qquad \text{ec.(2)}$$

$$\frac{dz}{dt} = xy - \beta z \qquad \text{ec.(3)}$$

Siendo,

σ: Número de Prandtl 1.
ρ: Número Rayleigh 2.
β: Razón entre la longitud y altura del sistema.
x: Velocidad y la dirección de circulación del fluido.
y: Variación de temperatura vertical.
z: Desviación del gradiente vertical de temperatura de la linealidad

Estas ecuaciones diferenciales nos describen el comportamiento de un sistema caótico, llamado “atractor de Lorenz”, el cual está formado por las ecuaciones 1, 2 y 3, es el conjunto de ecuaciones no lineales que tienen la característica de depender completamente de sus valores iniciales para generar un comportamiento específico. Una diferencia del orden de 10^{-8} en las condiciones iniciales es suficiente para que las trayectorias empiecen a cambiar notablemente, lo que otorga el comportamiento caótico que se requiere [5].

El amplio desarrollo de esta área presenta un enfoque innovador como es el caso de diseños de controladores para sincronización de circuitos de Chua con diferentes parámetros. De esta forma dos sistemas no idénticos logran un estado sincronizado, fundamental para la implementación de sistemas de comunicación caóticos [5].

Ecuaciones de Chua:

$$C_1 \frac{dV_1}{dt} = \frac{1}{R_v}(V_2 - V_1) - g(V_1) \qquad \text{ec.(4)}$$

$$C_2 \frac{dV_2}{dt} = \frac{1}{R}(V_1 - V_2) - I_L \qquad \text{ec.(5)}$$

$$L \frac{dI_L}{dt} = -V_2 \qquad \text{ec.(6)}$$

Siendo,

C_1: Capacitancia del primer capacitor en el circuito.
C_2: Capacitancia del segundo capacitor en el circuito.
L: Inductancia del inductor en el circuito.
R: Resistencia que conecta los dos nodos del circuito.
Rv: Resistencia variable asociada al primer nodo.
V_1: Voltaje en el primer nodo del circuito.
V_2: Voltaje en el segundo nodo del circuito.
I_L: Corriente a través del inductor.
$g(V_1)$: Representando al diodo de Chua a través de la respuesta de un elemento activo como un amplificador operacional con retroalimentación negativa (característica no lineal del circuito).

La modulación por código de pulso (PCM)**,** desarrollado en 1937 por AT&T y atribuida a Alex H. Reeves [6], es una técnica de conversión de señales analógicas a digitales, mediante la cuantificación de los valores discretizados de la señal que se codifica en formato binario, este se establece como el estándar en redes telefónicas digitales. Por ello, la estrategia para la adquisición del mensaje es convertir señales analógicas en digitales mediante muestreo, cuantificación y

codificación, utilizando un convertidor analógico-digital (ADC) para la transmisión y un convertidor digital-analógico (DAC) en la recepción [6], el requisito fundamental requiere que la frecuencia de muestreo cumpla con el teorema de Nyquist, descrito en la ecuación 7 y que la resolución de ADC permita la adecuada representación de la señal, de forma estandarizada la modulación hace uso de 8 bits.

$$f_s \geq 2f_{max} \qquad \text{ec.(7)}$$

De esta forma se obtiene el mensaje a transmitir, se procesa, se recibe y se interpreta de forma segura.

3. Etapas del criptosistema emisor

Este sistema consta de varias etapas:

1. Captura de la Señal.
2. Encriptación.
3. Modulación PCM.
4. Transmisión.

1. Captura de Señal Analógica: Se toma una muestra de una señal de audio de medio segundo y se muestrea a una frecuencia específica de 4000 muestras por segundo. La frecuencia fundamental de la voz humana ronda aproximadamente los 120 Hz a los 210 Hz en promedio, sin embargo, gracias a los armónicos el ancho de banda llega a extenderse a los 4000 Hz.

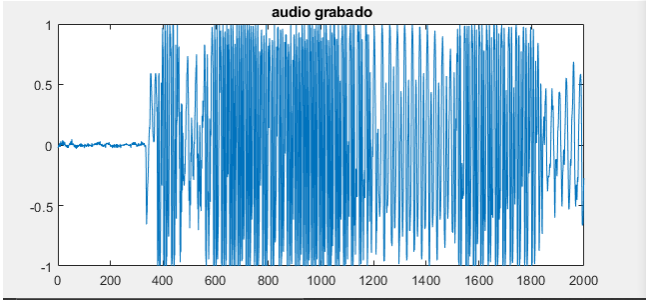
En este caso no se satisface el teorema de Nyquist, dado en la ecuación 7, debido a que solo se estará considerando hasta 2000 Hz adecuadamente, que representa el 50% del ancho de banda extendido, se establecen los 4000 Hz de muestreo para capturar los datos de un audio legible evitando las frecuencias mayores de 2000 Hz.

