**Datos necesarios**

Vector de char para las claves encriptadas : char encriptadas[]

Vector de char para las claves no encriptadas : char originales[]

Entero o entero largo para el rango inferior long cota\_inferior

Entero o entero largo para el rango superior: long cota\_superior

Vector de enteros para almacenar los id de los procesos calculadores

Entero para almacenar cuántos procesos calculadores hay

Entero para almacenar el nº de claves a desencriptar

typedef struct clave\_desencriptada{

Char clave\_encontrada[];

int id\_proceso\_calculador;

long intentos\_realizados ; //suma de los intentos que ha realizado cada proceso hasta hallarla

Double tiempo\_empleado ; //solo por el proceso que la averigua

}clave\_desencriptada;

clave\_desencriptada vector[total\_claves]; //vector de claves desencriptadas con los datos estadísticos

PODEMOS NUMERAR LAS CLAVES PARA HACER UN VECTOR DE ENTEROS DONDE VAYAMOS MARCANDO CUÁLES HAN SIDO YA ENCONTRADAS PARA NO VOLVERLAS A REPARTIR.

Double para el nº de claves comprobadas por segundo

**Mensajes necesarios**

Mensaje tipo 1: el que use el proceso de E/S para indicar que lo que manda es una clave a desencriptar exclusiva para un proceso.

Mensaje tipo 2: el que use el proceso de E/S para indicar que lo que manda es una clave a desencriptar sobre la que ya está trabajando otro proceso

Mensaje tipo 3: el que manda el proceso de E/S para mandar parar al proceso calculador para indicarle que alguien le va a ayudar

Mensaje tipo 4: el que manda cada proceso calculador al proceso de E/S para indicarle que han averiguado la clave

Mensaje tipo 5: el que indique al proceso o procesos calculadores que la clave que intentaban averiguar ya ha sido averiguada por otro proceso, que se detengan. (No sé si la reciben por parte del proceso de E/S (creo que sí) o por otro calculador, el caso es que para el tipo es indiferente)

Mensaje tipo 6: el mandado por el proceso de E/S a los calculadores para indicarles que ya se han averiguado todas las claves y mueran.