Explorando los secretos de la geometría y los fractales con python

🦄@marisbotero

¡Hola!

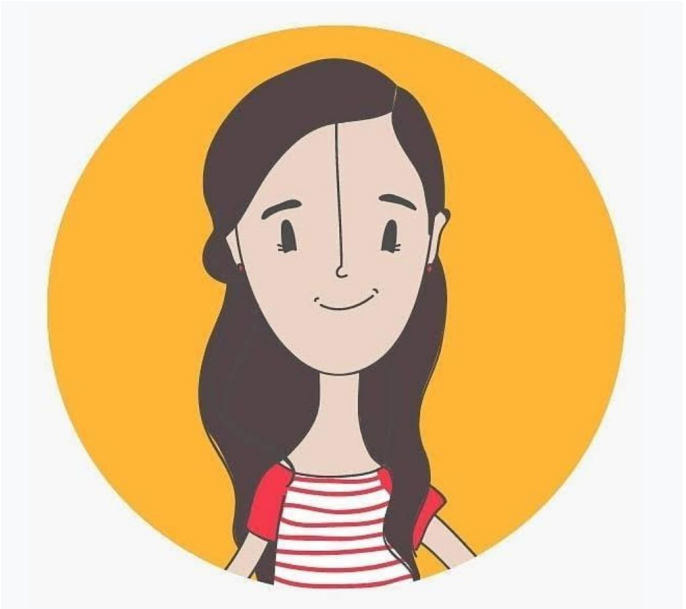
🦄Maris Botero

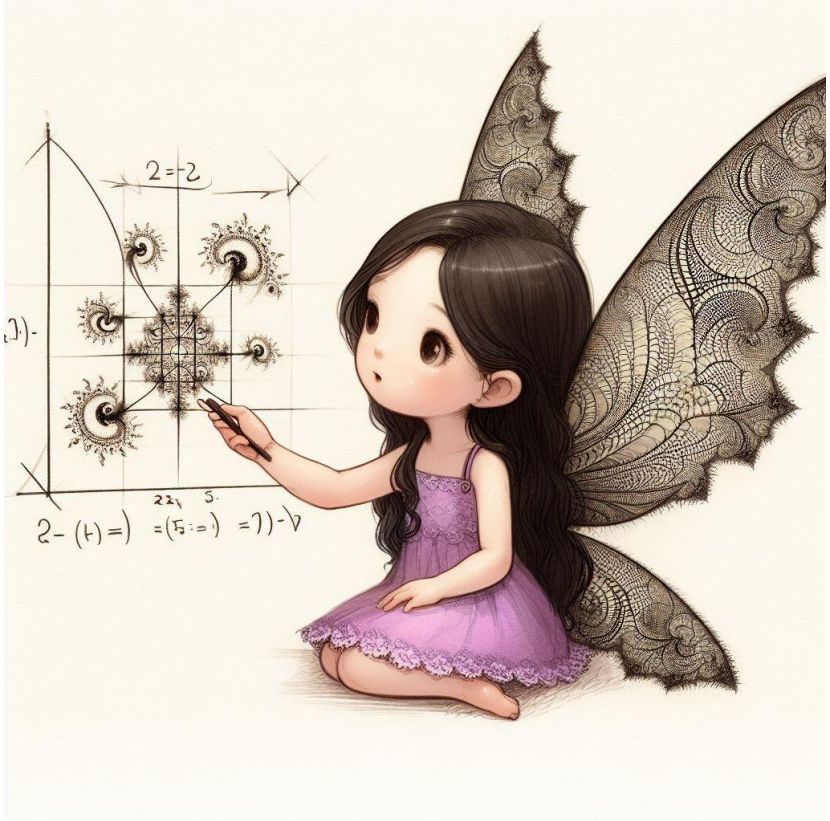
Colombia-Medellín

Científica de datos

Juego con datos, pintó con luz y
a veces cuido plantas🌳

@marisbotero





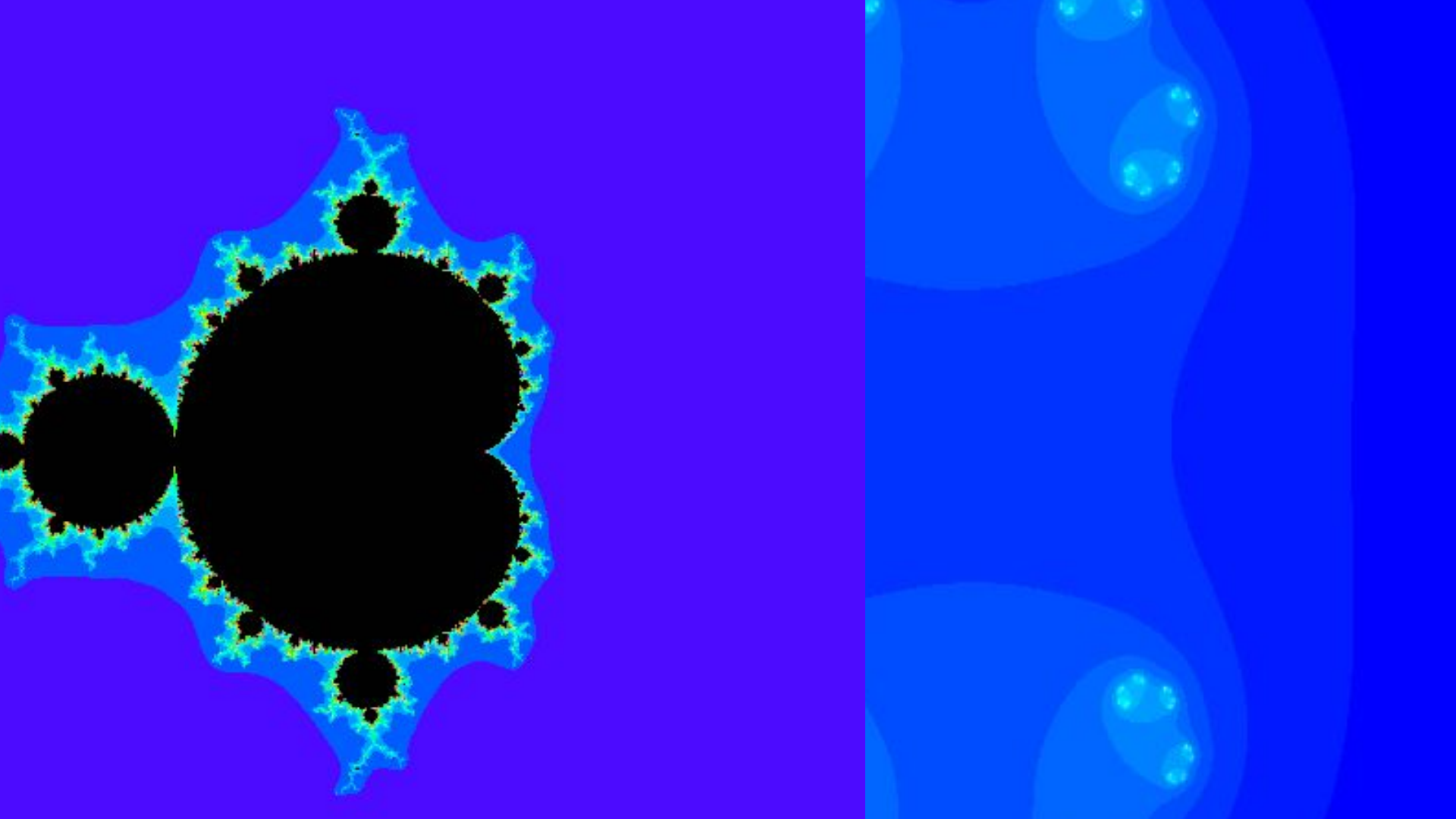
"El científico no estudia la naturaleza por la utilidad que le pueda reportar; la estudia por el gozo que le proporciona, y este gozo se debe a la belleza que hay en ella..."

Henri Poincaré

"Ni las nubes son esféricas, ni las montañas cónicas, ni las costas circulares, ni el tronco de un árbol cilíndrico, ni un rayo viajan en línea recta..."

