

REPORTE

Programa 1 - universo

Materia : Teoría de la Computación

Grupo : 4CM1

Alumno : Julio Cesar Hernández Reyes

Docente : Juárez Martínez Genaro

1. Introducción

En este reporte se explicara como se realizo el programa 1 - universo, el cual es un programa que calcula el universo de las cadenas binarias (Σ^n), dada una "n" que introduzca el usuario o que el programa lo determine automáticamente.

Resumen del programa:

Entrada: Un numero (Limite del programa) entre 0 y 1000

Salida: Universo del alfabeto binario en un .txt

Después del archivo de salida, se grafico el número de unos de cada cadena. El eje de las "x" representan la cadena y el eje de las "y" el número de unos que tiene esa cadena.

Específicamente en este reporte, se calculo y grafico cuando $n=26$.

Para la realización del programa se utilizo el lenguaje C++, para implementar el programa se uso el IDE Dev-C++ para la codificación y ejecución del programa, el cual ya tiene compilador y ejecutador, en la misma interfaz.

Para la graficación se uso Python, para la implementación se hizo otro pequeño programa para la grafica en PyCharm que igual tiene ejecutador dentro de la interfaz.

Se usaron IDEs n vez de compilar y ejecutar por consola, esto para una mayor facilidad a la hora de corregir errores y algunos detalles de los programa.



Dev-C++
Aplicación



PyCharm Community Edition
2021.2.1
Aplicación

2. Conceptos importantes

2.1. Alfabetos

Un alfabeto es un conjunto de símbolos finito y no vacío. Convencionalmente, utilizamos el símbolo Σ para designar un alfabeto. Entre los alfabetos más comunes se incluyen los siguientes:

1. $\Sigma = \{0, 1\}$, el alfabeto binario.
2. $\Sigma = \{a, b, \dots, z\}$, el conjunto de todas las letras minúsculas.

2.2. Cadenas de caracteres

Una cadena de caracteres (que también se denomina en ocasiones palabra) es una secuencia finita de símbolos seleccionados de algún alfabeto. Por ejemplo, 01101 es una cadena del alfabeto binario $\Sigma = \{0, 1\}$. La cadena 111 es otra cadena de dicho alfabeto.

2.3. La cadena vacía

La cadena vacía es aquella cadena que presenta cero apariciones de símbolos. Esta cadena, designada por ϵ , es una cadena que puede construirse en cualquier alfabeto.

2.4. Longitud de una cadena

El número de posiciones ocupadas por símbolos dentro de la cadena. Por ejemplo, 01101 tiene una longitud de 5. Generalmente podremos utilizar la expresión “número de símbolos” cuando realmente a lo que se está haciendo referencia es al “número de posiciones”.

La notación estándar para indicar la longitud de una cadena w es $|w|$.

Por ejemplo, $|011| = 3$ y $|\epsilon| = 0$.

2.5. Potencias de un alfabeto

Si Σ es un alfabeto, podemos expresar el conjunto de todas las cadenas de una determinada longitud de dicho alfabeto utilizando una notación exponencial. Definimos Σ^k para que sea el conjunto de las cadenas de longitud k , tales que cada uno de los símbolos de las mismas pertenece a Σ .

3. Desarrollo

3.1. Explicación de cadenas de un alfabeto

Observe que $\Sigma^0 = \{\epsilon\}$, independientemente de cuál sea el alfabeto Σ . Es decir, ϵ es la única cadena cuya longitud es 0.

Si $\Sigma = \{0, 1\}$, entonces $\Sigma^1 = \{0, 1\}$, $\Sigma^2 = \{00, 01, 10, 11\}$, $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$, etc.

Por convenio, el conjunto de todas las cadenas de un alfabeto Σ se designa mediante Σ^* . Por ejemplo, $\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$. Expresado de otra forma,

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$$

3.2. Código del programa

Declaración de librerías: Escribe el siguiente código en un fichero llamado `hello.c`:

```
#include <iostream> // Para poder usar cin y cout
#include <string.h> // Para trabajar con cadenas
#include <fstream> // Para poder trabajar con .txt
#include <cstdlib> //Aquí estan las funciones rand y srand
#include <ctime> // De aquí se usa la función time para
//obtener un indicador del tiempo actual del sistema para
//la semilla
```

