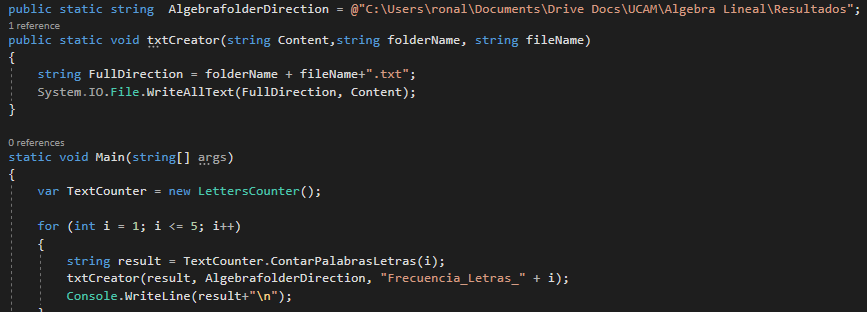
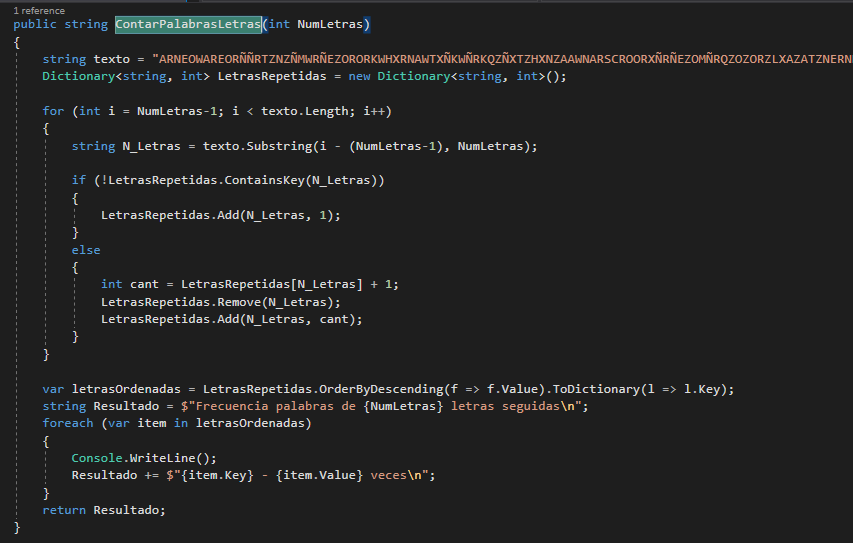
# Texto Nomenclator

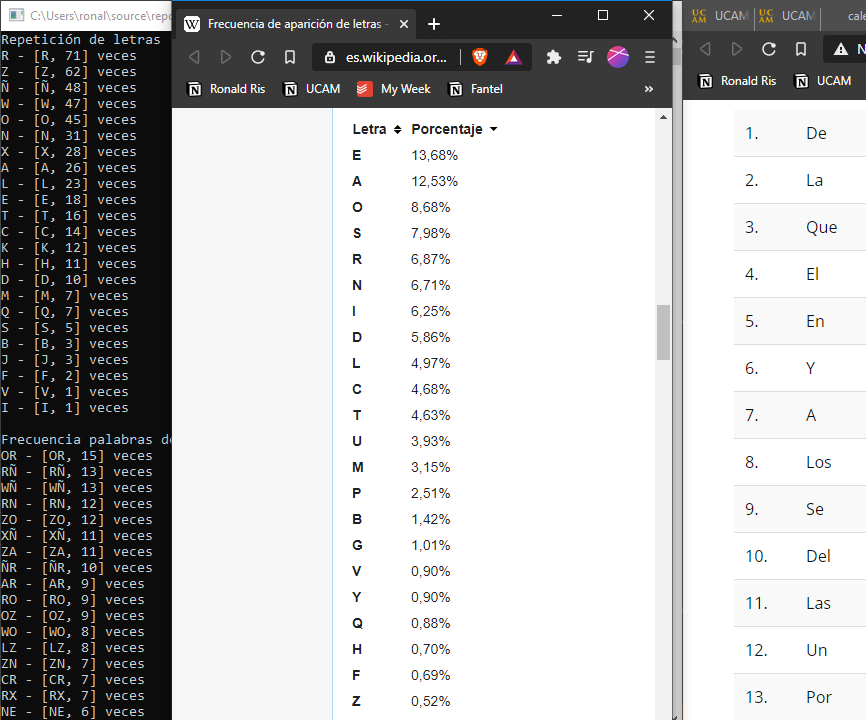
* Primero tras un poco de investigación sobre las letras más frecuentes y las palabras más frecuentes.
* Decidí hacer un programa que cuente por mí la frecuencia de aparición de las letras y palabras.