Para la adquisición de voz, como datos análogos se utilizó un software de matemáticas, en este caso particular es Matlab. Se desarrolló un programa para ejecutar el micrófono interno del dispositivo durante 2 segundos. Con una frecuencia de muestreo de 4000 Hz se generan 2000 datos, los cuales se guardarán en la variable Y. La figura 1, muestra el diagrama de flujo del programa para capturar la señal de audio y la figura 2, muestra la señal de audio capturada.

Fig 1. Diagrama de flujo del programa para capturar la señal.

Elaboración propia

Fig 2. Señal de audio capturada..



Elaboración propia

2. Encriptación del Mensaje: Una vez obtenido el mensaje se genera un mapa caótico basado en el mapa Chua para generar una señal portadora que se utilizará para ocultar la información del audio en su transmisión.

Se genera una secuencia caótica en base a una contraseña que ingresa el usuario, estos números se utilizarán como coeficientes del mapa caótico. El mapa caótico del circuito chua cuenta con parámetros críticos que generan el comportamiento caótico a valores específicos.

El diodo de Chua representa un comportamiento no lineal, nombrado como 'g(V_1)', se determina un rango de voltaje no lineal que lo describa, para el resto de los parámetros se consideran constantes, como se muestra en la Tabla I. Los valores fueron propuestos en [7].

Tabla I: Valores para los componentes del circuito Chua.

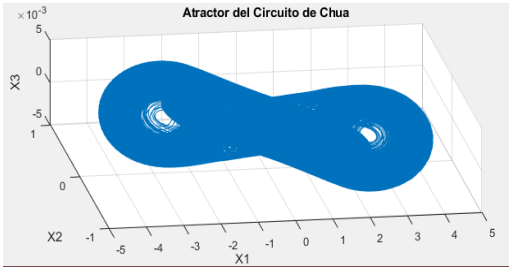
Parámetro [Unidad]	**Valor**	**Comentario**
C_1 [F]	10^{-9}	
C_2 [F]	100^{-9}	
Rv [Ω]	1000 a 1900	Se vuelve rango, debido a que es el parámetro clave de la contraseña
L[H]	18^{-3}	
g(V_1) [V]	-9 a 9	Rango de voltaje por el comportamiento no lineal

Elaboración propia

La elección del valor de dicha contraseña se determinó de manera experimental como un rango de valores que nos asegura un mapa caótico.

La generación de dicha secuencia caótica consiste en resolver las ecuaciones 4, 5 y 6, las cuales tienen una alta sensibilidad a variaciones pequeñas, esta secuencia debe mantener el mismo tamaño del mensaje, el cual es de 2000 datos. En la figura 3, se muestra la generación del mapa caótico de Chua con 2000 datos.

Figura 3. Generación de la secuencia caótica

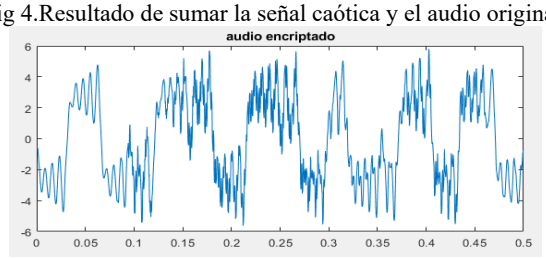


Elaboración propia

La encriptación consiste en tomar los datos de la secuencia caótica y los datos del audio y combinarlos, provocando una distorsión del audio.

La figura 4 muestra el resultado de sumar la secuencia caótica y el audio original.

Fig 4.Resultado de sumar la señal caótica y el audio original.



Elaboración propia

3. Modulación PCM: La señal ya encriptada se muestrea, se cuantifica y convierte en un arreglo de Bits, donde cada muestra representa un valor en formato digital (PCM). Se utiliza una profundidad de 8 Bits por muestra, lo cúal es suficiente para representar la señal con buena precisión.

