



# Instituto Politecnico Nacional Escuela Superior de Computo

## Teoria de la computacion

Practica 1 Sigma Asterisco(Universo)

Profesor: Juarez Martinez Genaro

4CM6

Torres Abonce Luis Miguel  
Febrero 2022

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Marco Teorico</b>	<b>4</b>
<b>3</b>	<b>Desarrollo</b>	<b>5</b>
3.1	Codigo . . . . .	5
3.2	Explicacion . . . . .	7
3.3	Capturas de los resultados n=27 . . . . .	8
3.4	Graficas Generadas con los archivos de texto . . . . .	11
<b>4</b>	<b>Conclusiones</b>	<b>14</b>
<b>5</b>	<b>Bibliografia</b>	<b>15</b>

# 1 Introducción

Descripción del problema:

El problema del sigma asterisco o universo consiste en concatenar todas las combinaciones posibles de un alfabeto en este caso es el binario (1,0) usando la formula ( $2^{\text{pot } n}$ ) para calcular las combinaciones posibles, esto es equivalente hacer el producto cartesiano de 2 numeros, la manera que utilice para resolver el problema es sencilla pero me costo un poco hallarla

Basicamente tenemos que hacer la union de todos las combinaciones desde 0 hasta n ej.

alfabeto =0,1

n=2 esta es la longitud de la cadena

E2=(e,0,1,00,01,10,11)

e= espacio vacio

## 2 Marco Teorico

String es una concatenacion de caracteres

La longitud de la cadena es el numero de caracteres que tiene la cadena

El sigma asterisco esta conformado por el sigma + este ultimo es

- $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$ .

Y el sigma asterisco es igual a sigma + incluyendo el espacio vacio.

- $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$ .

## 3 Desarrollo

### 3.1 Código

```
import random
from math import log10

def combinacion(n,x):
    for i in range(2**n):
        a=format(i,"b").zfill(n)
        f.write(a+",")
        simbolos.write(str(x)+","+str(n)+"\n")
        simbolos10.write(str(x)+","+str(round(log10(n),2))+"\n")
        unos.write(str(x)+","+str(a.count("1"))+"\n")
        try:
            unos10.write(str(x)+","+str(round(log10(a.count("1")),2))+"\n")
        except:
            unos10.write(str(x)+","+str("0")+"\n")
        x+=1
    return x

opcion=5
while opcion!=1:
    print ("\tMenu del programa")
    print("1. Dar valor a n")
    print("2. Dar valor a n aleatorio")
    opcion=int(input("Seleccione una opcion: "))
    if opcion==1:
        n=int(input("Valor de n: "))
    elif opcion==2:
        n=random.randint(0,1000)
        print("El valor de n es: "+str(n))
    n+=1
    f=open('universo.txt','w')
    simbolos=open('simbolos.csv','w')
    simbolos10=open('simbolos10.csv','w')
    f.write('E'+str(n-1)+'= ')
    f.write('{')
    unos=open('unos.csv','w')
```

```

unos10=open('unos10.csv','w')
x=1
l=0
while x!=n:
    l=combinacion(x,l)
    x+=1
f.write('}')
f.close()
simbolos.close()
simbolos10.close()
unos.close()
unos10.close()
opcion=int(input("Quieres Calcular otra n (1=no, 0=si): "))
if(opcion==0):
    print("Hasta luego")

```

### 3.2 Explicacion

En el programa se realizo en el lenguaje de python, primero importe las librerías que ocuparía en este caso random y itertools, después realice un menú para dar la opción de dar valor a n o aleatoriamente ademas de incluir el ciclo while para salir del programa hasta que el usuario lo requiera

Después cree la función combinación que básicamente recibe 2 enteros, después mi idea fue tener un entero e ir aumentando de 1 unidad en unidad y antes de escribir en el documento transformarlo a su forma binaria y rellenar con 0 a la derecha, de acuerdo al tamaño que le que corresponde a la cadena, esto ya que los números binarios hacen todas las combinaciones posibles de 1 y 0.

