|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Tapete Sierpinski** | | | **No.** | **1** |
| **Asignatura:** | **Grafican** | **Carrera:** | **Ing Sistemas Computacionales** | **Duración de la práctica (Hrs)** |  |

**Patricia Zaragoza Palma**

**II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):**

* **Aula**
* **Casa**

**III. Material empleado:**

* Computadora
* Visual Studio Code

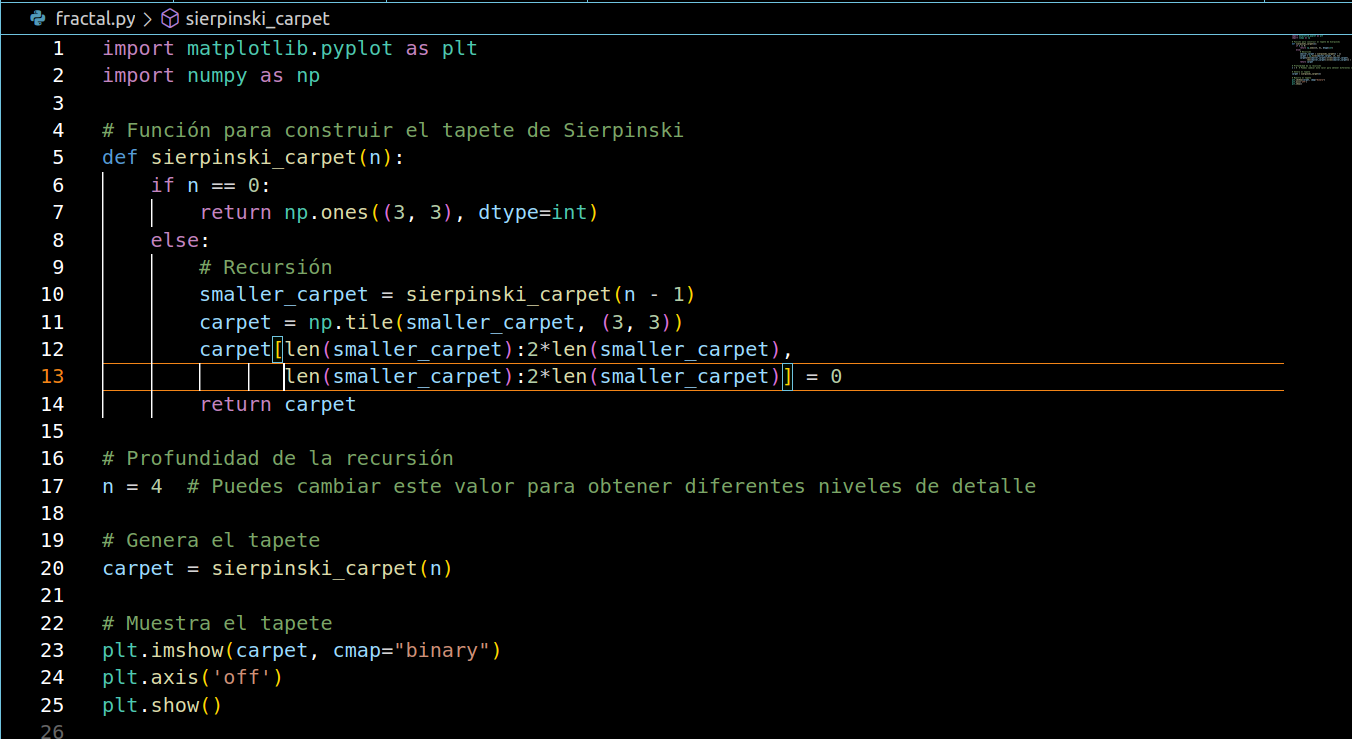
**IV. Desarrollo de la práctica:**

En esta practica se llevara acabo el tapete de Sierpinski

Para ello primeramente se va a crear una clase en Python.