Para generar la modulación se utilizan las herramientas internas de matlab, que convierte valores numéricos a un equivalente binario similar al formato PCM. Solo requiere normalizar el dato, adquiriendo el valor máximo de los datos que se van a enviar y dividiéndolo entre cada uno para poder representar todo el rango de valores en la forma correspondiente de 0 a 255, además se le suma 125 para eliminar el signo y tenerlo en codificación PCM. Este proceso se muestra en las ecuaciones 8 y 9.

Normalización a 1 con signo:

$$Y_{norm} = max(\frac{yc}{yc}) \qquad \text{ec.(8)}$$

Eliminación del signo y se ajustan los valores a 250 unidades para no tener decimales.

$$Y_{norm\ sin\ signo} = (Y_{norm} * 125) + 125) \quad \text{ec.(9)}$$

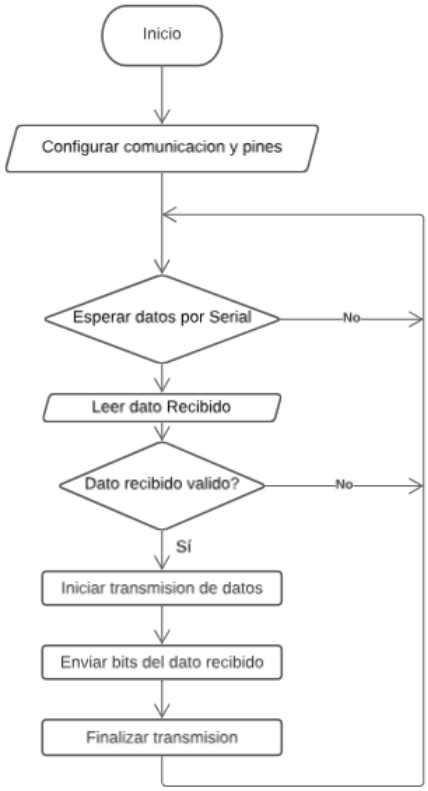
4. Transmisión: La señal encriptada se puede transmitir por cualquier canal de comunicación, para esta prueba se conectan dos dispositivos arduinos mediante un cable a través del pin digital, uno como emisor y otro como receptor.

En esta etapa se paso el proceso de matlab al arduino, utilizando el comando 'Write()' se envio un byte por el comando serial que conecta al arduino con la computadora, de esta forma enviará Byte por Byte del vector de valores de PCM.

En el arduino, se debe ejecutar una programación, que consiste en leer dato y mandarlo, para ello, se habilita el puerto serial del arduino para su conexión a la computadora, utilizando una tasa estándar 9600 bit por segundo, de ahí solo se especifica el orden de envío de bit en base a la lectura del puerto serial, al momento de que el arduino adquiere el dato de la computadora envía una secuencia de sincronización (1-0-0) por el pin digital de salida para luego enviar el arreglo de byte que leyó del puerto serial. Se encontró que la comunicación arduino-computadora tarda 10 ms y el tiempo de un bit enviado por el pin digital de 3 us, en total por cada byte son 31 us sin considerar los 10 ms del arduino-computadora.

En la figura 5, se muestra el diagrama de flujo para enviar la información.

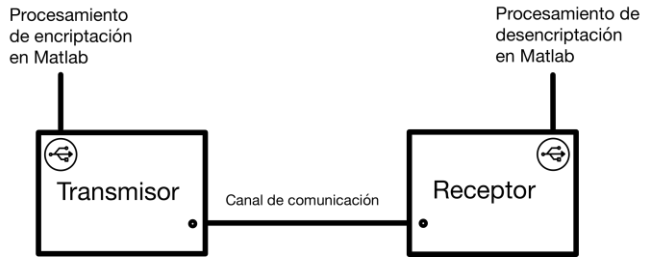
Fig 5. Diagrama de flujo de implementación de envío de datos.



Elaboración propia

En la figura 6, se muestra el diagrama de conexión realizada entre los dos dispositivos arduinos mediante el pin digital.

Fig 6. Conexiones de comunicación de arduino.



