UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CENTRO DE CIENCIAS BASICAS

SEGURIDAD EN SISTEMAS

PRIMEROS PASOS CIFRANDO

SEMESTRE 8° B

ALUMNA JUDITH YAHAIRA ORTEGA

ID: 334941

PROFESOR ARTURO OCAMPO SILVA

FECHA DE ENTREGA

20/02/2026

Índice

1. Introducción

Desde los albores de la civilización, la necesidad de proteger la información ha sido una constante en la historia humana. Gobiernos, militares y comerciantes han buscado formas de transmitir mensajes que solo el destinatario pudiera comprender. En ese contexto nacieron los primeros cifrados de sustitución: el Cifrado César y el Cifrado Atbash, dos métodos que, aunque simples desde la perspectiva actual, representaron un avance significativo en su época.

Sin embargo, la misma historia que vio nacer estos métodos también gestó su destrucción. El matemático y filósofo árabe Abu Yusuf Yaqub ibn Ishaq al-Kindi (أبو يوسف يعقوب بن إسحاق الكندي), conocido en Occidente como «Alkindus» y apodado el «Filósofo de los Árabes», vivió en Bagdad durante el siglo IX d.C. y redactó una obra que cambiaría para siempre la criptografía: Risalah fi Istikhraj al-Mu'amma (Manuscrito sobre el Descifrado de Mensajes Criptográficos), considerado el primer tratado de criptoanálisis de la historia.

La contribución de Al-Kindi al hackeo de cifrados simples

Al-Kindi observó que, en cualquier idioma natural, las letras no aparecen con la misma frecuencia. En español, por ejemplo, la letra 'e' es la más común, seguida de 'a', 'o' y 's'. Esta observación, aparentemente sencilla, tiene consecuencias devastadoras para los cifrados de sustitución simple:

- En el Cifrado César, cada letra del texto original siempre se transforma en la misma letra cifrada. Si en el texto original 'e' aparece el 20% de las veces, esa misma frecuencia aparecerá en el texto cifrado, pero asociada a otra letra.
- El atacante solo necesita analizar la frecuencia de aparición de cada carácter en el mensaje cifrado y comparar esa distribución con la del idioma natural. La letra más frecuente en el texto cifrado probablemente corresponda a 'e', lo que revela inmediatamente el desplazamiento utilizado.
- En Atbash, el análisis es incluso más directo, pues al ser un cifrado de espejo fijo, la distribución de frecuencias del texto cifrado es idéntica a la del original, solo con las letras intercambiadas.

El proceso que Al-Kindi sistematizó en el siglo IX se conoce hoy como Análisis de Frecuencia y constituye la base de múltiples técnicas modernas de criptoanálisis. Para aplicarlo sobre César o Atbash basta con:

- Contar cuántas veces aparece cada carácter en el texto cifrado.
- Ordenar los caracteres de mayor a menor frecuencia.
- Comparar esa distribución con la tabla de frecuencias del idioma del mensaje original.

- Deducir el desplazamiento (César) o confirmar la inversión (Atbash) y descifrar el texto completo.

¿Por qué ya no son viables César y Atbash como métodos de protección de datos?

Más allá del análisis de frecuencia, los avances tecnológicos han vuelto estos cifrados completamente inútiles en el contexto moderno:

- Fuerza bruta trivial: el Cifrado César solo tiene 25 variantes posibles. Una computadora puede probarlas todas en microsegundos sin necesidad de ningún análisis estadístico.
- Sin difusión ni confusión: los cifrados modernos (como AES-256) aplican principios de difusión —donde un solo bit de cambio afecta todo el mensaje— y confusión —haciendo la relación entre clave y texto cifrado extremadamente compleja—. César y Atbash no poseen ninguno de estos mecanismos.
- Tamaño de clave insignificante: AES-256 tiene 2^256 claves posibles; César tiene exactamente 25. La diferencia es astronómica e infranqueable.
- No resisten criptoanálisis moderno: herramientas automatizadas de análisis de frecuencia descifran un mensaje cifrado con César o Atbash en fracciones de segundo, independientemente de la longitud del texto.

En síntesis, Al-Kindi demostró hace más de doce siglos que la mera sustitución de caracteres no es suficiente para proteger la información, y la computación moderna ha convertido esa vulnerabilidad en un problema trivial. Hoy, César y Atbash solo tienen valor didáctico: sirven para comprender los fundamentos de la criptografía antes de abordar algoritmos seguros.

2. Objetivo

Desarrollar e implementar una aplicación web funcional que permita cifrar y descifrar mensajes mediante los métodos clásicos César y Atbash, utilizando como base la tabla de valores ASCII para el procesamiento de caracteres. La aplicación deberá:

1. Permitir al usuario seleccionar el módulo de cifrado a utilizar (César o Atbash), identificando claramente qué tipo de operación se está ejecutando en cada momento.
2. Aceptar un conjunto de caracteres personalizado proporcionado por el usuario, de modo que el cifrado opere sobre ese conjunto específico en lugar de un alfabeto fijo.