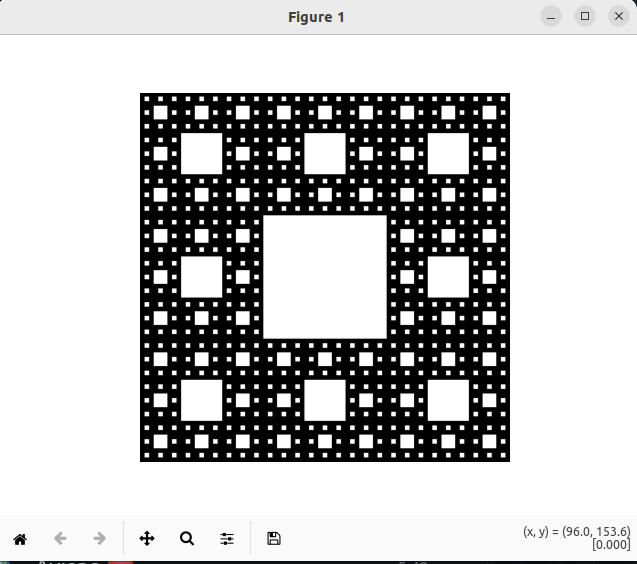


Después de ello se va a desarrollar nuestro código.



* Importa la biblioteca matplotlib.pyplot, que se utiliza para crear gráficos y visualizaciones en 2D. Se importa con el alias plt para simplificar el código.
* mporta la biblioteca numpy, que se usa para operaciones matemáticas y manejo de arrays en Python. Se importa con el alias np.
* def sierpinski\_carpet(n):: Define una función llamada sierpinski\_carpet que toma un argumento n, el cual representa el nivel de profundidad de la recursión para construir el tapete de Sierpiński.
* if n == 0:: Verifica si n es 0. Este es el caso base de la recursión.
* return np.ones((3, 3), dtype=int): Si n es 0, devuelve una matriz de 3x3 llena de unos, que representa el nivel más básico del tapete de Sierpiński.
* else:: Si n no es 0, se ejecuta el bloque de código para el caso recursivo.
* smaller\_carpet = sierpinski\_carpet(n - 1): Llama recursivamente a la función sierpinski\_carpet con n - 1 para obtener el patrón del tapete de Sierpiński de un nivel inferior.
* carpet = np.tile(smaller\_carpet, (3, 3)): Usa np.tile para replicar el patrón de smaller\_carpet en una matriz 3x3. Esto construye una matriz que contiene 9 copias del patrón más pequeño.
* carpet[len(smaller\_carpet):2\*len(smaller\_carpet), len(smaller\_carpet):2\*len(smaller\_carpet)] = 0: Establece el valor de una submatriz central en 0 (vacío) dentro de la matriz más grande. Esta operación elimina el centro del tapete, creando el efecto de "vacío" en el patrón de Sierpiński.
* return carpet: Devuelve la matriz completa del tapete de Sierpiński para el nivel n.

Después de ello se ejecuta nuestro proyecto y debe de aparecer de la siguiente manera.



**V.Conclusiones:**

Un *fractal* es un ente geométrico el cual en su desarrollo espacial se va reproduciendo a si mismo cada vez a una escala menor. Una característica esencial de los fractales consiste en que si se observara a través de una lupa, una parte cualquiera del mismo, ésta reproduce a escala menor la figura total del fractal.

Este fractal se construye partiendo de un cuadrado de lado uno, se divide en tres partes iguales cada lado y se extrae el cuadrado central. En cada uno de los ocho cuadrados de lado  que forman la figura restante se repite esta operación, y así sucesivamente.

La alfombra de Sierpinski es el conjunto que queda después de este proceso infinito. La visualización del fractal se realiza utilizando matplotlib, que muestra el patrón resultante en una representación gráfica en blanco y negro. A medida que se incrementa el número de iteraciones (nivel de recursión), el tapete revela un mayor nivel de detalle y complejidad, demostrando la naturaleza infinita y auto-similar de los fractales.