

Fractales

INTRODUCCIÓN A LA PROGRAMACIÓN

RECURSIVIDAD

PROFESOR: CARLOS BENAVIDES

Jason Latouche Jiménez

2015156294

TECNOLÓGICO DE COSTA RICA | ESCUELA DE COMPUTACIÓN

Contenido

Introducción.....	2
Objetivos del proyecto	2
¿Qué son fractales?	3
Copo de Koch	3
Árbol	4
Alfombra de Sierpinski.....	4
Curva del Dragón	5
Manual de Usuario	6
Conclusiones.....	15
Logros	15

Introducción

Este trabajo fue realizado con el propósito de familiarizarse con la programación de recursión y el término de fractales. A lo largo de este documento se explicará cómo funcionan los fractales que en esta tarea se presentan, se presentará un manual de usuario y se dará una breve conclusión. Además también se presentará documentación interna y externa sobre lo que son los fractales a que se deben su nombre y formas.

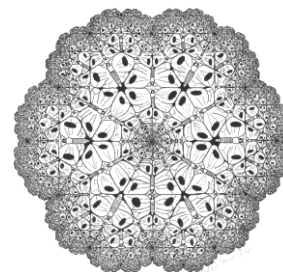
Objetivos del proyecto

- Familiarizarse con la programación por recursión.
- Familiarizarse con el término fractales y que son.
- Incorporar el concepto fractales con la programación mediante la recursión.

¿Qué son fractales?

Es un fenómeno matemático en donde se exhiben patrones repetitivos en una figura a diferentes escalas. Estas figuras se diferencian de otras por la forma que crean a un mismo dibujo sin importar el tamaño al que se vea.

Aunque ya hacía muchos años, durante el siglo 17 se habían adoptado estas figuras, no fue hasta el siglo 19 que el matemático Benoît Mandelbrot adopta el término de fractal. Esta palabra deriva del latín *fractus*, que significa quebrado o fracturado.



Aunque existen muchos fractales de formas muy interesantes, en esta tarea tan solo se presentan 4 fractales creados por matemáticos: el copo de Koch, el Árbol, la Alfombra de Sierpinski y la curva del Dragón. Además, se presentan como añadidura dos fractales más inventados incluidos en esta tarea.

Antes, mostraremos como funciona cada fractal. Es necesario mencionar que existen fractales que funcionan mejor con el Sistema-L. Este es un sistema presentado por el botánico Aristid Lindenmayer en 1968, con el objetivo de explicar matemáticamente el crecimiento de las plantas. Se basa en un sistema de funciones, pero veremos más detenidamente su funcionamiento dentro de cada fractal que lo use.

Copo de Koch

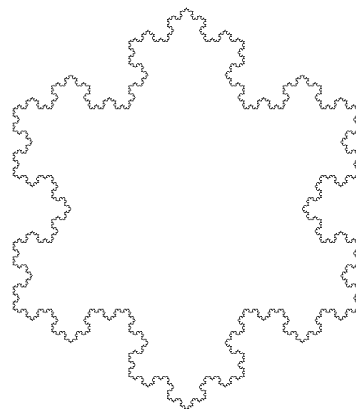
El copo de nieve de Koch, también llamado estrella de Koch, es una curva cerrada continua pero no diferenciable en ningún punto descrita por el matemático sueco Helge von Koch en 1904 en un artículo titulado "Acerca de una curva continua que no posee tangentes y obtenida por los métodos de la geometría elemental".

Este copo está basado en la curva de Koch, que es en realidad una línea en donde, con cada iteración, se crean un triángulo con una base de un tercio del tamaño de la línea original anterior. Después, se usa cada lado del triángulo para crear la siguiente iteración, y así sucesivamente.

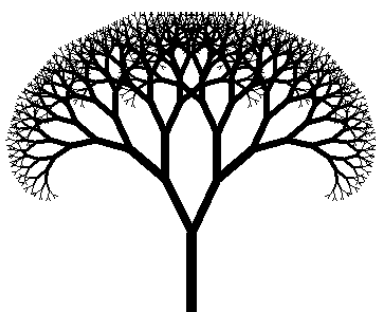


Cuando las iteraciones son suficientes su forma se vuelve muy densa de triángulos, lo que forma la famosa curva de Koch. Si se toma tres veces la línea original base, se unen formando un triángulo con ellas, y además se incrementa las interacciones a un número considerable, se formaría una figura muy parecida a un copo de nieve, la cual se conoce como el copo de Koch.

La figura adjunta representa el copo de Koch en la 6ta iteración. Como se puede observar, cada triángulo contiene un triángulo de una tercera parte del tamaño anterior. Haciendo este mismo procedimiento 6 veces, se forma la figura de la derecha.



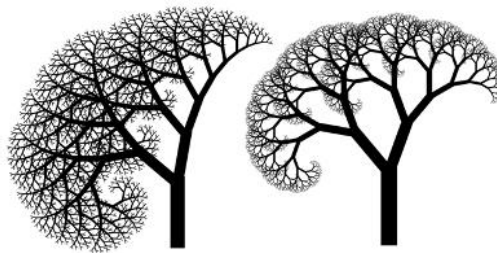
Árbol



La figura del árbol es realmente muy sencilla. Esta solo toma un tronco principal en una primera iteración, luego lo divide en dos con un corte en el tamaño y el grosor del anterior. En cada iteración, las ramas derivan de las anteriores a un tamaño menor y tomando el ángulo de referencia que distancian a ambas ramas principales como guía para continuar dibujando.

