

Funcionalidad principal

1. Cifrado Hill 2x2

- ❖ Convierte letras a números ($A=0\dots Z=25$).
- ❖ Agrupa en bloques de tamaño 2.
- ❖ Multiplica cada bloque por la matriz clave K y aplica módulo 26.
- ❖ Si el último bloque no completa 2 letras, lo rellena con 'X'.

2. Validación de la matriz

- ❖ Calcula el determinante y el MCD con 26.
- ❖ Si el determinante **no es coprimo con 26**, muestra una advertencia: no se podría descifrar después, pero el cifrado aún se realiza.

3. Interfaz

- ❖ Campo de entrada de texto.
- ❖ Entrada para la matriz clave 2x2 (a, b, c, d).
- ❖ Botones **Cifrar** y **Limpiar**.
- ❖ Área de salida que muestra el texto cifrado y los **pasos detallados**.

Cifrado Hill (solo CIFRAR) - Matriz 2x2

Mensaje a cifrar:
hola

Salida (resultado y pasos):

```
=== Resultado ===
Texto original: hola
Matriz clave (2x2):
[[3, 3], [2, 5]]
Texto cifrado: LGHW

=== Pasos (detallado) ===
Texto normalizado (solo A-Z): HOLA
Números (A=0 ... Z=25): [7, 14, 11, 0]
Bloques (tamaño 2): [[7, 14], [11, 0]]
Multiplicación K * bloque (mod 26):
Bloque 1: [7, 14] -> K * bloque = [11, 6]
Bloque 2: [11, 0] -> K * bloque = [7, 22]
Resultado numérico final: [11, 6, 7, 22]
Texto cifrado: LGHW
```

Matriz clave 2x2 (enteros):

	a	b
a:	3	
b:	3	
c:	2	
d:	5	

CIFRAR

Limpiar

Determinante = 9 (mod 26 = 9).
GCD(det_mod26, 26) = 1.

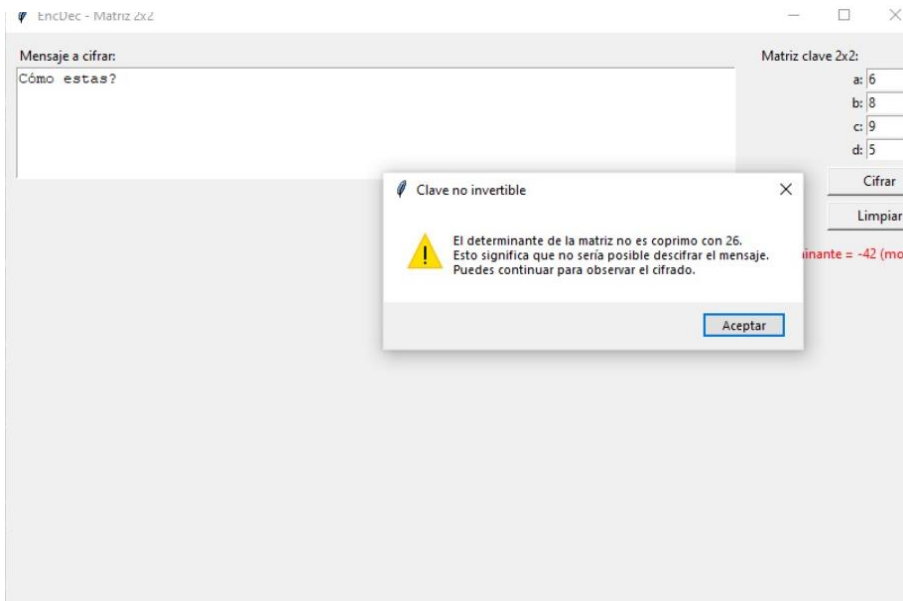
Funciones importantes

- `texto_a_numeros(texto)` → Convierte el texto a números.
- `numeros_a_texto(numeros)` → Convierte números a texto.
- `crear_bloques(numeros, n)` → Agrupa en bloques de tamaño n , rellena con 'X'.
- `determinante_2x2(K)` → Calcula determinante de la matriz 2×2 .
- `cifrar_mensaje(texto, K)` → Aplica el cifrado Hill y devuelve el resultado y los pasos.
- `leer_matriz_2x2()` → Lee los valores de los Entry y devuelve una matriz numpy 2×2 .
- `on_cifrar()` → Ejecuta todo el flujo de cifrado al presionar **Cifrar**.
- `on_limpiar()` → Limpia la interfaz.

The screenshot shows a web application window titled "EncDec - Matriz 2x2". The interface is divided into two main sections. On the left, under the heading "Mensaje a cifrar:", there is a large, empty text input field. On the right, under the heading "Matriz clave 2x2:", there are four input fields labeled "a:", "b:", "c:", and "d:". The values entered in these fields are 3, 3, 2, and 5, respectively. Below these input fields are two buttons: "Cifrar" and "Limpiar". A vertical scrollbar is visible on the right side of the interface.

Detalles interesantes

- Rellena con 'X' si el mensaje no completa bloques.
- Muestra **detalles paso a paso**, lo que es excelente para aprendizaje.
- Si la matriz clave no es invertible mod 26, muestra advertencia con color rojo.



Mejoras posibles

1. **Agregar descifrado** usando el inverso de la matriz módulo 26.
2. **Validar solo letras** en el texto de entrada, ignorando números y símbolos.
3. **Opcional:** permitir mensajes minúsculos sin convertirlos explícitamente.
4. **Mejorar la interfaz:** permitir matrices de mayor tamaño (3x3) para cifrados más complejos.
5. **Guardar o copiar** el resultado a portapapeles para facilidad.

Conclusión

En este avance del proyecto logramos implementar correctamente un **cifrador de matriz 2x2** con **interfaz gráfica en Tkinter**, cumpliendo los objetivos de convertir un texto en números, agruparlo en bloques y aplicar la transformación matricial módulo 26 para obtener el texto cifrado.

Se desarrollaron funciones claras para:

- Normalizar el texto y convertirlo a números.
- Validar la matriz clave, incluyendo el cálculo del determinante y su relación con 26, mostrando advertencias en caso de que no sea invertible.

La interfaz gráfica permite al usuario ingresar el texto y la matriz clave, ejecutar el cifrado y visualizar **paso a paso todo el proceso**, lo que facilita la comprensión del algoritmo y su funcionamiento.

Este avance demuestra la **correcta integración de la lógica de cifrado con la interacción gráfica**, estableciendo una base sólida para futuras mejoras, como la implementación de descifrado, soporte para matrices mayores o validaciones más avanzadas del texto de entrada.

Se consiguió un prototipo funcional que combina **educación, seguridad y usabilidad**, cumpliendo con los objetivos planteados para esta etapa del proyecto.