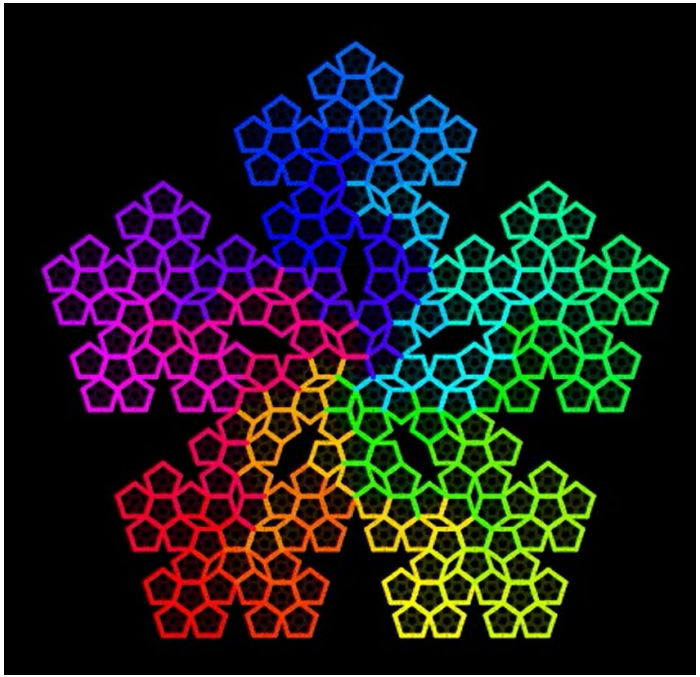
Benoît Mandelbrot

¿Qué es un fractal?



Propiedades de los Fractales

- Estructura fina
- Auto-similitud
- Los métodos clásicos de geometría y matemáticas no son aplicables
- El "Tamaño" depende de la escala a la que se mida
- Una construcción recursiva simple
- Una apariencia natural