Adicionalmente, este trabajo busca comprender el contexto histórico del criptoanálisis, desde las aportaciones de Al-Kindi hasta la obsolescencia actual de los cifrados de sustitución simple, sentando las bases conceptuales necesarias para el estudio de algoritmos criptográficos modernos.

3. Desarrollo

3.1 Al-Kindi y el Análisis de Frecuencia

Abu Yusuf Yaqub ibn Ishaq al-Kindi (c. 801–873 d.C.) fue el primer enciclopedista árabe y uno de los intelectuales más prolíficos de la Casa de la Sabiduría de Bagdad. Escribió aproximadamente 260 tratados sobre matemáticas, filosofía, astronomía y medicina. Entre ellos, el Manuscrito sobre el Descifrado de Mensajes Criptográficos es el más relevante para la historia de la seguridad de la información.

En dicho manuscrito, Al-Kindi describió con precisión el principio que hoy denominamos análisis de frecuencia: la observación de que, en cualquier texto suficientemente largo, la distribución de letras sigue un patrón estadístico predecible propio del idioma en que fue escrito. Para el idioma árabe de su época, Al-Kindi calculó esas frecuencias y las usó como clave para romper mensajes cifrados. El mismo principio se aplica directamente al español o cualquier otro idioma.

El impacto de este descubrimiento sobre los cifrados de sustitución simple es total: ningún mensaje cifrado con César o Atbash puede considerarse seguro ante un analista que aplique este método, pues la sustitución monoalfabética preserva la distribución de frecuencias del texto original.

3.2 Cifrado César — Funcionamiento y base ASCII

El Cifrado César, atribuido a Julio César quien lo utilizó para comunicaciones militares, desplaza cada letra del alfabeto un número fijo de posiciones. Con un desplazamiento de 3 posiciones: A → D, B → E, y así sucesivamente, volviendo al inicio del alfabeto de forma circular.

En la implementación moderna con base ASCII, la operación se expresa matemáticamente como:

Cifrado: ASCII(char_cifrado) = (ASCII(char_original) - base + desplazamiento) mod N + base

Descifrado: ASCII(char_original) = (ASCII(char_cifrado) - base + N - desplazamiento) mod N + base

Donde:

base = código ASCII del primer carácter del conjunto definido por el usuario

N = tamaño del conjunto de caracteres (ej. 26 para a-z, 95 para ASCII imprimible)

Al basar el cifrado en los valores ASCII en lugar de un alfabeto fijo, la aplicación puede operar sobre cualquier conjunto de caracteres que el usuario defina: solo letras minúsculas, letras y números, o incluso todos los caracteres ASCII imprimibles (del 32 al 126).

Vulnerabilidad: con solo 25 desplazamientos posibles para el alfabeto latino estándar (o N-1 para un conjunto personalizado), la fuerza bruta es trivial. El análisis de frecuencia de Al-Kindi reduce aún más el esfuerzo necesario.

3.3 Cifrado Atbash — Funcionamiento y base ASCII

El Cifrado Atbash es de origen hebreo y su nombre proviene de las cuatro letras que ilustran su funcionamiento: Aleph (primera) ↔ Tav (última), Beth (segunda) ↔ Shin (penúltima). Es un caso particular de sustitución donde el desplazamiento equivale exactamente a invertir el orden del alfabeto.

La fórmula matemática basada en ASCII es:

ASCII(char_cifrado) = (ASCII_max + ASCII_min) - ASCII(char_original)

Donde:

ASCII_min = código ASCII del primer carácter del conjunto

ASCII_max = código ASCII del último carácter del conjunto

Ejemplo con conjunto a-z (97 a 122):

'a' (97) → 'z' (122) porque: (122+97) - 97 = 122

'b' (98) → 'y' (121) porque: (122+97) - 98 = 121

Una propiedad notable de Atbash es que es completamente simétrico: aplicar la misma operación dos veces devuelve el texto original. Esto significa que la función de cifrado y la de descifrado son idénticas. Aunque matemáticamente elegante, esta simetría no añade seguridad adicional y el análisis de frecuencia lo rompe igualmente.

3.4 ¿Por qué ya no son viables?

Combinando las vulnerabilidades descritas en la introducción con los fundamentos matemáticos anteriores, se puede resumir la inviabilidad de estos cifrados en la siguiente comparativa:

Criterio	César / Atbash	AES-256 (estándar actual)
Espacio de claves	25 opciones	2^256 opciones
Resistencia fuerza brut.	Milisegundos	Miles de millones de años
Análisis de frecuencia	Totalmente vulnerable	No aplicable
Difusión	Ninguna	Alta (avalanche effect)
Uso recomendado	Solo didáctico	Producción / datos reales

3.5 Lógica del Programa

El programa cuenta con cuatro secciones principales que el usuario debe configurar antes de ejecutar el cifrado.