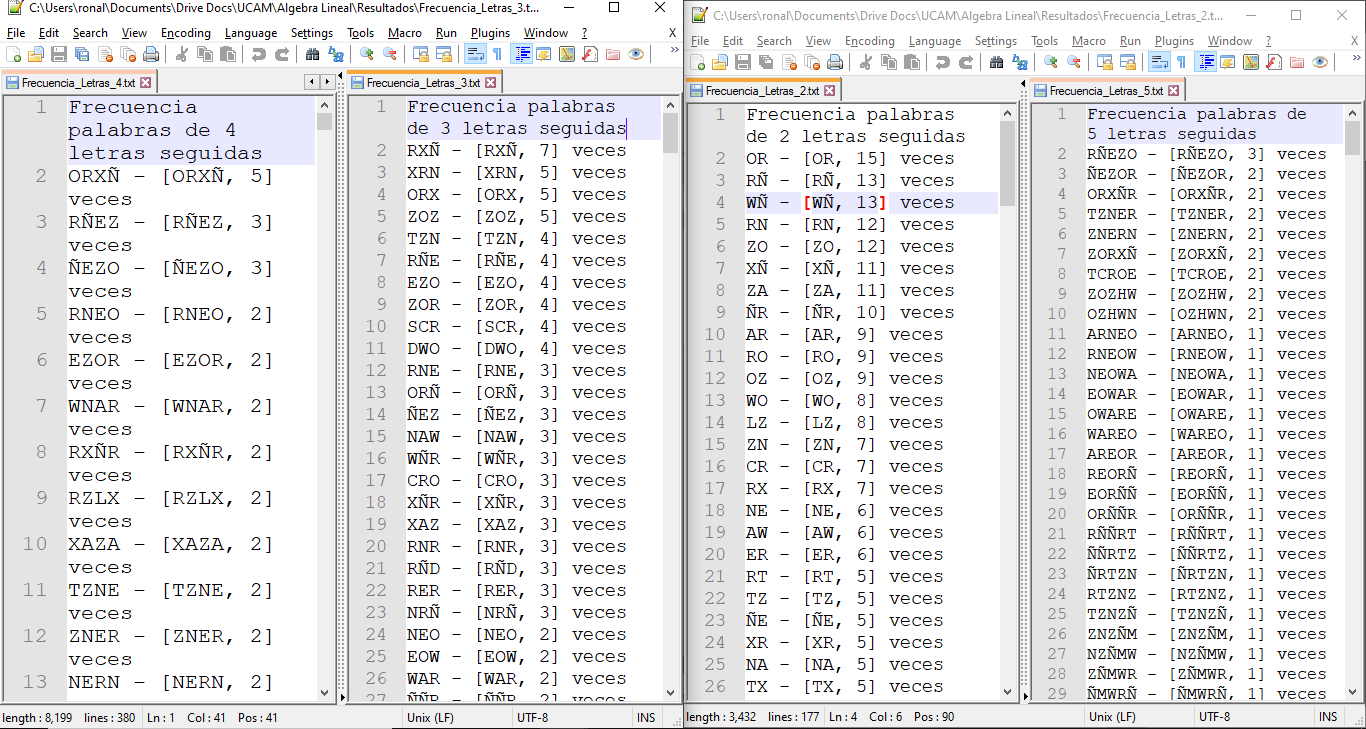
**Código:**

****

****

**Resultados:**

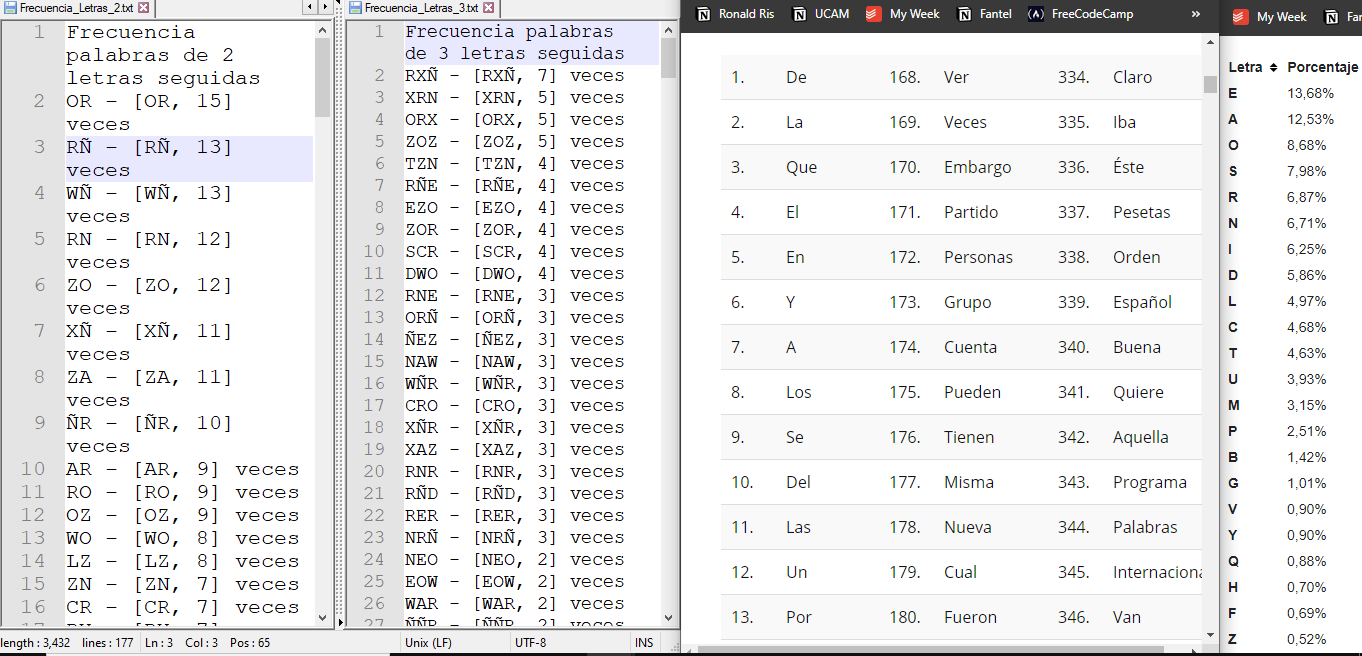




Esos fueron los resultados que obtuvé. Los demás los guarde en la capeta que anexo.

**Así descubrí las primeras letras:**

* R es E
* Z es A

Y así continúe con el análisis en base a las palabras con más frecuencia del español

Luego intenté:

Encontrar palabras de 2 letras basándome en la A y la E que son las que ya he descubierto. Así empecé a hacer varias combinaciones con algunas “palabras” de 2, 3, 4 y 5 letras más repetidas que encontré con los algoritmos que hice previamente.

Primero busqué las **letras más repetidas del español** (A, E, O, S, R, N) y tras descartar combinaciones que no tienen sentido y dejando abiertas varias posibilidades.