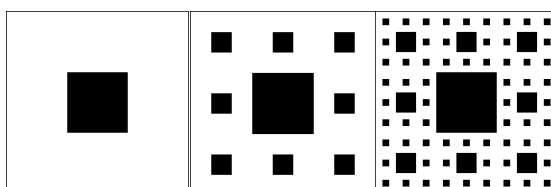
Este método de dibujo es muy usado en video juegos, en donde se renderiza un árbol basado en la figura inicial, o el tronco base del árbol. Este método ahorra muchas líneas de código, y se pueden crear diferentes tipos de árboles jugando con los parámetros.

Jugar con los parámetros del árbol, como el ángulo de apertura, el tamaño de las ramas por separado, producen arboles con mayor movimiento y mucho mas similares a la realidad.

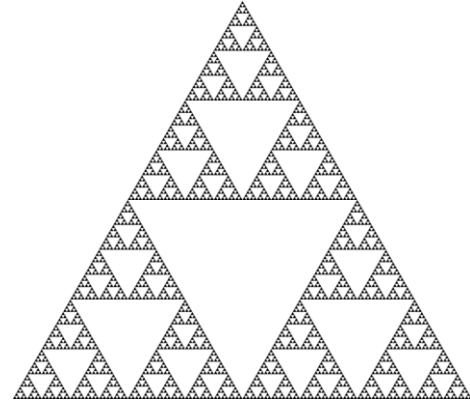


Alfombra de Sierpinski

Esta peculiar alfombra es un conjunto fractal descrito por primera vez por Waclaw Sierpiński en 1916. Constituye una generalización a dos dimensiones del conjunto de Cantor. Comparte con él muchas propiedades: también es un conjunto compacto, no numerable y de medida nula.

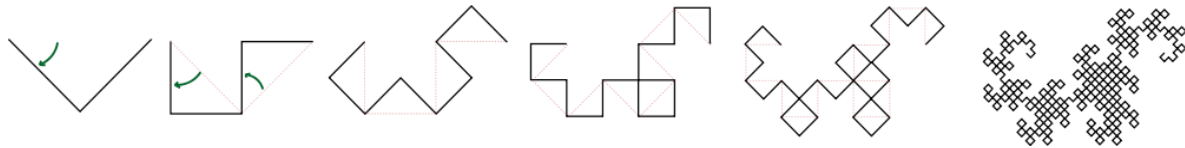


La construcción de la alfombra de Sierpinski se inicia de forma recursiva con un cuadrado. El cuadrado se corta en 9 cuadrados congruentes, y eliminamos el cuadrado central. El paso anterior vuelve a aplicarse recursivamente a cada uno de los 8 cuadrados restantes. La alfombra de Sierpinski es el límite de este proceso tras un número infinito de iteraciones. Con este método también se pueden crear otros fractales, como la alfombra tipo triángulo que se encuentra a la derecha de este texto. Esta se presenta en la 5ta recursión o iteración.

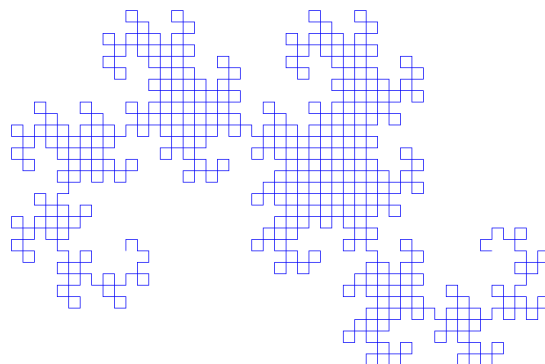


Curva del Dragón

La curva del Dragón es un miembro de las propias semi curvas de fractal. Este es uno de los fractales que pueden ser simulados con el Sistema-L. Esa extraña figura se forma a partir de un segmento, se construye el triángulo rectángulo e isósceles, como lo muestra las dos primeras figuras. Luego se borra el segmento inicial. Se repite un sinnúmero de veces el proceso de remplazar un segmento por otros dos para cada línea de la curva, alternando siempre la orientación de los triángulos.



Usar el Sistema-L para renderizar esta extraña figura facilita la forma del cómo se programa en código. Este sistema fue propuesto por el botánico Aristid Lindenmayer con el objetivo principal de modelar el proceso de crecimiento de las plantas. Usa un conjunto de reglas y símbolos como las siguientes: $(X \rightarrow X+YF+)$, $(Y \rightarrow -FX-Y)$, en donde “-” significa giro de 90° a la izquierda, “+” es un giro de 90° a la derecha, mientras que “F” quiere decir hacia adelante. Si se toma el inicio de FX, cada vez que entre a la función “X” o “Y”, la recursión disminuye en 1. Este es el fractal finalizado a la iteración 10.



