

# Cifrado afín

by Canek García ([kaan.ek@ciencias.unam.mx](mailto:kaan.ek@ciencias.unam.mx))

# Agenda

- ¿Qué es Cifrado afín?
- Algoritmo y ejemplo
- Implementación



¿Qué es el Cifrado  
afín?

# Cifrado afín

Es un tipo de cifrado por sustitución en el que cada símbolo del alfabeto en claro es sustituido por un símbolo del alfabeto cifrado siendo el número de símbolos del alfabeto en claro igual que el número de símbolos del alfabeto cifrado. Para hallar el símbolo del alfabeto cifrado que sustituye a un determinado símbolo del alfabeto en claro, se usa una función matemática afín en aritmética modular.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

- Wikipedia



# Algoritmo

# Fórmula matemática

$$c_i = (a * m_i + b) \mod n$$

donde:

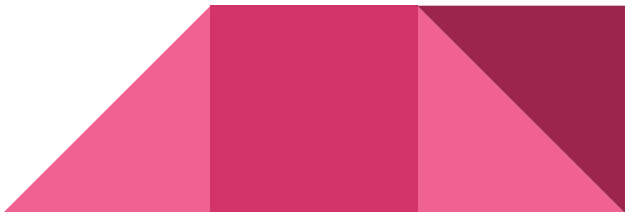
$c_i$ : Identifica el símbolo  $i$  del texto cifrado

$a$ : Se la llama **constante de decimación**

$m_i$ : Identifica el símbolo  $i$  del texto en claro

$b$ : Se la llama **constante de desplazamiento**

$n$ : Es el número de símbolos del alfabeto de cifrado (el orden)



# Clasificación

Dependiendo de los valores de  $a$  y  $b$ , existe esta clasificación:

- Si  $a = 1$ , se dice que el cifrado es **por desplazamiento puro**.

$$c_i = (m_i + b) \mod n$$

- Si  $b = 0$  se dice que el cifrado es **por decimación pura**.

$$c_i = (a * m_i) \mod n$$

- Si  $a \neq 1$  y  $b \neq 0$  se dice que el cifrado es **por sustitución afín**.

$$c_i = (a * m_i + b) \mod n$$


# Ejemplo para el símbolo 'a'


a=5

b=15

n=27 (total de caracteres en el alfabeto)

$$(5 * 1 + 15) \mod 27 = 20$$

por tanto, el carácter asociado será el que ocupa la posición 20 empezando desde 0, la 's'.





# Implementación

```

/**
 * c = (am + b) mod n
 * @param cadena [description]
 * @param a      [description]
 * @param b      [description]
 * @return       [description]
 */
public static String cifrar(String cadena, int a, int b) {
    String str = "";

    for (int i=0; i<cadena.length(); i++) {
        char singleCharacter = cadena.charAt(i);

        if (Character.isLetter(singleCharacter)) { //filtro, solo letras
            // (ax + b) % 27
            singleCharacter = (char) ( (a * (int)(singleCharacter + 'a') + b) % 27 + 'a');
        }

        str +=singleCharacter;
    }

    return str;
}

```

# Código fuente

[https://github.com/kanekko/cryptography\\_2019-1/tree/master/Lab01](https://github.com/kanekko/cryptography_2019-1/tree/master/Lab01)