1. Módulo — Se selecciona si se desea **Cifrar** o **Descifrar** el mensaje. En el caso de Atbash no importa cuál se elija ya que al ser simétrico produce el mismo resultado en ambas direcciones.

2. Tipo de Cifrado — Se elige entre **César** o **Atbash**. Al seleccionar César aparece un control deslizante para definir el número de posiciones de desplazamiento (del 1 al 50). Al seleccionar Atbash este control se oculta automáticamente ya que Atbash no requiere clave.

3. Conjunto de Caracteres — Este campo define exactamente qué caracteres participarán en el cifrado. Los caracteres que no estén en este conjunto se dejarán sin modificar. Por ejemplo, si el conjunto es solo letras minúsculas, los espacios y números pasarán al resultado tal como están.

4. Entrada — Se escribe el mensaje a procesar y se presiona el botón **Ejecutar Cifrado**.

Casos de Prueba Realizados

Caso 1 — Atbash, solo minúsculas

Charset	abcdefghijklmnopqrstuvwxyz
Entrada	base de datos distribuidas
Resultado	yzhv wv wzglh wrhgiryfrwzh

Los espacios se conservan porque no están en el charset. Cada letra se reemplaza por su espejo en el alfabeto: a↔z, b↔y, d↔w, etc.

Caso 2 — Atbash, verificación de simetría

Charset	abcdefghijklmnopqrstuvwxyz
Entrada	hvtfirwzw wv hrhgvnzh
Resultado	seguridad de sistemas

Esto demuestra que Atbash es simétrico: cifrar el resultado regresa al texto original, sin necesidad de cambiar el modo.

Caso 3 — César, desplazamiento 7

Charset	abcdefghijklmnopqrstuvwxyz
Entrada	sistemas de seguridad
Desplaz	7
Resultado	zpzalthz kl zlnbypkhk

Nótese cómo la letra t (posición 19) al sumarle 7 da 26, pero como el alfabeto termina en la posición 25, el programa regresa al inicio con la operación mod 26 y produce la letra a. Esto es la circularidad del cifrado César.

Caso 4 — César, verificación de descifrado

Charset	abcdefghijklmnopqrstuvwxyz
Entrada	zpzalthz kl zlnbypkhk
Desplaz	7
Resultado	sistemas de seguridad

Al descifrar con el mismo desplazamiento se recupera el texto original, confirmando que el programa funciona correctamente en ambas direcciones.

4. Conclusión

Con este trabajo pude entender de dónde viene la criptografía y por qué evolucionó hasta donde está hoy. Me pareció interesante darme cuenta de que desde el siglo IX Al-Kindi ya había encontrado la manera de romper estos cifrados, mucho antes de que existieran las computadoras.

Trabajar con César y Atbash me ayudó a entender conceptos que de otra forma serían muy abstractos, como el desplazamiento, el espejo de un alfabeto, o lo que significa que un cifrado sea simétrico. Son métodos simples, pero precisamente esa simplicidad los hace vulnerables: cualquier mensaje cifrado con ellos puede romperse en segundos, ya sea probando todas las combinaciones posibles o analizando la frecuencia con que aparece cada letra.

El programa que desarrollé cumple con lo que se pedía: funciona en la web, permite definir el conjunto de caracteres, distingue entre César y Atbash, y puede cifrar y descifrar correctamente. Separar el código en HTML, CSS y JavaScript también me ayudó a entender mejor cómo se organiza un proyecto web real.

Lo que me llevo de todo esto es que en seguridad informática no basta con que algo parezca complicado, tiene que demostrarse que resiste ataques reales. César y Atbash fallan esa prueba, y por eso hoy se usan algoritmos como AES-256 que han sido analizados durante décadas por expertos de todo el mundo.

5. Bibliografía

Al-Kindi, A. Y. (c. 850). Risalah fi Istikhraj al-Mu'amma [Manuscrito sobre el Descifrado de Mensajes Criptográficos]. Casa de la Sabiduría, Bagdad. Recuperado de la traducción de Salam, M. A., & Al-Kadi, I. A. (1992). Al-Kindi's Treatise on Cryptanalysis. Cryptologia, 16(2), 97–126.

Mozilla Developer Network. (2025). String.prototype.charCodeAt() / String.fromCharCode(). Recuperado de: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/charCodeAt

National Institute of Standards and Technology — NIST. (2001). Advanced Encryption Standard (AES). FIPS Publication 197. Recuperado de: https://csrc.nist.gov/publications/detail/fips/197/final