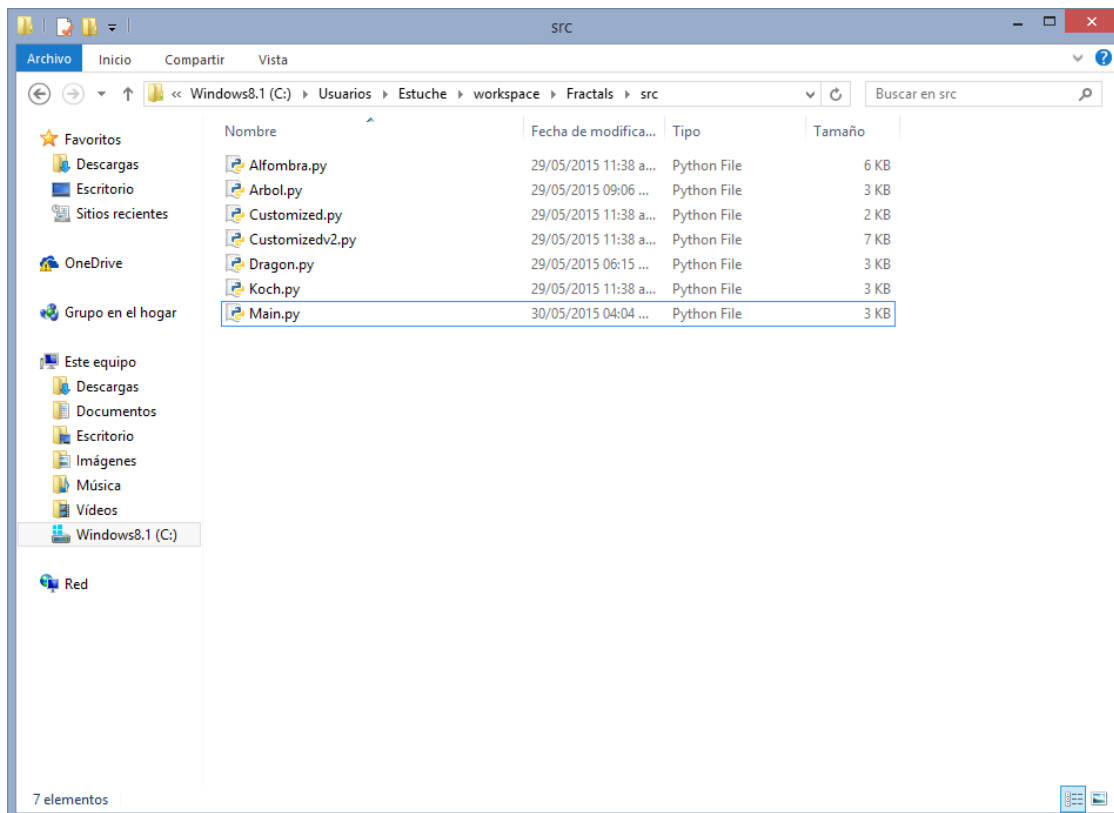
Manual de Usuario

Usar estos fractales programados es muy sencillo. Primero hablaremos sobre los requisitos del equipo para poder correrlos. Luego mencionaremos como verlos en funcionamiento y los parámetros recomendados para ejecutarlos y poder verlos bien sin ningún problema.

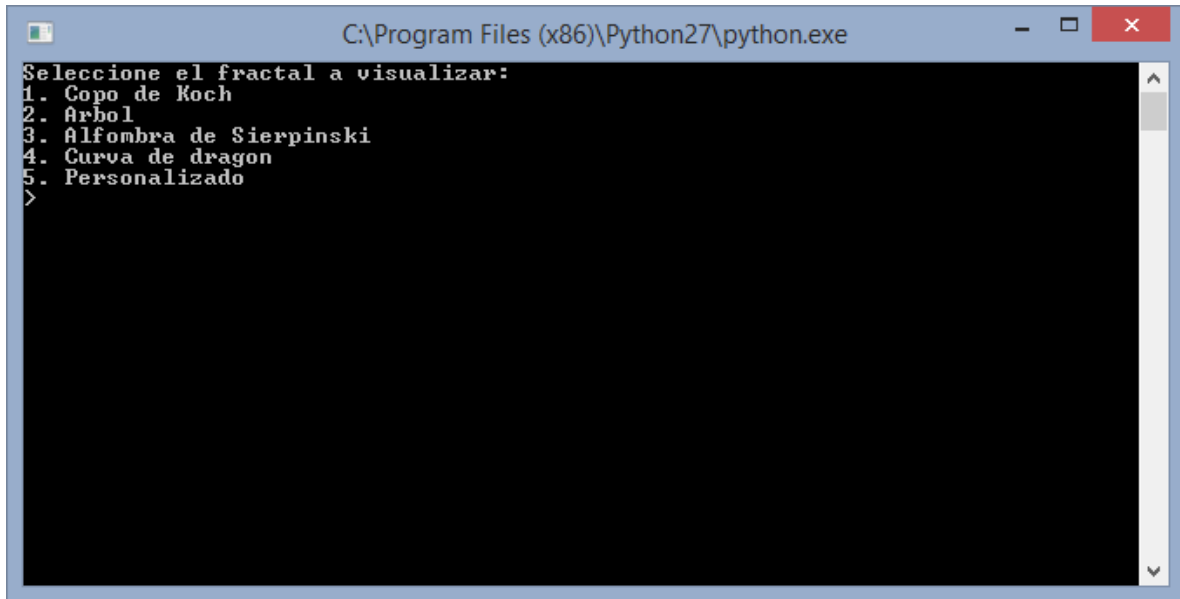
Primeramente la computadora debe tener instalado Python en su versión 2.7.x. En este caso “x” quiere decir que no importa la subversión que se esté usando. Como complemento, también deben estar instaladas las librerías del generador de gráficos Turtle.

En el caso que se usa un sistema operativo basado en Linux como Debian o Ubuntu, Turtle puede ser fácilmente instalado por el comando *“sudo apt-get install python-tk”*. En el caso que se disponga de un sistema operativo de Microsoft, como Windows, este ya vendrá instalado automáticamente en el sistema.

Teniendo los pasos anteriores completamente listos, ya podemos proceder a ejecutar los fractales elaborados en Python. Para ello vamos a descomprimir el archivo Fractales.zip en un directorio que nos convenga. Podemos descomprimirlo en el Escritorio o en el mismo directorio que se encuentra el archivo .zip. Entramos a la carpeta y veremos algo así:

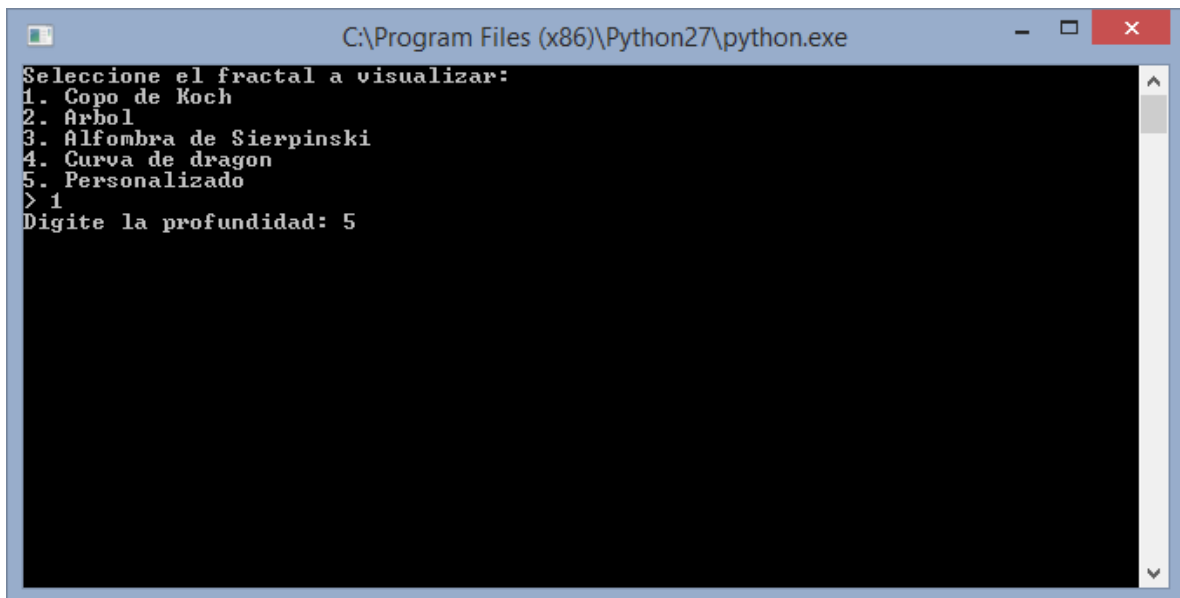


Aunque ahí podamos ver todos los archivos de los fractales, tan solo vamos a ejecutar el que se llama "Main.py". Este es nuestro archivo principal que nos lanzará a los demás fractales, por lo que no nos servirá de nada abrir cada uno individualmente. Damos doble clic a "Main.py" y se nos abrirá una ventana como la siguiente:



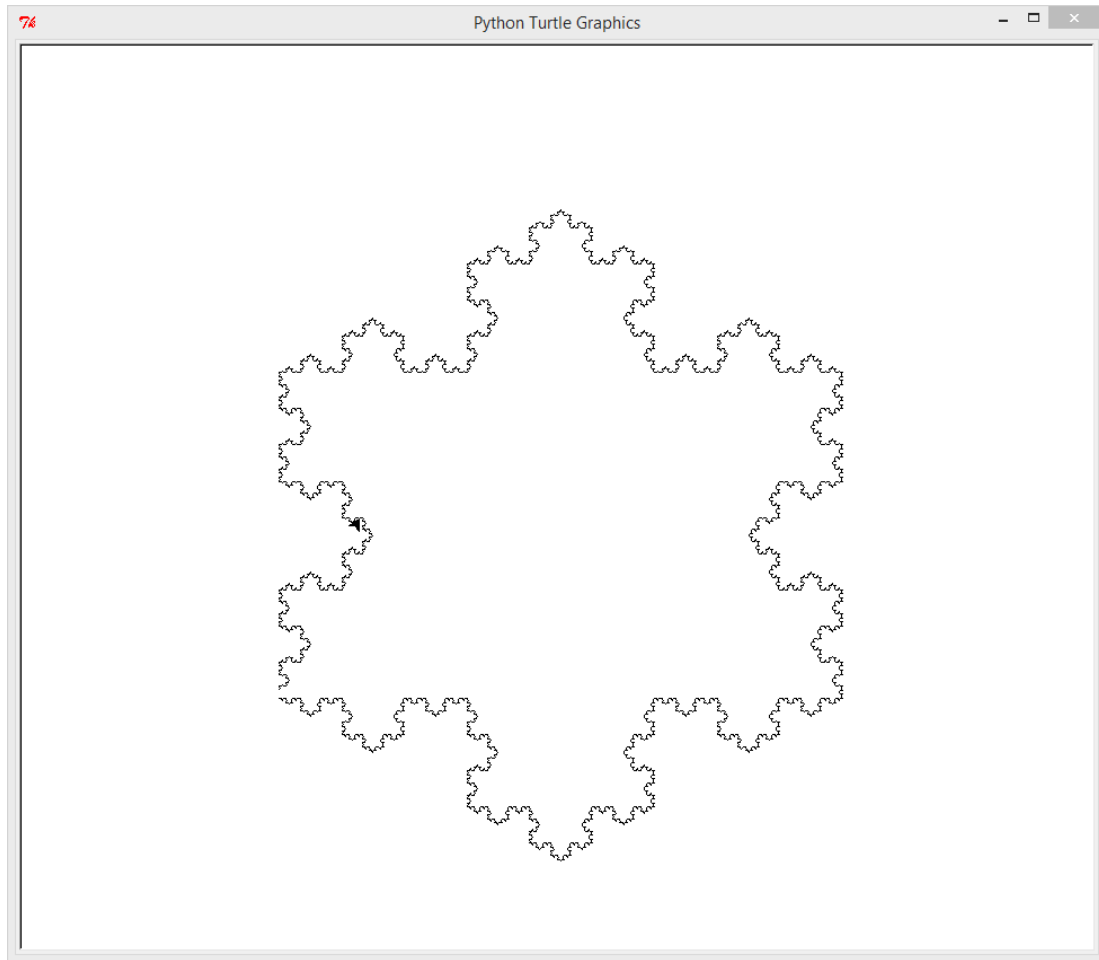
```
C:\Program Files (x86)\Python27\python.exe
Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
>
```