Declaración de namespace para poder usar `cout` y `cin` sin el `std::`: al principio

```
using namespace std;
```

Declaración de las funciones del programa:

```
void menu();
void manualmente();
void automaticamente();
string Decimal_A_Binario(int);
int limitedeunos(int);
string llenar0(string, int);
void sigmaasterisco(int);
void ContadorDeUnosATexto();
```

Inicio del main(), en donde solo se llama a la función menu()

```
int main () {  
    menu();  
    return 0;  
}
```

La funcion menu(), en donde se despliega el menu del programa:

```
void menu() {  
    int opcion = 0;  
    do{  
        system("cls");  
        cout << "Programa que calcula el universo del alfabeto  
            binario\n" << endl;  
        cout << "Opciones:" << endl;  
        cout << "1.- Ingresar el valor de n manualmente." <<  
            endl;  
        cout << "2.- Ingresar el valor de n automaticamente." <<  
            endl;  
        cout << "3.- Salir." << endl;  
        cin >> opcion;  
        switch (opcion) {  
            case 1 :  
                system("cls");  
                manualmente();  
                system("pause");  
                break;  
            case 2 :  
                system("cls");  
                automaticamente();  
                system("pause");  
                break;  
            case 3:  opcion = 3; break;  
            default:  
                system("cls");  
                cout << "Favor de ingresar un valor  
                    valido" << endl;  
                system("pause");  
            }  
        }while(opcion !=3);  
    }  
}
```

La función manualmente(), donde pide al usuario el valor para n :

```
void manualmente() {
    int numero;
    cout << "Manualmente" << endl;
    cout << "Favor de ingresar el valor de n :"<<endl;
    cin >> numero;
    sigmaasterisco(numero);

    ContadorDeUnosATexto();//Se imprime las cantidades de 1 por
        cadena
}
```

La función automaticamente(), donde el valor de n se asigna de forma automatica:

```
void automaticamente() {
    int numero;
    srand(time(NULL));
    numero = rand() % 1001;//0 a 1000
    cout << "Automaticamente" << endl;
    cout << "El valor de n es:" << numero << endl;
    sigmaasterisco(numero);

    ContadorDeUnosATexto();//Se imprime las cantidades de 1 por
        cadena
}
```

La función que convierte un numero decimal a binario:

```
string Decimal_A_Binario(int numero){
    string binario = "";
    if(numero > 0) {
        while(numero > 0){
            if(numero%2 == 0){
                binario = "0" + binario;
            }else {
                binario = "1" + binario;
            }
            numero = (int) numero/2;
        }
    }else if(numero == 0){
        binario = "0";
    }
    return binario;
}
```

Funcion que calcula el limite de caracteres en la cadena, dependiendo la Σ :

```
int limitedeunos(int n){
    int lu=0;//limite de unos
    for(int i=0; i < n; i++){
        lu = ((2*lu) +1);
    }
    return lu;
}
```

Función que completa las cadenas de caracteres con 0, pues dependiendo el sigma, todas las cadenas deben de ser del mismo tamaño:

```
string llenar0(string cadena, int limite){
    int conta;

    int largo = cadena.size();

    string cadenacompleta = "";
    cadenacompleta = cadena;

    int cant0 = limite-largo;
    if (largo < limite){
        for(int i = 0; i < cant0;i++){
            cadenacompleta = "0" + cadenacompleta;
        }
    }
    return cadenacompleta;
}
```

Función que calcula Σ^* , dependiendo el valor de n:

```
void sigmaasterisco(int k){

    fstream archivo;
    archivo.open("SigmaUniverso.txt",ios::out);
    for(int contador=0;contador <= k; contador++){

        int limite = contador;
        string binario="";

        for(int conta = 0; conta <= limitedeunos(limite);conta
            ++){
```

```

        binario = Decimal_A_Binario(conta);

        if (limitedeunos(limite) == 0){
            archivo << "{"<<"e";
        }
        if(binario.size() < limite){
            archivo <<","<<llenar0(binario,limite);
        }
        if(binario.size() == limite){
            archivo <<","<< binario ;
        }
    }

}
archivo << "}.|";
archivo.close();
}