Luego recordé **Verbos en Infinitivo.** Gracias a sus terminaciones descubrí quién era la R y apoyándome en varios descartes de combinaciones de palabras sin sentido he avanzado a esto:

* R -> E
* Z -> A
* Ñ -> S
* W -> O
* O -> R
* N -> N
* X -> I
* A -> D

Un avance considerable. Seguiré aplicando lo mismo y veré si encuentro alguna palabra con sentido

Tengo el texto impreso, empecé a escribir las letras que ya he encontrado y hallé algo curioso, **mi nombre encaja al final.**

Ya tengo las letras

**O R**

**W O**

**N N**

**Z A**

L

**A D**

**R E**

**O R**

**N N**

**R E**

**Ñ S**

E

**W O**

Las letras en negrita ya las había averiguado antes, al sustituir tengo eso.

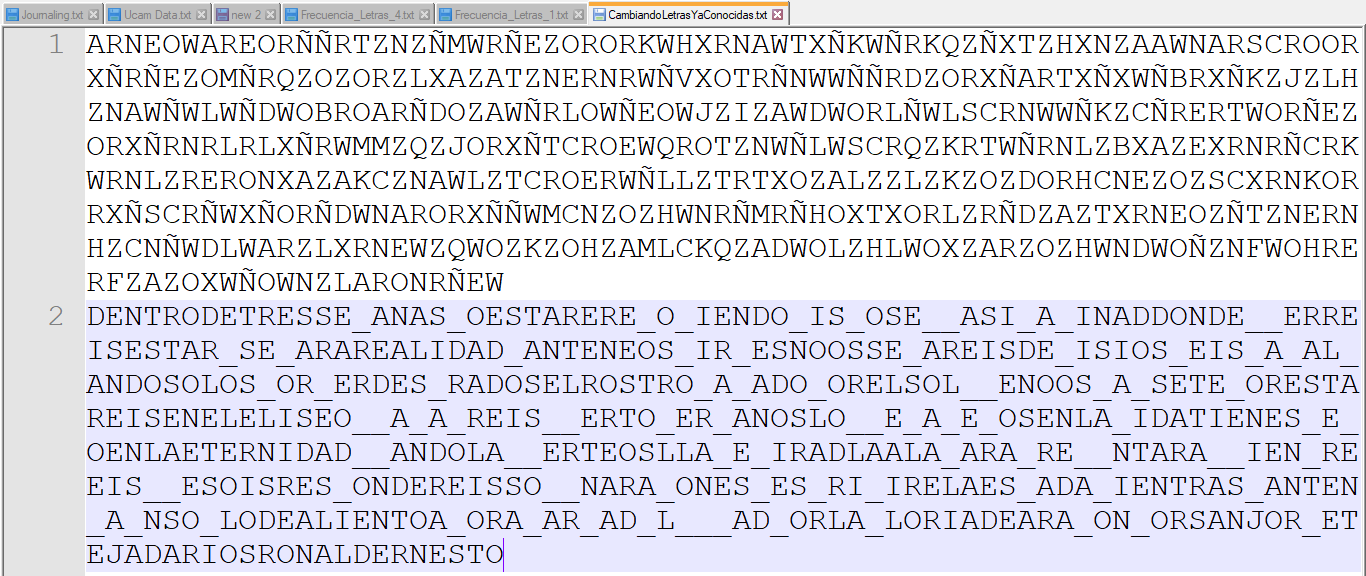
Extrapolando y analizándolo completo, está todo mi nombre.

Ahora conozco el siguiente repertorio de letras:

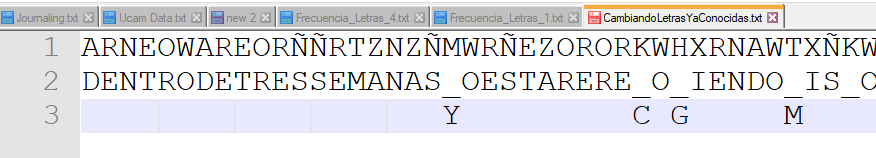
1. R -> E
2. Z -> A
3. Ñ -> S
4. W -> O
5. O -> R
6. N -> N
7. X -> I
8. A -> D
9. L ->L
10. E ->T
11. F ->J

He descubierto 3 letras más y he confirmado las letras que previamente había descubierto

**Seguiré sustituyendo letras ya conocidas.** Está vez me apoyaré de un programa para que me vaya corrigiendo a medida que hago avances.