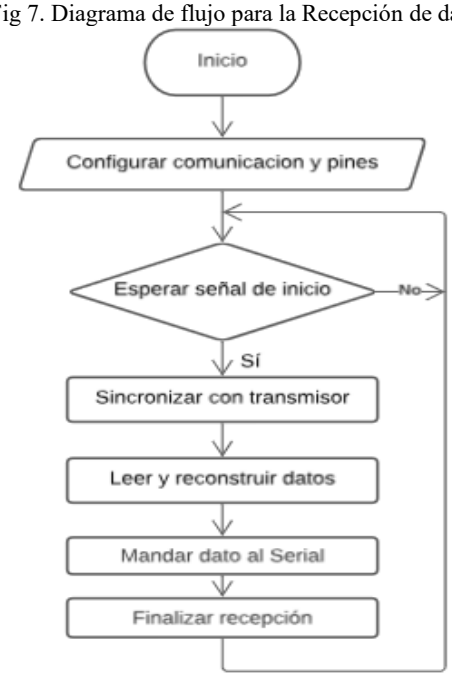
Elaboración propia

4. Etapas del criptosistema receptor.

1. Recepción: El dispositivo arduino en función de receptor, recibe valores de 8 bits en un tiempo aproximado de 3 µs por bit y con 7µs de sincronización por cada byte dando un total de 31µs aproximadamente de transmisión de cada valor discreto procesado y recuperado. Este valor discreto se envia

por el puerto serial a Matlab para su acoplamiento y decodificación. En la figura 7, se muestra el diagrama de flujo en el dispositivo arduino para la recepción de datos.

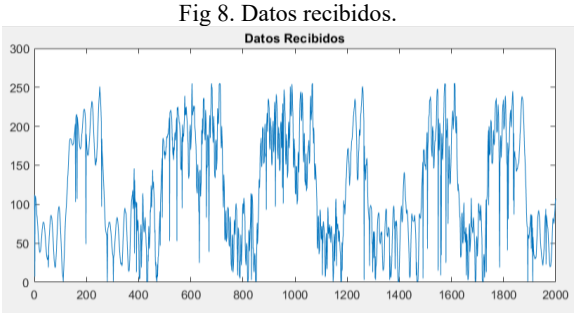
Fig 7. Diagrama de flujo para la Recepción de datos.



Elaboración propia

En la figura 8, se muestran los datos recibidos.

Fig 8. Datos recibidos.

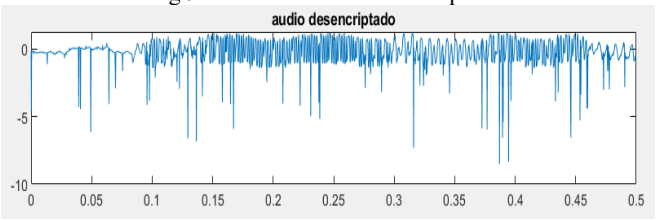


Elaboración propia

2. Acoplamiento y Decodificación: Ya con los valores discretos en matlab de 0 a 255 se normaliza la señal para así acoplar a las dimensiones previas a la modulación PCM.

3. Desencriptación: Una vez igualada la señal recibida encriptada, se genera el mapa caótico de Chua con la misma clave y se resta a la señal acoplada para anular la portadora, dando como resultado la información del audio original, como se muestra en la figura 9.

Fig 9. Señal resultante desencriptada.



Elaboración propia

5. Interfaz de Usuario

Se diseñó una interfaz del sistema que permite al usuario ajustar parámetros como la clave de encriptación, enviar sin encriptar, entre otras cosas.

Se presentarán dos casos de uso, donde el sistema realiza la transmisión y el caso del receptor.

Caso de transmisor

- **Especifica el dato a enviar:** pide al usuario que escoja que tipo de señal quiere enviar, ya sea el audio normal o enviar el audio encriptando.
- **Repetir el proceso:** pregunta al usuario si quiere repetir el proceso y enviar de nuevo o ya apagar el programa.

En la figura 10 se muestra la interfaz de usuario para el caso del transmisor.

Fig 10. Interfaz de usuario para transmisión.

```
quieres continuar? Y/N
=y
quieres enviar informacion encriptada? Y/N
=y
dame los valores de la clave desde 1000 a 1900
=1500

clave =

        1500

Iniciando la grabación...
Grabación finalizada.
Archivo recortado guardado como audio_grabado.wav
Reproduciendo el audio recortado...
```

Elaboración propia

Caso de emisor

- **Selección de señal recibida**: Preguntar si la señal recibida está encriptada y de ser así pedir la clave para desencriptarla correctamente.

En la Figura 11 se muestra la interfaz de usuario para el caso del receptor.

Fig 11. Interfaz de usuario para recepción.

```
Elije una opcion de recepcion de datos:
Y .- Con Caos (contraseña)
N .- Sin Caos (sin contraseña)
Ejemplo (Y), enseguida presiona "ENTER": y

Ingresa Contraseña (1000 - 1900): 1017
```

Elaboración propia

- **Visualización de resultados**: Despliegue de audio recibido y de audio desencriptado.