Acá escribiremos un número del 1 al 5 para abrir cualquier fractal que queramos. Entre las restricciones encontramos que no debemos ingresar letras. Vamos a ingresar el número 1.



```
C:\Program Files (x86)\Python27\python.exe
Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
> 1
Digite la profundidad: 5
```

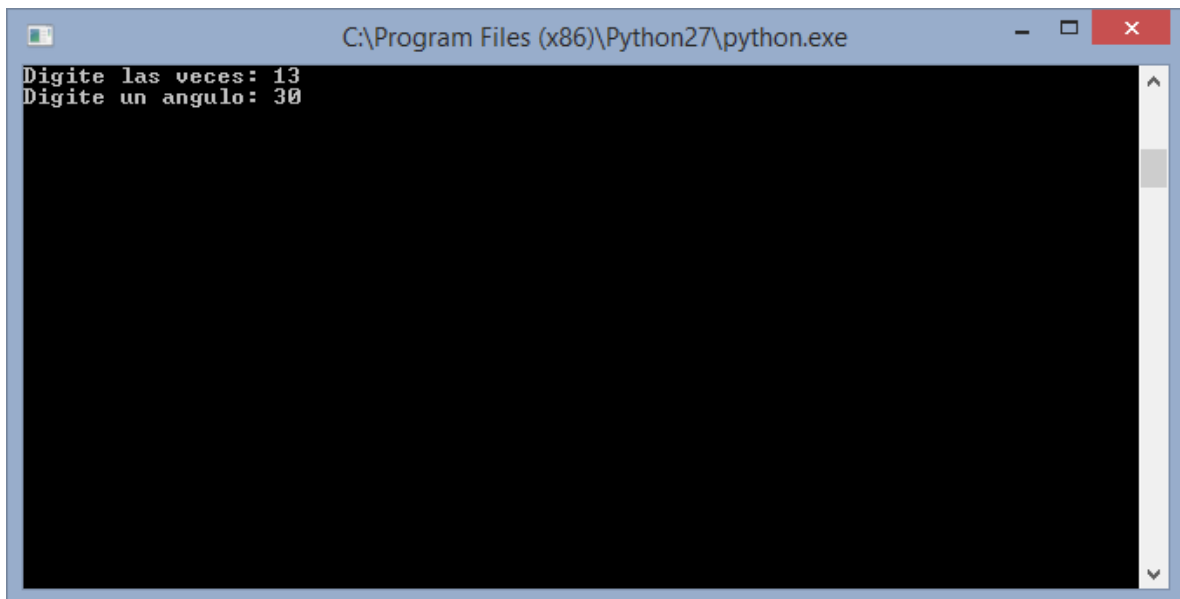
Nos pregunta por la profundidad. Podemos ingresar cualquier número que queramos, pero el recomendado es 5. Se abrirá una ventana y veremos el fractal dibujado como la siguiente:



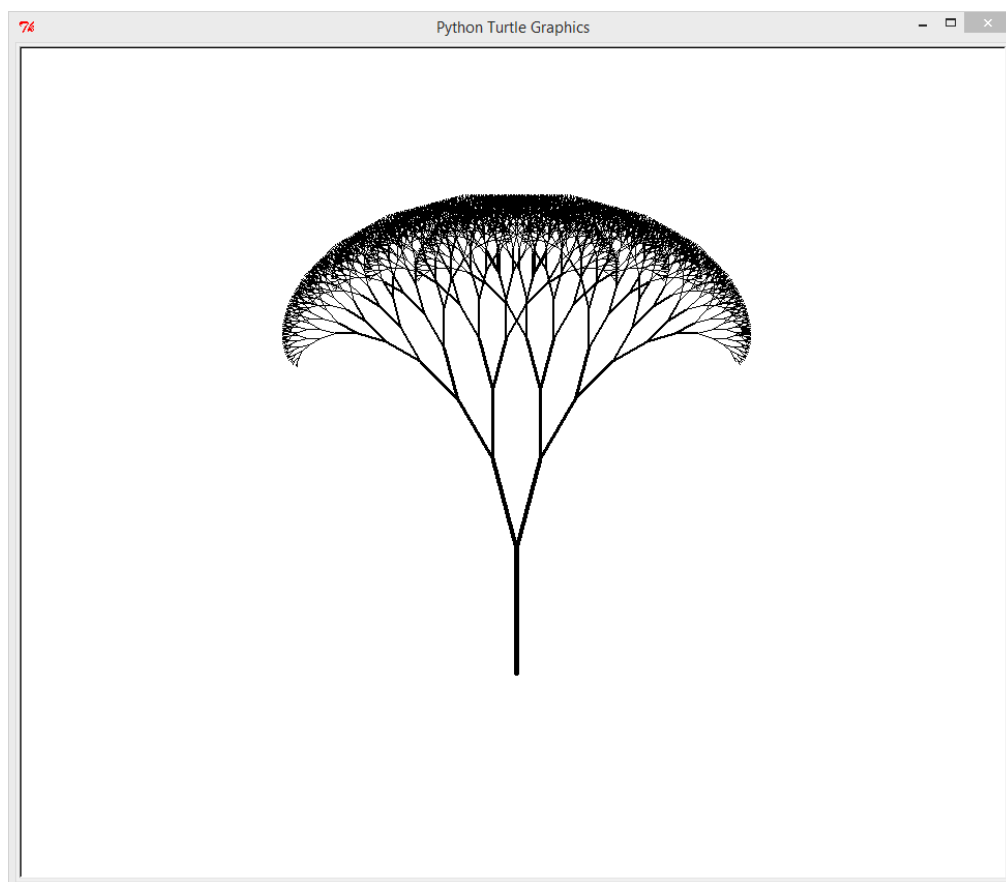
Podemos jugar con la profundidad para ver cómo se forman diferentes figuras. Cuando cerremos esta ventana se nos preguntará si queremos ver otro. Responderemos con una “S” para sí, y con una “N” para no. Pondremos que “Si”.

```
C:\Program Files (x86)\Python27\python.exe
Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
> 1
Digite la profundidad: 5
Desea visualizar otro fractal S/N:
> s
```