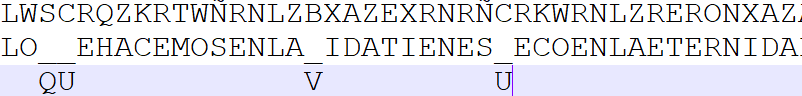


Intentaré deducir letras de la siguiente manera:

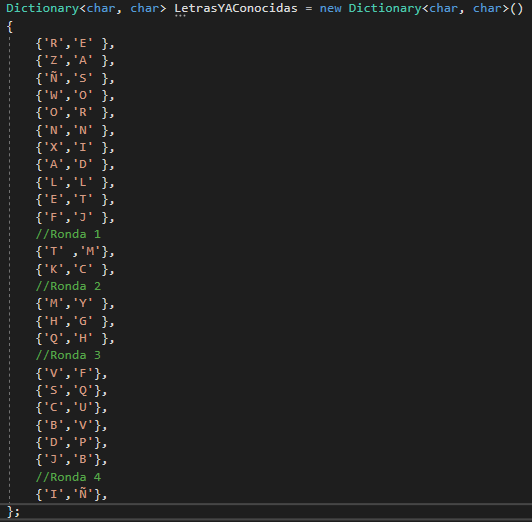


Y los resultados ir agregándolos al diccionario de letras ya conocidas, así cada vez que ejecute el programa esas sean cambiadas y sea más fácil descubrir las letras restantes.

Repetiré el proceso varias veces



Resultados:



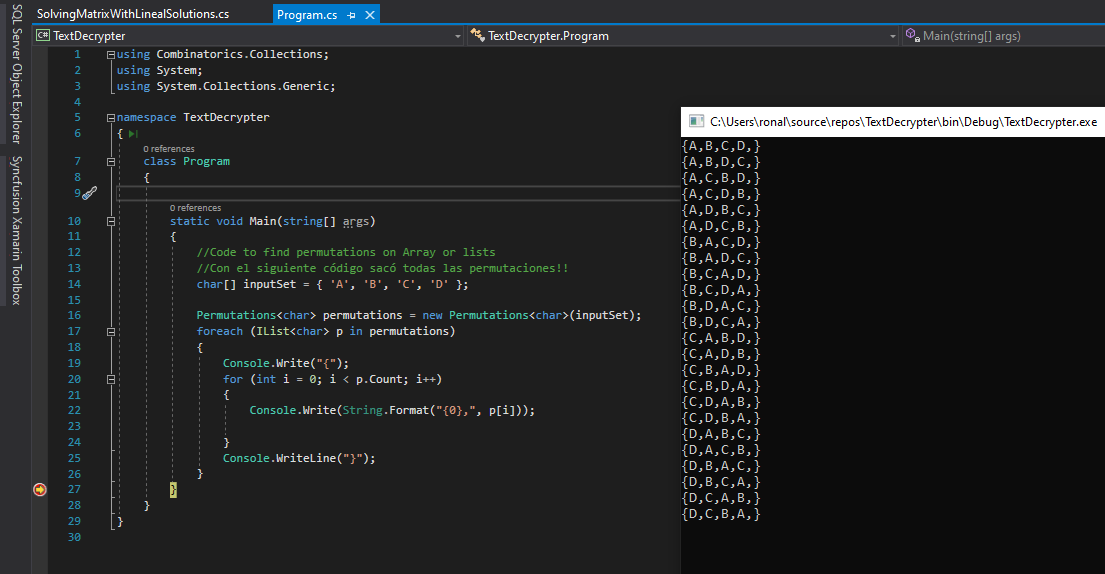
**Ya he encontrado todas las letras del texto.**