En la figura 12, se muestra el mensaje que envía el programa del receptor una vez que llega un mensaje.

Fig 12. Especificación de audio mostrado.

```
Reproduciendo Audio Caotico

Reproduciendo Audio Desencriptado
```

Elaboración propia

6. Resultados

Para la comprobación del sistema caótico se busco la desencriptación del mensaje con contraseñas con diferencia de 0.000001 unidades de diferencia

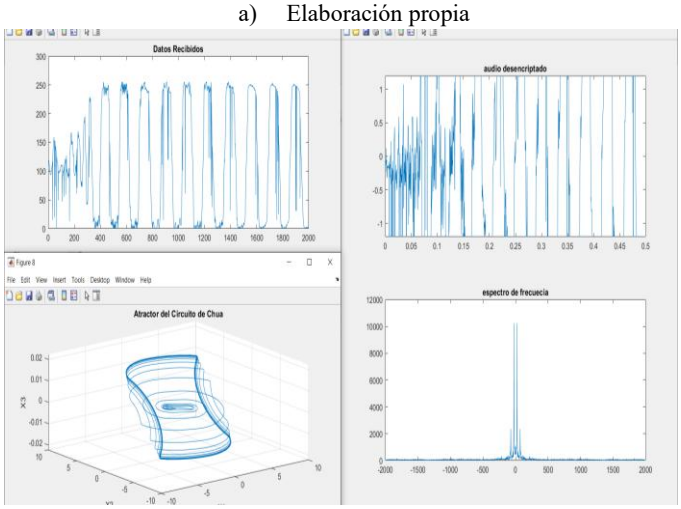
Ejemplo 1, clave incorrecta 1500.000001.

En la figura 13, se muestra como la interfaz exige una clave la cual será incorrecta por 0.000001, de tal forma que los datos desencriptados serán muy diferentes al real.

Fig 13. a) Interfaz y b) resultados para el ejemplo 1.

```
Elije una opcion de recepcion de datos:
Y .- Con Caos (contraseña)
N .- Sin Caos (sin contraseña)
Ejemplo (Y), enseguida presiona "ENTER": y

Ingresa Contraseña (1000 - 1900): 1500.000001

Reproduciendo Audio Caotico

Reproduciendo Audio Desencriptado

fx >>
```

a) Elaboración propia



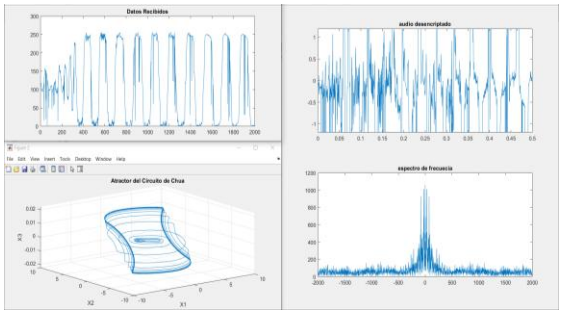
b) Elaboración propia

Ejemplo 2, clave incorrecta 1499.99999:
En la figura 14 se puede observar, que igual a la primera prueba con el mismo valor de error pero ahora restando a la clave original, el mensaje es distorsionado pero distinguimos una divergencia entre errores, demostrando también que aunque la magnitud del error sea el mismo la orientación juega un papel importante también en la producción del error.

Fig 14. a) Interfaz y b) resultados para el ejemplo 2.

```
Elije una opcion de recepcion de datos:
Y .- Con Caos (contraseña)
N .- Sin Caos (sin contraseña)
Ejemplo (Y), enseguida presiona "ENTER": y

Ingresa Contraseña (1000 - 1900): 1499.99999

Reproduciendo Audio Caotico

Reproduciendo Audio Desencriptado

fx >>
```

a) Elaboración propia



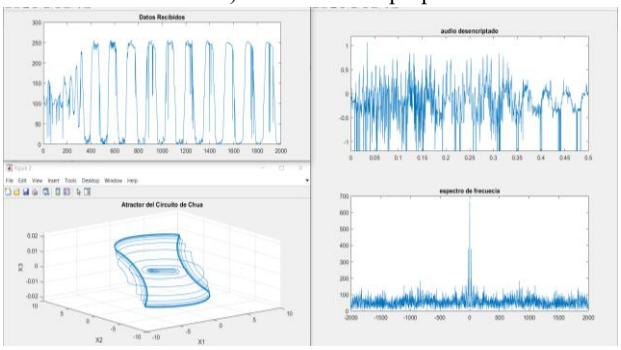
b) Elaboración propia

Ejemplo 3, clave correcta 1500:
En la figura 15, se observa la correcta forma de la señal capturada.