```

Función que cuenta cada 1 que existe en cada cadena de cada Σ y guarda las cantidades en un archivo de texto:

```

void ContadorDeUnosATexto(){
    cout << "Contando cantidad de 1 en cada cadena..." << endl;

    fstream archivo;
    archivo.open("SigmaUniverso.txt",ios::in);
    string linea;
    if (archivo.is_open())
    {
        fstream archivodeunos;
        archivodeunos.open("CantDeUnos.txt",ios::out);
        while(!archivo.eof())
        {
            getline(archivo,linea);
            long long int conta = 1;
            int cantidadunos = 0;
            archivodeunos << "X" << endl;
            while((linea[conta] != '.')){
                if (linea[conta] == '1'){
                    cantidadunos++;
                }
                if (linea[conta] == ',' ){
                    archivodeunos << cantidadunos
                        << endl;
                }
            }
        }
    }
}

```

```

        cantidadunos = 0;

    }
    if (linea[conta] == '.' ){

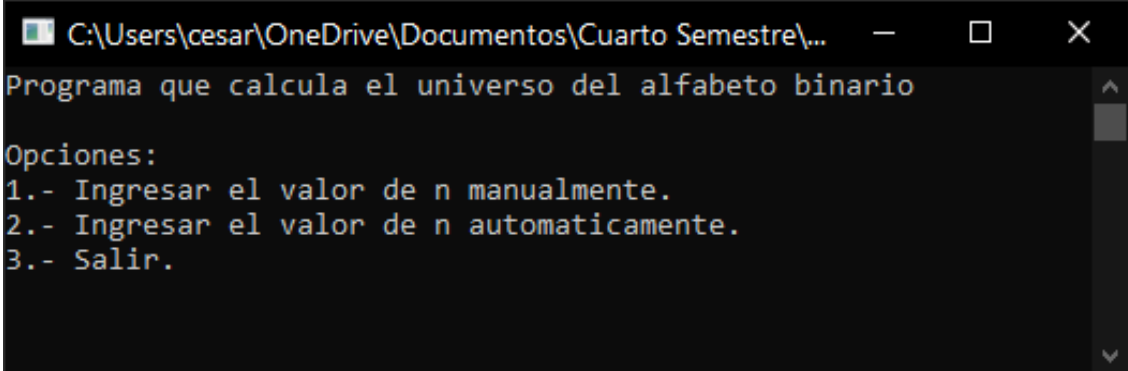
        archivodeunos << cantidadunos
        << endl;
        cantidadunos = 0;

    }
    conta++;
}
//cout << conta << endl;
}
archivodeunos.close();
}
archivo.close();
cout << "Archivo Creado." << endl;
}

```

3.3. Ejecución del programa, en Dev-C++

Primera parte, el menú:



The screenshot shows a Windows command prompt window with the title bar "C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\...". The window contains the following text:

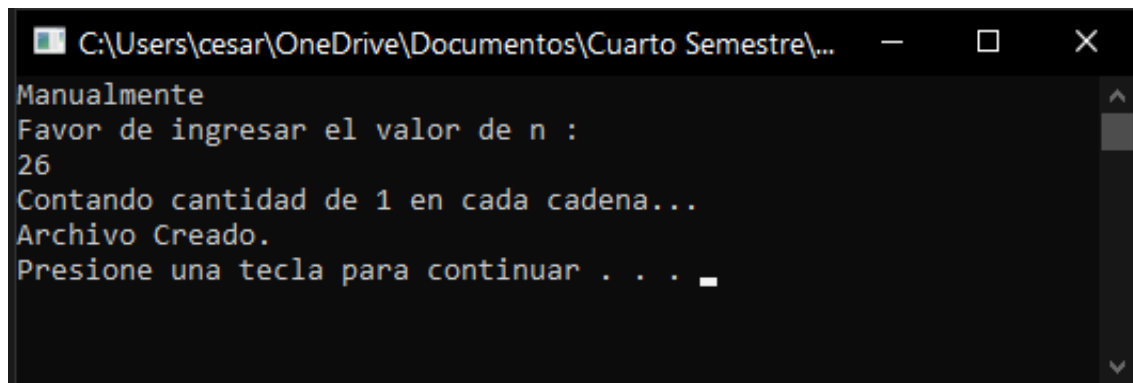
```

Programa que calcula el universo del alfabeto binario

Opciones:
1.- Ingresar el valor de n manualmente.
2.- Ingresar el valor de n automaticamente.
3.- Salir.