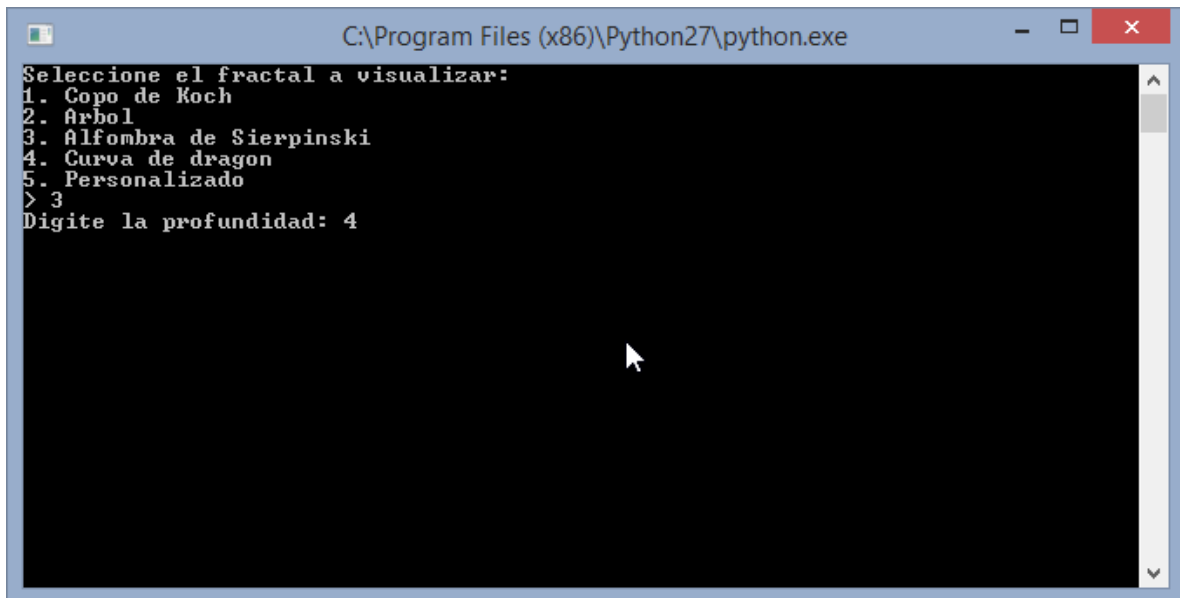
Ahora pondremos el fractal número 2. Y nos preguntará por las veces, en donde yo recomiendo 13. También pregunta por el ángulo de apertura entre dos ramas, en donde el valor recomendado es 30.



Se formará la siguiente figura:

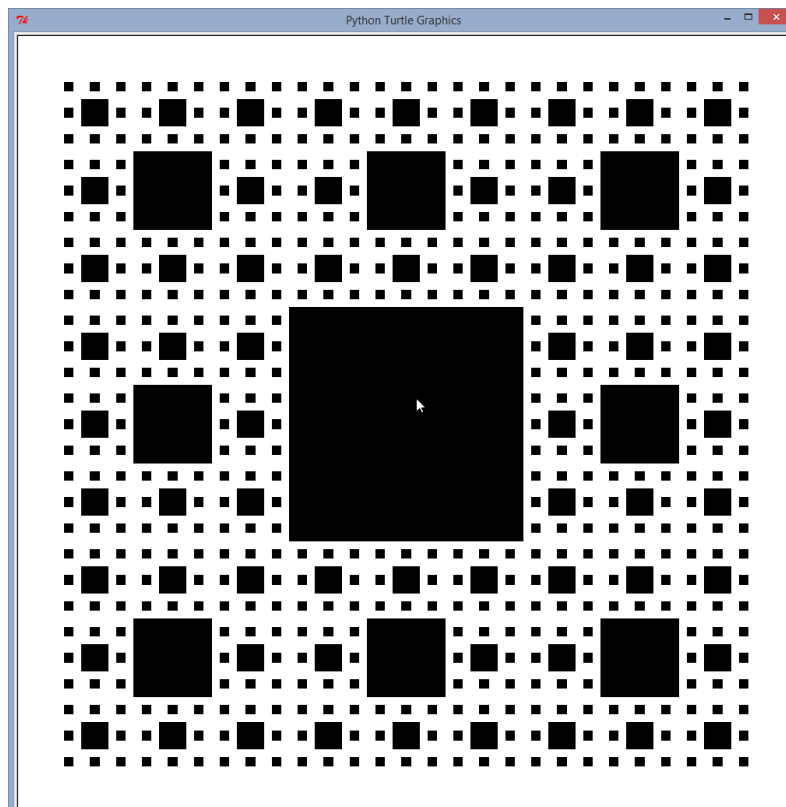


Jugando con el valor del ángulo se pueden formar figuras muy curiosas. Cuando pregunte si desea ver otro fractal, dale que Si, pon 2 otra vez e intenta usar 45, 90 y 180 grados como parámetro para el árbol. Cuando ya termines de jugar con el árbol, podremos pasar de ver los demás fractales. Ahora veremos el funcionamiento del fractal número 3.