DENTRO DE TRES SEMANAS YO ESTARE RECOGIENDO MIS COSECHAS IMAGINAD DONDE QUERREIS ESTAR Y SE HARA REALIDAD MANTENEOS FIRMES NO OS SEPAREIS DE MI SI OS VEIS CABALGANDO SOLOS POR VERDES PRADOS EL ROSTRO BAÑADO POR EL SOL QUE NO OS CAUSE TEMOR ESTAREIS EN EL ELISEO Y YA HABREIS MUERTO HERMANOS LO QUE HACEMOS EN LA VIDA TIENE SU ECO EN LA ETERNIDAD CUANDO LA MUERTE OS LLAME MIRADLA A LA CARA PREGUNTARA QUIEN CREEIS QUE SOIS RESPONDEREIS SOY UN ARAGON ES Y ESGRIMIRE LA ESPADA MIENTRAS MANTENGA UN SOPLO DE ALIENTO AHORA CARGAD Y LUCHAD POR LA GLORIA DE ARAGON POR SAN JORGE TEJADA RIOS RONALD ERNESTO

**IMPORTANTE:** Solo hay 23 letras. Es decir, hay 4 letras sobras cuyos valores no sé y me generan 4! (24) posibles resultados los cuales posiblemente me afecten en el futuro.

**G, P, U, Y. No aparecen en todo el texto nomenclátor, pero si aparecen en el texto codificado con la matriz.**

Es decir, esto aparte de no saber la matriz clave, me genera 24 posibles sets de sistemas de ecuaciones



Con el código de arriba ya tengo todas las permutaciones de un arreglo. Me he apoyado de la librería (nuget en C#) llamada [Combinatorics](https://www.nuget.org/packages/Combinatorics/).

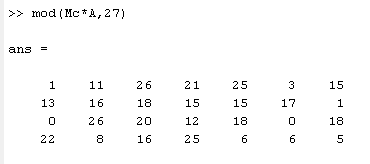
No proseguiré de está manera, no conozco la Matriz Clave ni la permutación clave, buscaré otro enfoque que omita las permutaciones estas.

# Texto codificado con la matriz

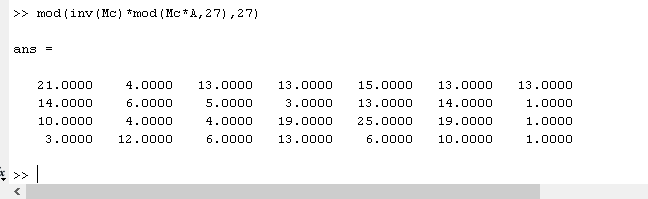
Evitaré tratar con las 24 variantes que me genera la falta de las letras G, P, U, Y. De manera que todas las columnas que tengan alguna de esas letras, las omitiré. La cantidad de veces que esas 4 letras salen en el texto codificado es: 131 veces. El textoMuret tiene 210 columnas. Así que en teoría puedo conseguir suficientes columnas para sacar ecuaciones (asumiendo también que en el texto original no lleva K, X, W, Z que son las letras que se transforman en G, P, U, Y).

Debo conseguir:

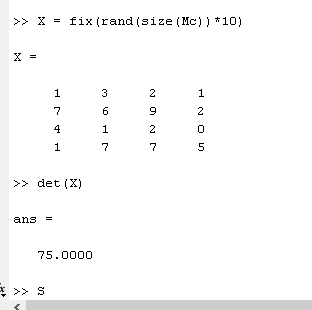
* Manera eficiente de multiplicar para conseguir ecuaciones:
  + Obviar las columnas que tienes G,P,U,Y a modo que todos los resultados de Mc\*textoMuret tengan solución (a modo de encontrar Mc inversa primero )
  + Resolver el sistema de 16 ecuaciones a mano o bien escribir un algoritmo que lo resuelva por mi (esta última me tienta)
* Probando Matlab – Resolveré el ejemplo usando Matlab para comprobar que entiendo el proceso completo de como codificar y decodificar la matriz.