```

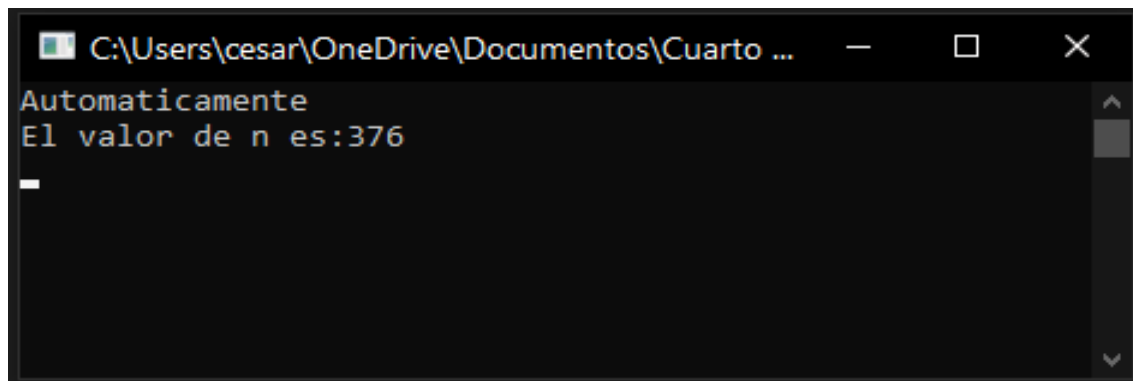

Cuando se tiene que ingresar n manualmente:



```
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\...
Manualmente
Favor de ingresar el valor de n :
26
Contando cantidad de 1 en cada cadena...
Archivo Creado.
Presione una tecla para continuar . . .
```

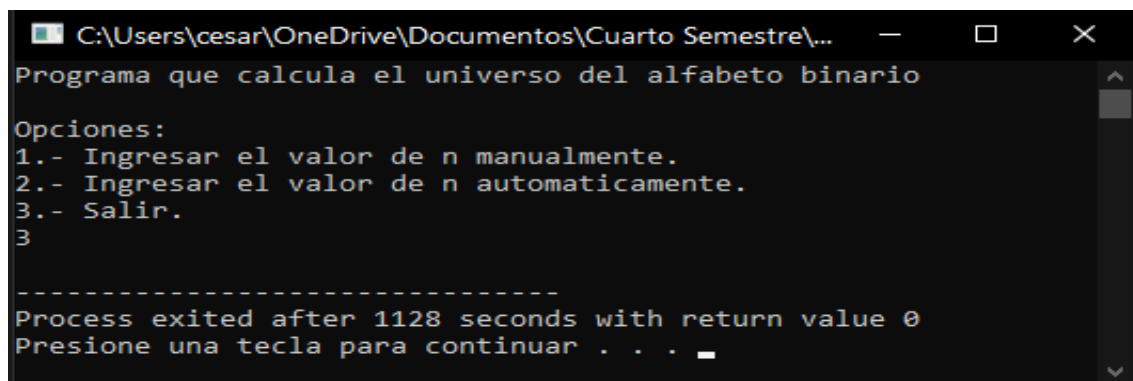
Cuando n se tiene que calcular automáticamente:

(Solo es una captura de ejemplo porque calcular esto se tardaría mucho tiempo)



```
C:\Users\cesar\OneDrive\Documentos\Cuarto ...
Automaticamente
El valor de n es:376
_
```

Al finalizar la ejecucion:



```
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\...
Programa que calcula el universo del alfabeto binario

Opciones:
1.- Ingresar el valor de n manualmente.
2.- Ingresar el valor de n automaticamente.
3.- Salir.
3

-----
Process exited after 1128 seconds with return value 0
Presione una tecla para continuar . . .
```

Los archivos creados,(Tienen el icono de notepad++ porque es la aplicación que uso para .txt).