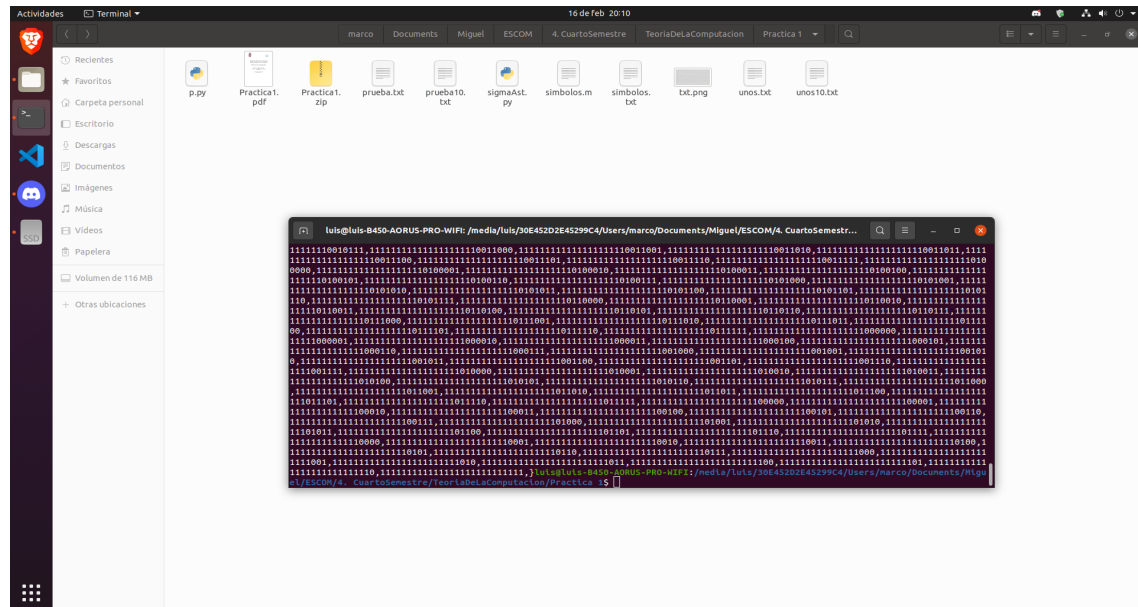
Después en la función principal después de el menú se abren se abren todos los archivos a utilizar para el archivo de texto se utilize un txt y para los datos de cantidad de simbolos, cantidad de simbolos asi como su log base 10 de cada uno se utilizaron archivos csv que separa los datos mediante una coma para que sea mas facil de graficarlo en matlab

Inicializo la variable x que utilizare dentro de un ciclo while para mandar a la función combinación y así recorra desde el 0 hasta n que se requiere y hago el universo, ademas la l solo se utiliza para los archivos que servirán para graficar y llevar una cuenta correcta