Haciendo la siguiente operación es como regreso a la matriz original:

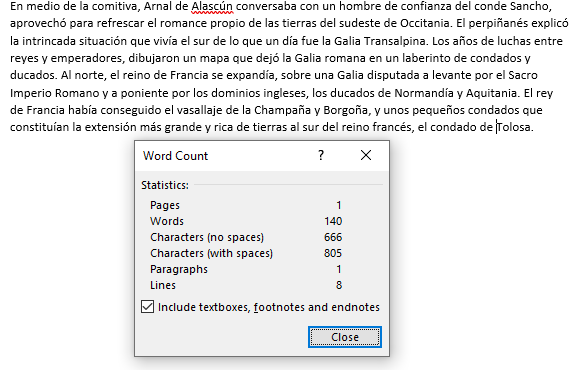


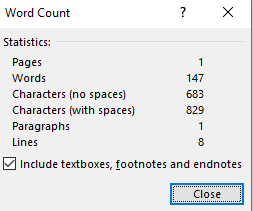
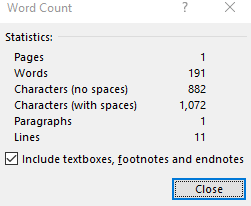
Intenté obtener matrices aleatorias, pero desistí porque su determinante variaba mucho y no me parecía que llegaría lejos así.



**Ideas:**

1. **Hacer una especie de matriz LU** (para ello necesito conocer el texto de entrada, pero lo desconozco).
   1. La matriz codificada tiene 840 caracteres
      1. Necesito en Muret un parrafo que tenga 837, 838, 839 o 840 caracteres para poder usarlos como posibles textos de entrada.
      2. Tengo el libro descargado en Kindle, intenté ponerlo en txt, pero cuando lo copio no guarda formato de parrafo. Es decir tendría que copiar CADA PARRAFO uno por uno o al menos los parrafos grandes que podrían tener más de 837 caracteres.
      3. El libro está en formato EPUD o AZW, es complicado sacar el texto, tengo la opción de hacer un web scrapper ya sea en web al AZW, pero ambos están encriptados, **así que optaré por ir eligiendo manualmente los parrafos que sean posibles soluciones.**
         1. Me apoyaré de word es iré página por página sacando los textos que puedan cumplir estás condiciones





* + - Caracteres como comas y puntos cuentan. Así que buscaré textos con más de 840 caracteres sin espacios. Guardaré todo en un txt.

1. **Inventar una matriz random que sea mi Mc inversa para llegar al texto original, descartar textos hasta encontrar uno que tenga sentido en español.** 
   1. Esta idea es muy buena hasta cierto punto, intenté hacerlo en código apuyandome de librerias de código abierto y sistemas de ecuaciones con factorización de matrices LU (SolvingMatrixWithLinealSolutions.cs)
   2. Por el momento lo descarto ya que he encontrado librerías que solo solucionan Vectores (Matrices de una columna).
2. **Buscar un software que haga la desencriptación por mi.**
   1. **Criptoclásicos v2.1 -** <http://www.criptored.upm.es/software/sw_m001c.htm>
      1. Software de dominio publico Universidad Politécnica de Madrid, ETS de Ingeniería de Sistemas Informáticos
      2. Descartado tras varios testeos. La parte de cifrado de Hill solo acepta matrices claves que solo tengan números positivos