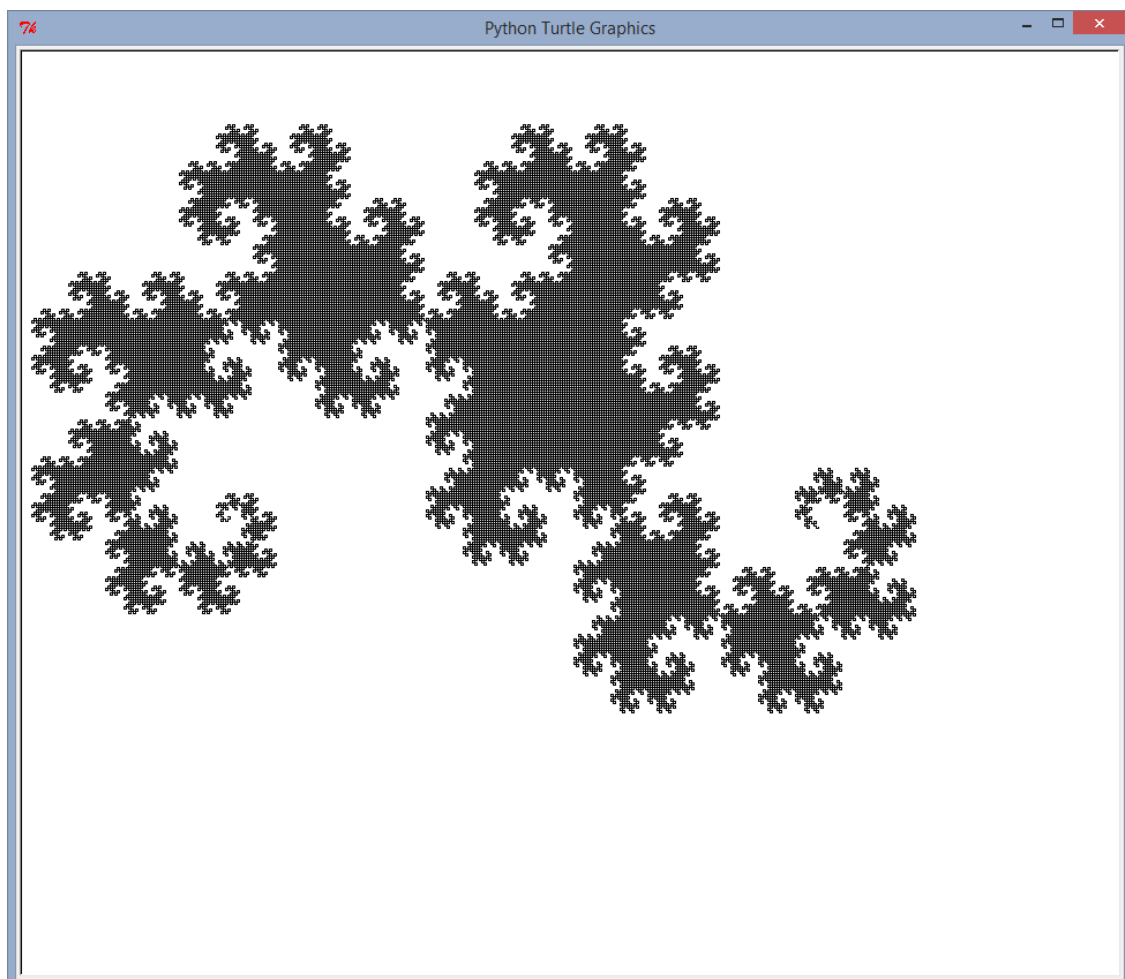
```
C:\Program Files (x86)\Python27\python.exe
Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
> 3
Digite la profundidad: 4
```

Veremos que nos pregunta por el número de profundidad que queremos ver, en este caso le daremos 4, la cual genera una imagen muy agradable a la vista, como la siguiente:



```
C:\Program Files (x86)\Python27\python.exe
Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
> 4
Digite la profundidad: 16
```

Ahora veremos el funcionamiento del fractal número 4. Hacemos lo mismo. Cuando nos pregunte por la profundidad, colocaremos 16, la cual es la recomendada. Veremos una figura como la siguiente:

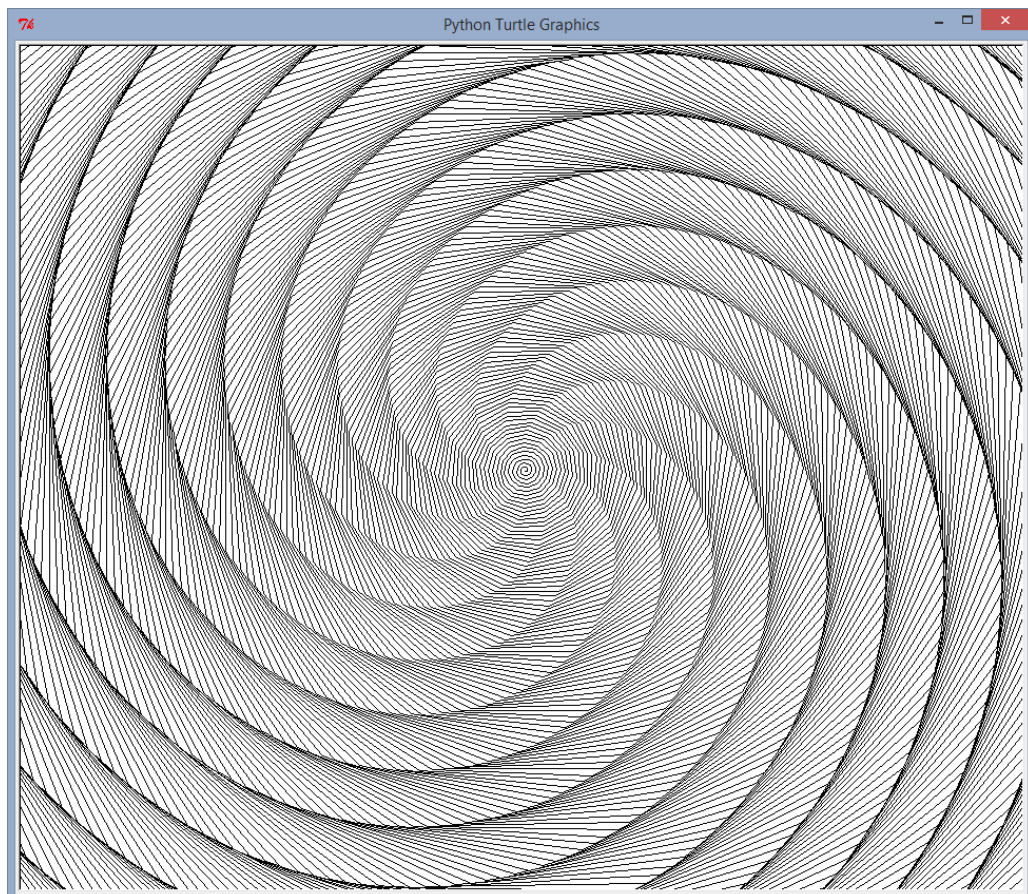


Veremos el funcionamiento de los fractales personalizados. Si damos 5 nos aparecerá un submenú para ver el fractal 1 y 2. Igual que el menú anterior, no ingrese letras. El uno nos preguntará por una profundidad, en donde colocaremos 500.

```
C:\Program Files (x86)\Python27\python.exe

Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
> 5
1. Version 1
2. Version 2
> 1
Digit la profundidad: 500
```

Se generará una figura muy divertida como es la siguiente:

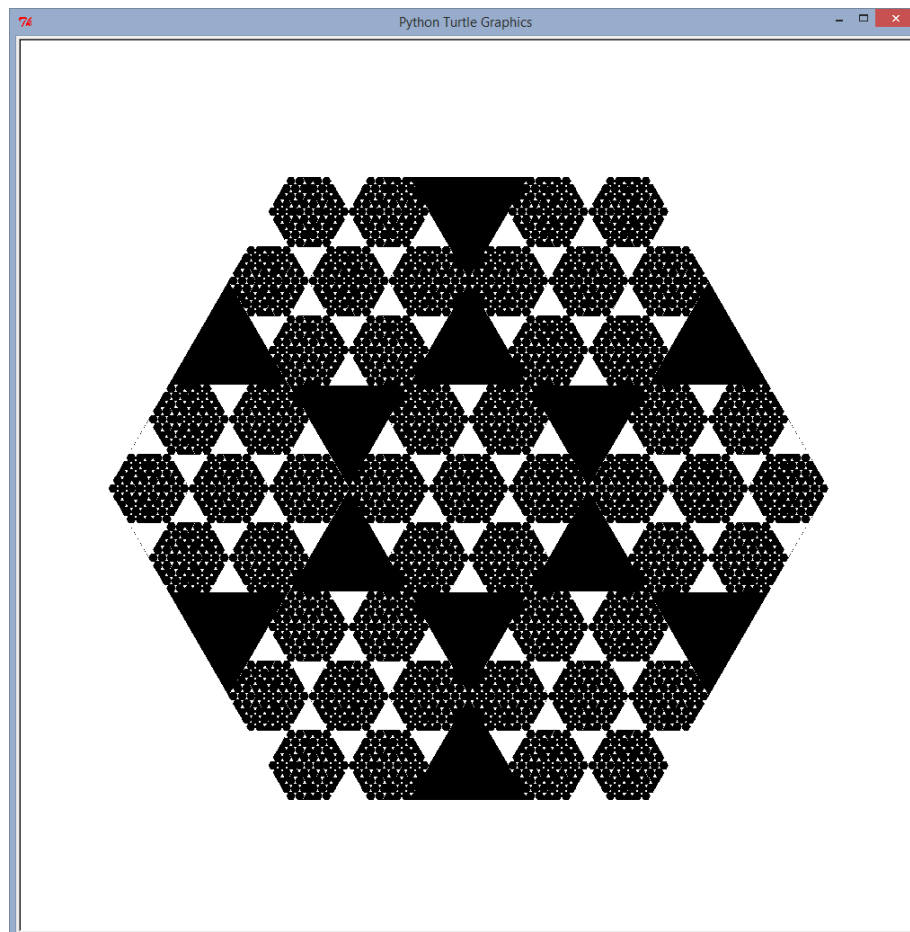


Si generamos el fractal número 2, igual nos preguntará por una profundidad, en donde colocaremos una de tamaño 5.

```
C:\Program Files (x86)\Python27\python.exe

Seleccione el fractal a visualizar:
1. Copo de Koch
2. Arbol
3. Alfombra de Sierpinski
4. Curva de dragon
5. Personalizado
> 5
1. Version 1
2. Version 2
> 2
Digite la profundidad: 5
```

Nos generará la siguiente imagen:



Estos dos últimos fractales están basados en hexágonos, por lo que podemos generar un hexágono que crezca, como en el primero, o varios hexágonos que estén dentro del principal. Este último genera triángulos, lo que hace que se vuelva una figura realmente curiosa.

Conclusiones

Durante esta tarea se aprendió el funcionamiento de los fractales, además promovió la investigación de métodos para programar fácilmente estos fractales. Usar interfaz gráfica para simular cada uno de ellos nos condujo a la investigación de código, lo cual ayudará no solo para esta tarea o este curso, sino también para futuras investigaciones que se deban realizar acerca de estos temas.

Logros

Entre los logros que se realizaron están:

1. Logran generar una interfaz gráfica amigable para simular los fractales.
2. Importar librerías de cada interfaz gráfica.
3. Generar un código fuente a base recursiva para crear los fractales.
4. Se lograron crear todos los fractales solicitados.
5. Incluso se amplió los fractales inventados a dos.
6. Se promueve a generar un menú fácil de entender para un usuario básico.