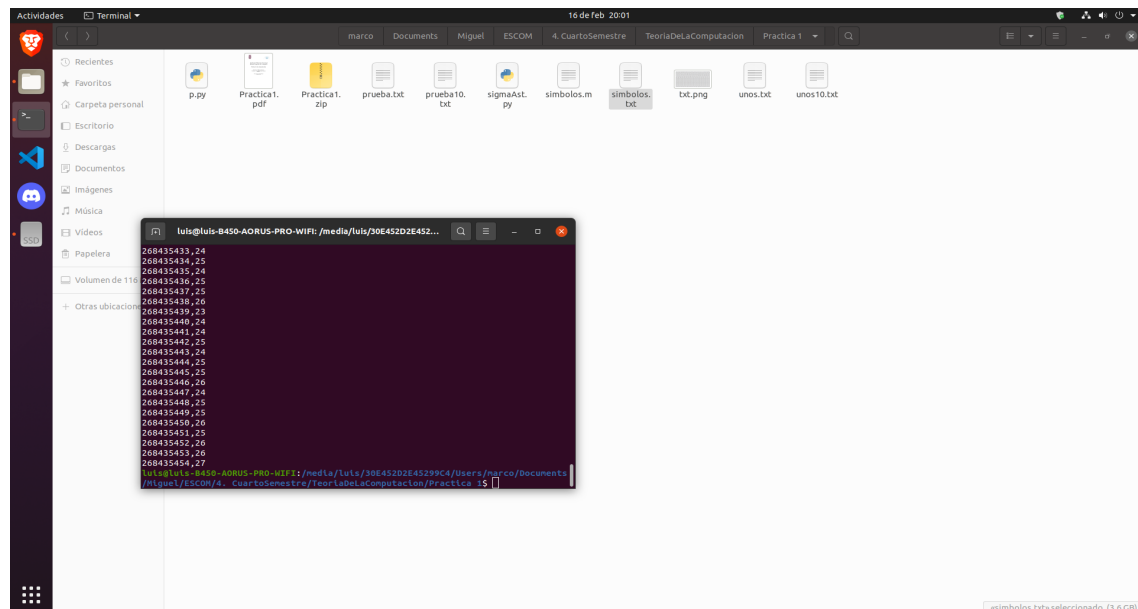
Por último se cierran todos los archivos de texto que se utilizaron.

### 3.3 Capturas de los resultados n=27

Universo:

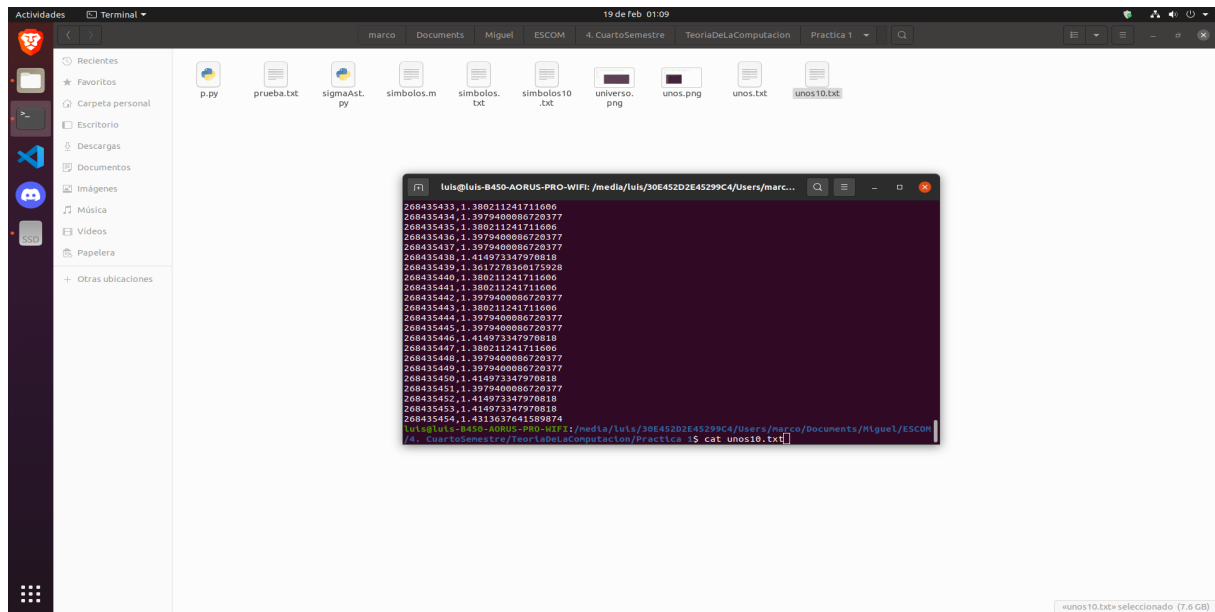


Cantidad de unos:

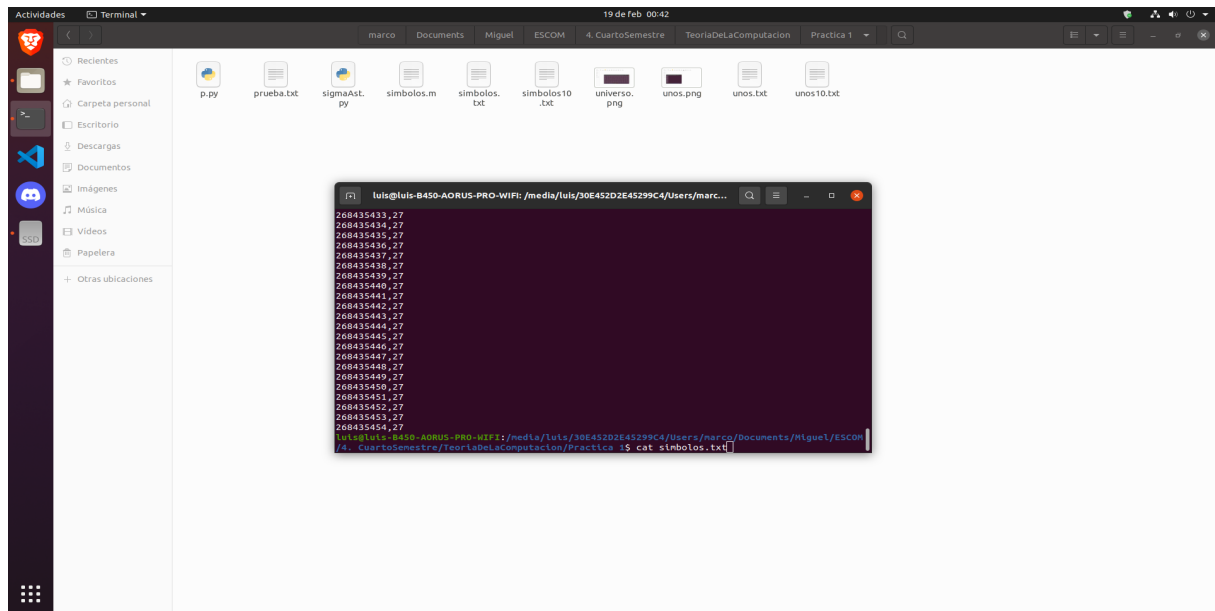




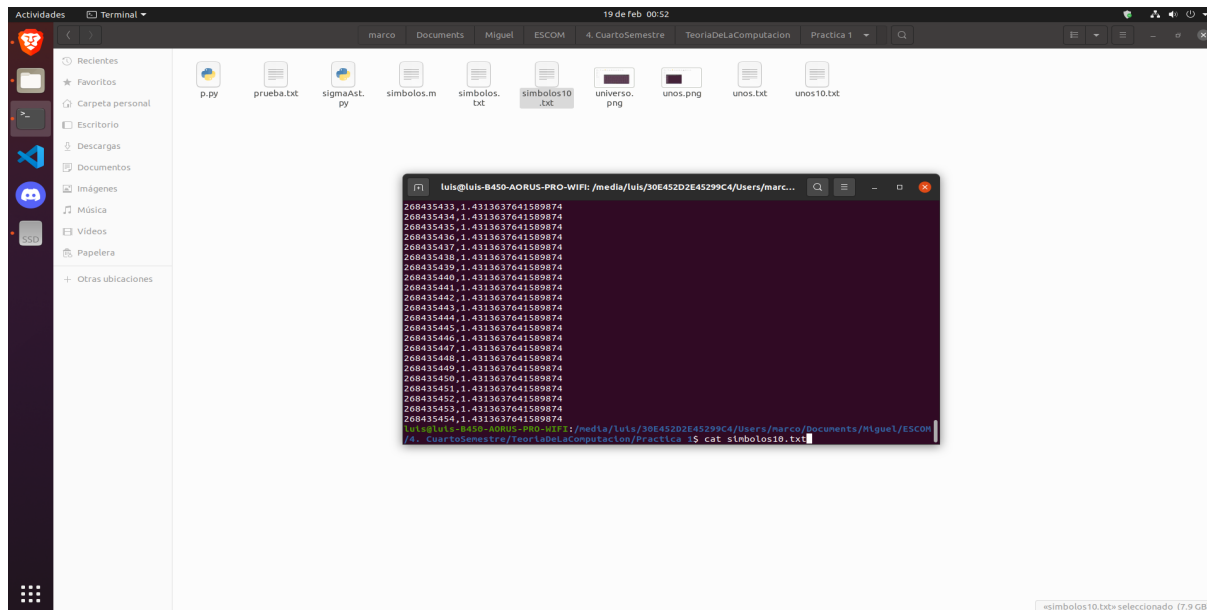
## Log base 10 de cantidad de unos en las cadenas:



## Cantidad de simbolos:



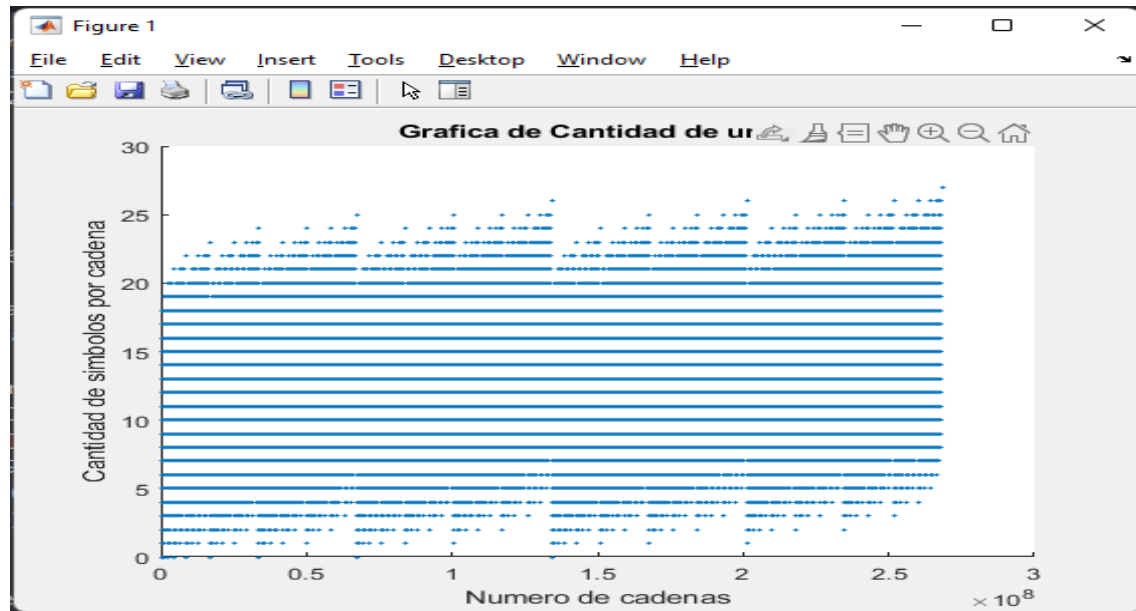
## Log base 10 de cantidad de simbolos:



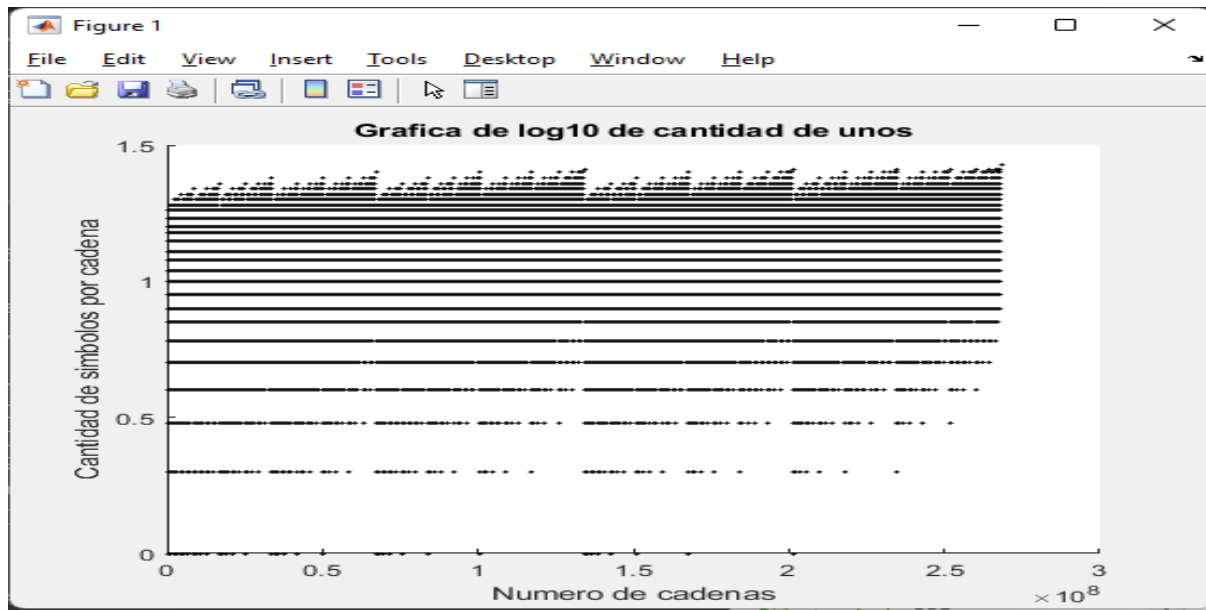
El archivo mas pesado fue de alrededor de 7Gb mientras mas aumentaba la n el tiempo se invrementaba de manera exponencial.

### 3.4 Graficas Generadas con los archivos de texto

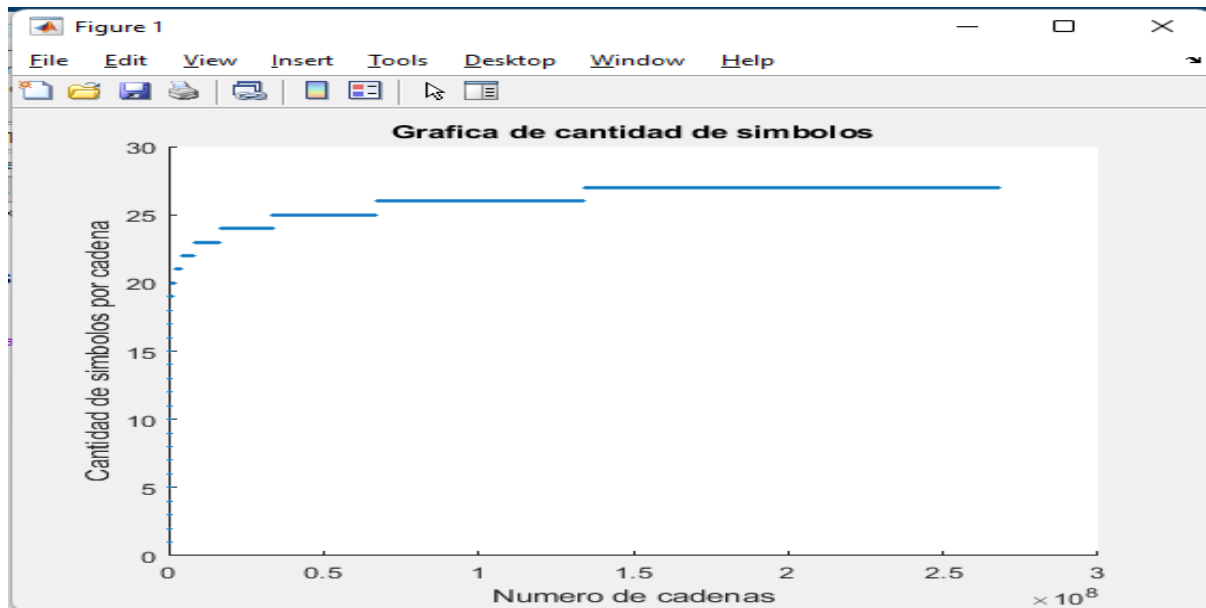
Cantidad de unos:



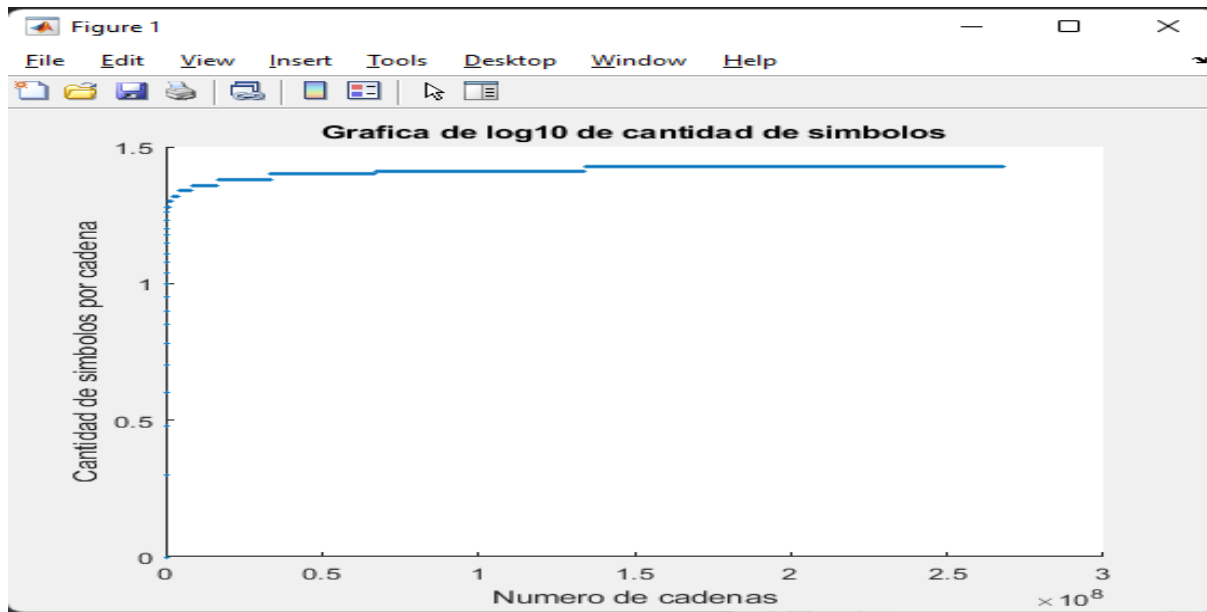
Log base 10 de cantidad de unos en las cadenas:



Cantidad de simbolos:



Log base 10 de cantidad de simbolos:



## 4 Conclusiones

El programa fue sencillo pero si me dio para pensar una solución pero en cuanto encontré por lo menos esta solución fue fácil de programar, un problema que tuvo fue con abrir los archivos de texto ya que son demasiado pesados para abrir con Windows, después de probar varios editores de texto(y preguntarle al profesor), abrí el archivo de texto desde la terminal de linux, tardo alrededor de 8 minutos, menos de la mitad de tiempo en ejecutar el programa de  $n=27$  el cual tardo alrededor de media hora. Me di cuenta que al no utilizar listas como intente resolver el problema, y utilizar solo strings y reutilizar la cadena, mi computadora por lo menos no utilizaba casi memoria ram ya que no almacenaba ninguna materia, ademas corregí el programa y quite todos los comandos lenght que usaba.

## 5 Bibliografia

1. libro "Introduction to Automata Theory, languages, and computation", John E. Hopcroft, 2nd edition
2. <https://docs.python.org/3/library/itertools.html>
3. <https://www.delftstack.com/es/howto/matplotlib/how-to-connect-scatterplot-points-with-line-in-matplotlib/>