Fig 15. Interfaz y resultados para el ejemplo 3.

```
Elije una opcion de recepcion de datos:
Y .- Con Caos (contraseña)
N .- Sin Caos (sin contraseña)
Ejemplo (Y), enseguida presiona "ENTER": y

Ingresa Contraseña (1000 - 1900): 1500

Reproduciendo Audio Caotico

Reproduciendo Audio Desencriptado

fx >>
```

a) Elaboración propia



b) Elaboración propia

7. Conclusión

El sistema de encriptación desarrollado combina exitosamente la modulación por código de pulsos (PCM) y los principios del caos matemático para garantizar transmisiones de datos seguras y robustas. Esta integración demuestra un enfoque innovador al emplear plataformas accesibles como Arduino y Matlab para validar su funcionamiento, lo que facilita su implementación en entornos demostrativos y educativos.

Los resultados obtenidos resaltan la alta sensibilidad de la clave de encriptación, mostrando que incluso pequeñas variaciones generan una distorsión significativa en la señal desencriptada, lo cual refuerza la seguridad del sistema. Sin embargo, los experimentos también evidenciaron limitaciones relacionadas con la velocidad de procesamiento, la sincronización entre transmisor y receptor, y la robustez del canal de comunicación. Esto abre un panorama interesante para futuros desarrollos, en los que la incorporación de hardware especializado y algoritmos de encriptación más avanzados podría optimizar el rendimiento del sistema, tanto en velocidad como en precisión.

A nivel práctico, este trabajo sienta las bases para la implementación de sistemas de comunicación caótica en aplicaciones reales donde la privacidad de los datos sea prioritaria, como la seguridad en telecomunicaciones, monitoreo remoto o intercambio de información confidencial. A largo plazo, la exploración de canales alternativos como comunicaciones ópticas o inalámbricas podría ampliar aún más el potencial de esta tecnología.

En conclusión, este sistema representa una contribución significativa al campo de la seguridad en comunicaciones, integrando conceptos matemáticos avanzados, herramientas de ingeniería y tecnología digital para abordar la transmisión segura de la información.

Referencias:

[1] S. M. Mohamed, M. H. Yacoub, W. S. Sayed, L. A. Said, y A. G. Radwan, "Efficient hardware implementations of trigonometric functions and their application to sine-based modified logistic map," *Digital Signal Processing*, vol. 159, p. 104993, 2025. doi: 10.1016/j.dsp.2024.104993.

[2]V.-T. Pham, J. M. Munoz-Pacheco, A. Velichko, S. M. Boulaaras, and S. Momani, "A compact structure for triple-memristor maps with a hyperplane of fixed points," *Integration, the VLSI Journal*, vol. 101, p. 102334, 2025. DOI: 10.1016/j.vlsi.2024.102334.

[3] W. E. Hailu, R. B. Bellam, K. B. Prasad, S. T. J. L., R. Gowda, and S. Selvakumar, "A Secure Authentication Algorithm for Medical IoT using Steganography and Cryptography," *Journal of Machine and Computing*, vol. 5, no. 1, pp. 409-420, Jan. 2025. DOI: 10.53759/7669/jmc202505032.

[4] Barbará Morales, E., Alba Blanco, E. and Rodríguez Ramírez, O. (2012). Modulating electrocardiographic signals with chaotic algorithms. Ingeniería e Investigación, 32(2), 46–50. https://doi.org/10.15446/ing.investig.v32n2.31939

[5] H. Chen & Q. Ding, "A New Controller Controlling the Synchronous Communication between Different Chua's Circuits," *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, Bangkok, Thailand, 2009, pp. 57-61, doi: 10.1109/CAR.2009.77.

[6] Wayne Tomasi, *Sistemas de Comunicaciones Electrónicas*, 4ª ed. Phoenix, Arizona: Pearson Educ., 2003.

[7] Alcalá Martínez Minerva & Ángeles Garcia Francisco, *Circuito de Chua en la Sincronización de los Sistemas Caóticos,* Ipn.mx. Recuperado el 25 de marzo de 2025, de https://tesis.ipn.mx/bitstream/handle/123456789/12320/CIRCUITO%20DE%20CHUA%20EN%20LA%20SINCRONIZACION%20DE%20LOS%20SISTEMAS%20CAOTICOS.pdf?sequence=1&isAllowed=y