

Crypto IPC

Autor: Héctor Josué Ponsoy Ayala

Carné: 201807220

Clases:

- **Principal:** Clase donde se encuentra la clase Main y los métodos con los cuales se hizo el procedimiento de cifrar.
- **Descifrar:** Clase utilizada para hacer todo el procedimiento requerido para descifrar las matrices ingresadas.

Métodos:

- **Clase Principal**
 - **public static void matrizAscii:** No recibe ningún parámetro, dicho método sirve para validar que la longitud del texto se aloje en su respectivo múltiplo y luego procederá a convertir el texto en código ASCII.
 - **public static void leerArchivo:** Recibe como parámetro un entero el cual servirá para darle tamaño a la matriz en la cual se alojara la matriz impresa en el .txt.
 - **public static void multiplicarMatrices:** No recibe ningún parámetro, la función de este método es multiplicar la matriz en código ASCII con la matriz leída por el método leerArchivo().

- **public static void reporteCifrado:** Este método crea un reporte en HTML para el procedimiento de cifrar, dicho método recibe como parámetro un String, un char, un Integer y dos int, esto con el objetivo de poder concatenar dichas variables en el cuerpo del código HTML.

- **Clase Descifrar**

- **public void generarCantColumns:** No recibe ningún parámetro, dicho método sirve para calcular la cantidad de columnas que posee el .txt de la primera ruta ingresada.
- **public void generarCantFilas:** Recibe un String y un entero como parámetros, dicho método sirve para calcularla cantidad de filas que posee el .txt de la primera ruta ingresada.
- **public void leerArchivo1:** Este método sirve para leer el archivo que contiene la matriz de NxM.
- **public void generarCantColumns2:** Este método calcula la cantidad de columnas que tiene la matriz del segundo archivo txt.
- **public void generarCantFilas2:** Este método calcula la cantidad de filas que posee la matriz alojada en el segundo archivo txt.

- **public void leerArchivo2:** Método el cual su función principal es leer la matriz del segundo archivo y alojarla en una matriz de sus mismas dimensiones.
- **public static void casteoPrimerArchivo:** Su función es hacer el casteo de int a double de la matriz de NxM.
- **public static void casteoSegundoArchivo:** Su función es hacer el casteo de Integer a double de la matriz de NxN.
- **public static double[][] matrizInversa:** Método el cual calcula la matriz inversa de la matriz de NxN.
- **public static double[][] matrizAdjunta:** Este método calcula la matriz adjunta con la cual se procede a sacar la inversa.
- **public static double[][] matrizCofactores:** Método el cual calcula la matriz de cofactores.
- **public static double[][] matrizTranspuesta:** Método que sirve para calcular la matriz transpuesta.
- **public static double determinante:** La función de este método es calcular el determinante de la matriz.

- **public static void multi:** La función de este método es multiplicar la matriz inversa de $N \times N$ por la de $N \times M$, con lo cual se genera una matriz resultante.
- **public static void casteoMatrizResultante:** Este método genera el casteo de double a int para la matriz resultante.
- **public static void casteoAChart:** Este método genera el casteo de int a char para la matriz casteada anteriormente.
- **public static void reporteDescifrar:** método el cual genera un reporte en HTML detallado de los procedimientos efectuados.