Para poder abrir los archivos se instalo la aplicación EmEditor que si fue capaz de abrir los archivos:

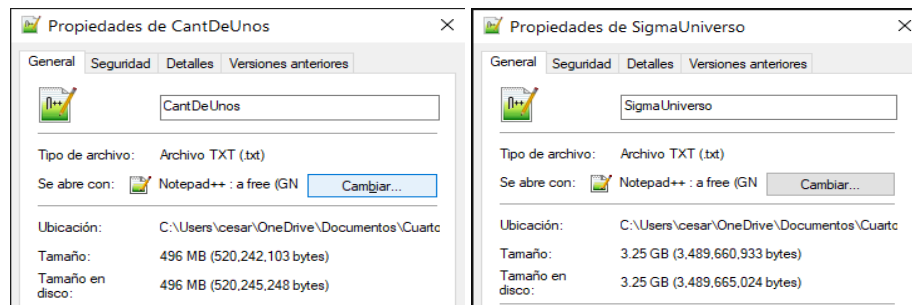
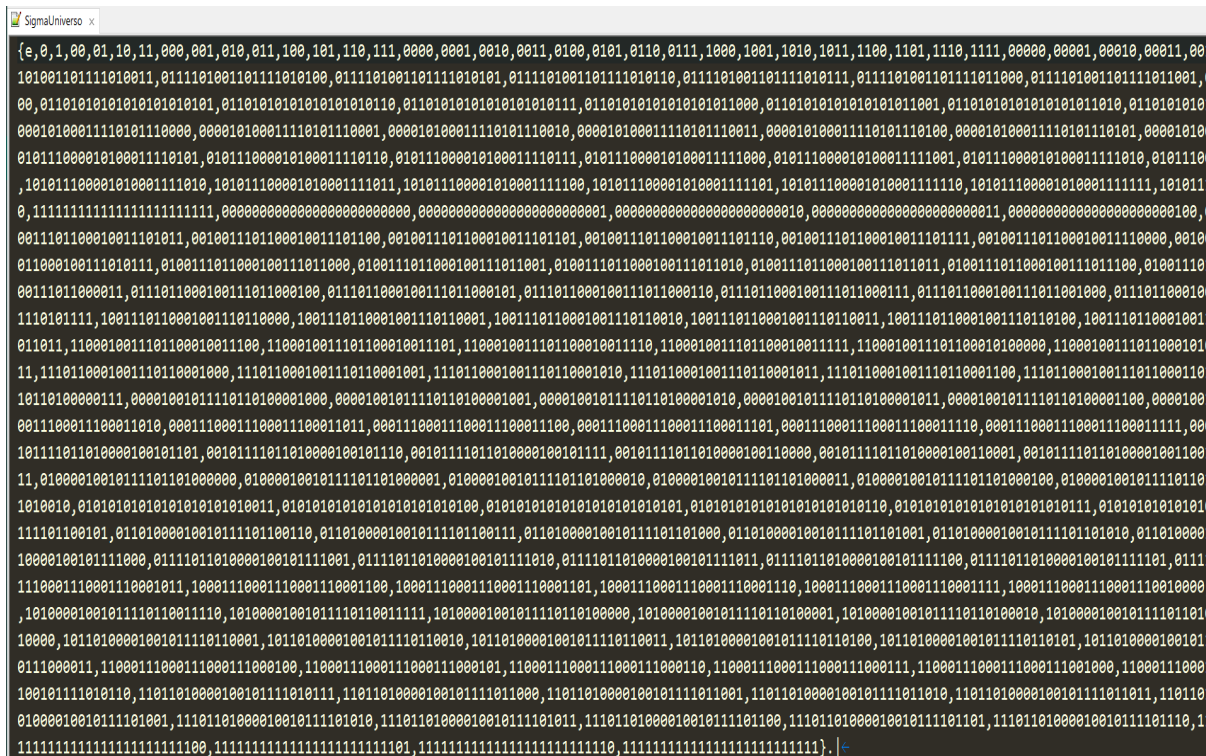


Imagen de una parte de SigmaUniverso.txt abierto en EmEditor



3.4. Código de programa para la gráfica en Phyton

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('CantDeUnos.txt', header=1, delim_whitespace=True)