## 

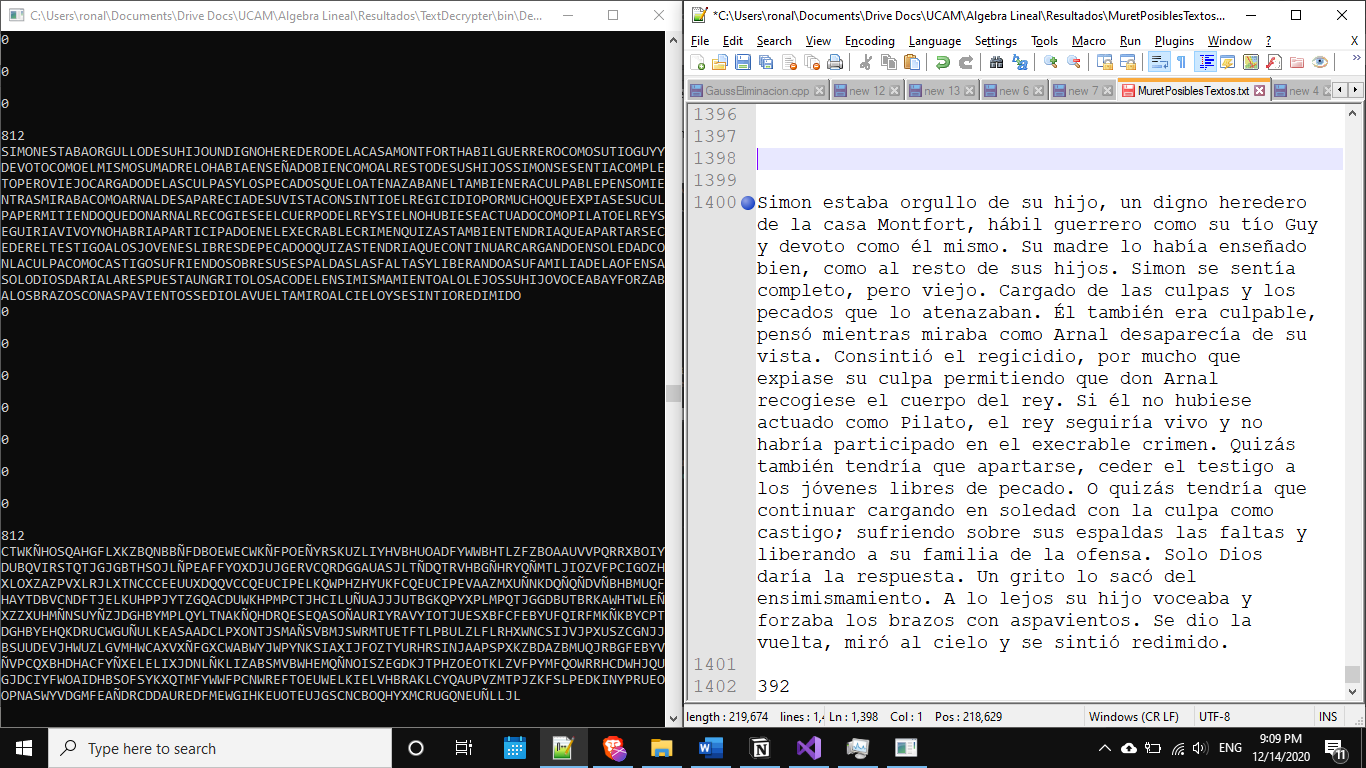
Software bastante útil, pero NO FUNCIONAL para este caso debido a las siguientes razones:

* Software asume que A=0,B=1… Y así, a pesar de que se puede manejar en Español (Z-A con 27 caracteres)
* Las matrices clave únicamente acepta valores positivos.

Finalmente decidí descartarlo tras varias pruebas inclusive convirtiendo el texto de ejemplo “UN TRABAJO LABORIOSO Y BONITO” a la numeración que ellos usan, calcular el texto codificado manualmente y luego probar el software a ver si obtendría dicha matriz clave que use para codificar.

## Cifrado de Hill 227

Ese es el nombre del tipo de cifrado que se ha usado para hacer el texto codificado con las matriz.



El texto mostrado en la pantalla tiene exactamente la misma cantidad de caracteres que el texto que estoy de la Matriz Codificada. Tiene pinta de ser ese.