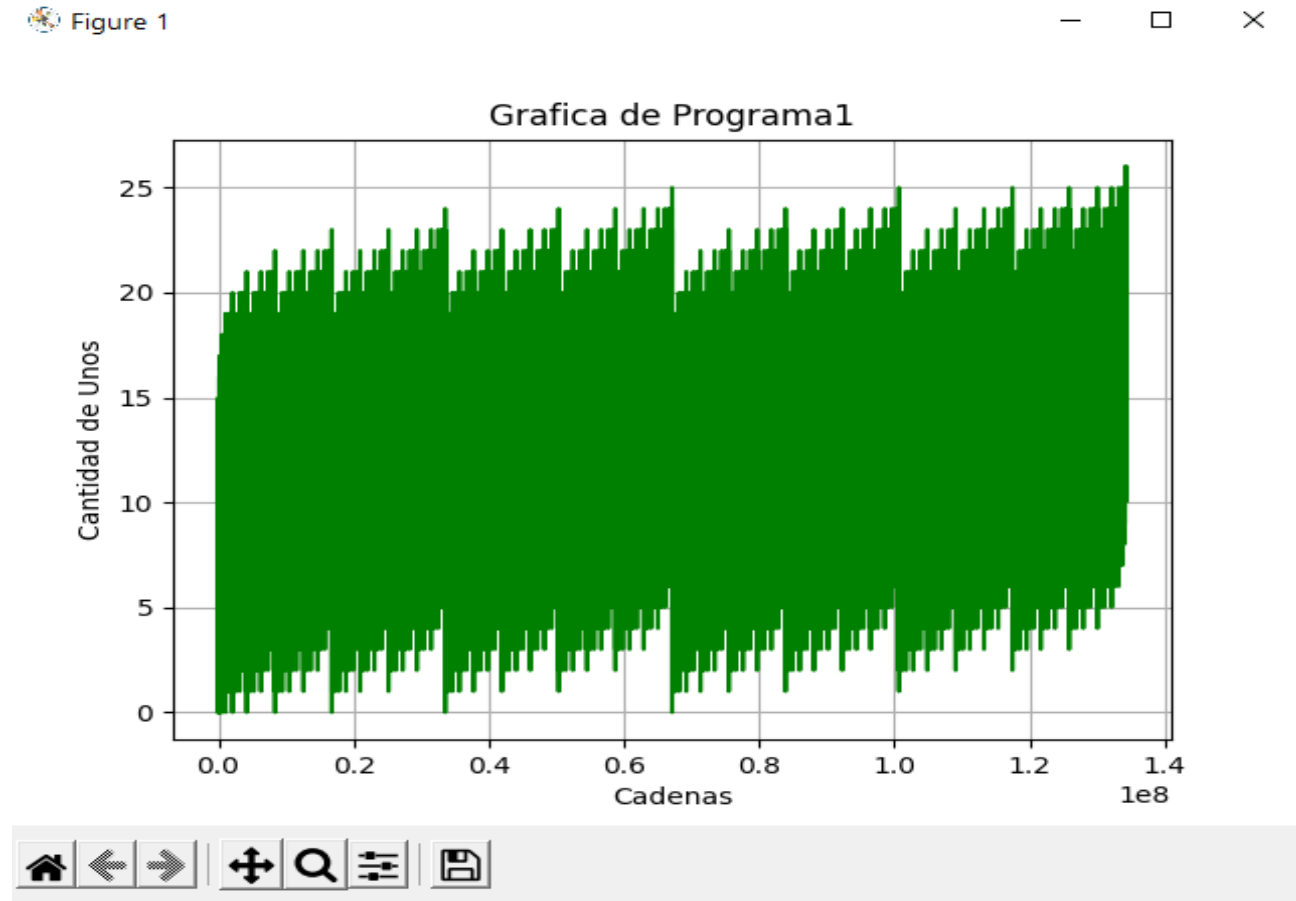
print(data)
plt.title("Grafica de Programal")
plt.xlabel("Cadenas")
plt.ylabel("Cantidad de Unos")
plt.grid(True)
plt.plot(data, "g")
plt.show( )
```

3.5. Salida por consola en PyCharm

```
GraficaPractica1 x
C:\Users\cesar\AppData\Local\F
0
0 0
1 1
2 0
3 1
4 1
... ..
134217721 25
134217722 24
134217723 25
134217724 25
134217725 26

[134217726 rows x 1 columns]
```

3.6. Gráfica



4. Conclusiones

Este programa fue un reto poder hacerlo pues no había trabajado con graficacion , y menos con Phyton entonces fueron muchas cosas nuevas las que aprendí en esta primer programa, la parte de C++ fue un poco engorrosa pues ya tenia tiempo que no escribia codigo asi de modular como lo tuve que hacer en este programa, pero en c++ esta vez no aprendi cosas nuevas sino que fue bueno para recordar al lenguaje y practicarlo un poco mas.