Fractales y Python

turtle — Turtle graphics

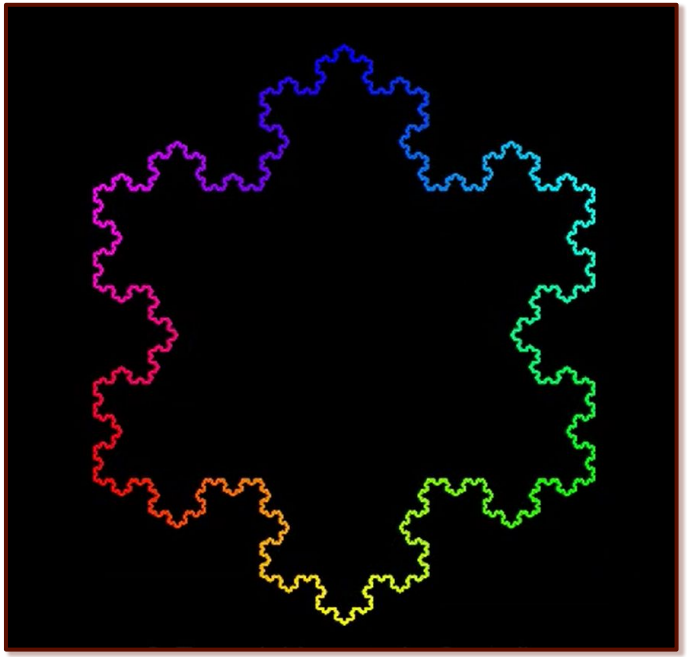
matplotlib

Tk

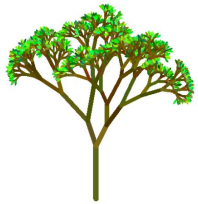
pygame

NumPy

pillow

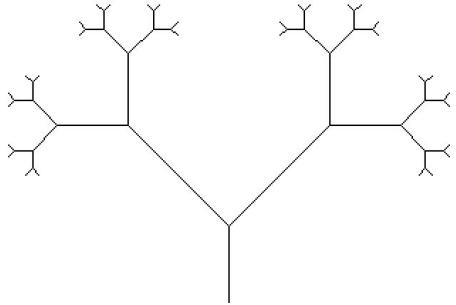


Recursividad



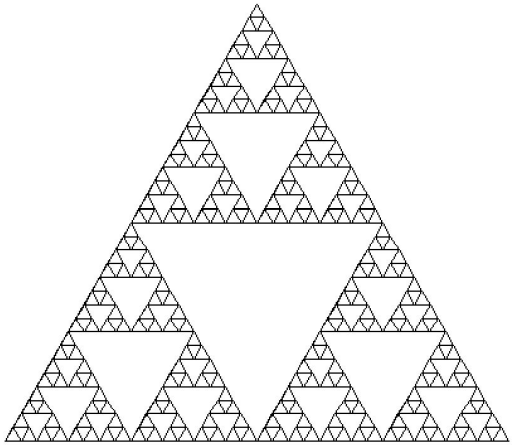
```python
import turtle

def draw_tree(order, size):
    """
    Dibuja un árbol de Pitágoras de un cierto orden y tamaño.
    """
    if order == 0:  # caso base: solo dibuja una línea
        turtle.forward(size)
        turtle.backward(size)
    else:
        # dibuja la "rama" (un triángulo rectángulo)
        turtle.forward(size)

        turtle.right(45)

        # dibuja dos "sub-árboles"
        draw_tree(order-1, size/2)

        turtle.left(90)

        draw_tree(order-1, size/2)

        turtle.right(45)

        # vuelve a la posición y orientación original
        turtle.backward(size)

# establece la velocidad de dibujo
turtle.speed(1)

# mueve la tortuga a una buena posición de inicio
turtle.up()
turtle.goto(-100, -200)
turtle.left(90)  # Hace que la tortuga mire hacia arriba
turtle.down()
```

```python
# dibuja el árbol
draw_tree(5, 200)

# oculta la tortuga
turtle.hideturtle()

# mantiene la ventana abierta hasta que el usuario la cierre
turtle.done()
```

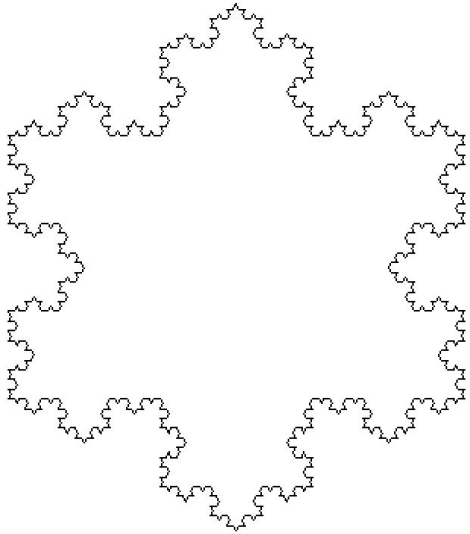


Geometría Fractal- Dibujando el Triángulo de Sierpinski

```python
import turtle

def draw_sierpinski(length, depth):
    if depth == 0:
        for _ in range(3):
            turtle.forward(length)
            turtle.left(120)
    else:
        draw_sierpinski(length / 2, depth - 1)
        turtle.forward(length / 2)
        draw_sierpinski(length / 2, depth - 1)
        turtle.backward(length / 2)
        turtle.left(60)
        turtle.forward(length / 2)
        turtle.right(60)
        draw_sierpinski(length / 2, depth - 1)
        turtle.left(60)
        turtle.backward(length / 2)
        turtle.right(60)

# Inicializar la tortuga
turtle.speed(0)

# Posición inicial de la tortuga
turtle.up()
turtle.goto(-200, -175)
turtle.down()

# Dibujar el triángulo de Sierpinski
draw_sierpinski(400, 5)

# Ocultar la tortuga
turtle.hideturtle()
```

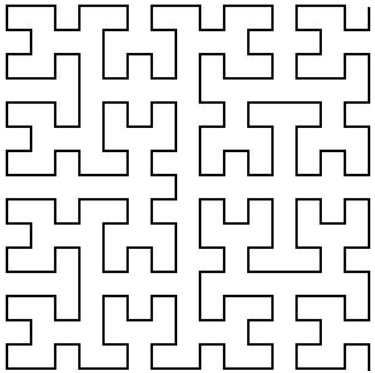


El copo de nieve de Koch



```python
import turtle

def draw_koch_snowflake(length, depth):
    if depth == 0:
        turtle.forward(length)
    else:
        draw_koch_snowflake(length/3, depth-1)
        turtle.left(60)
        draw_koch_snowflake(length/3, depth-1)
        turtle.right(120)
        draw_koch_snowflake(length/3, depth-1)
        turtle.left(60)
        draw_koch_snowflake(length/3, depth-1)

def draw_snowflake(length, depth):
    for _ in range(3):
        draw_koch_snowflake(length, depth)
        turtle.right(120)

turtle.speed(0)

turtle.up()
turtle.goto(-150, 90)
turtle.down()

draw_snowflake(300, 4)

turtle.hideturtle()
turtle.done()
```

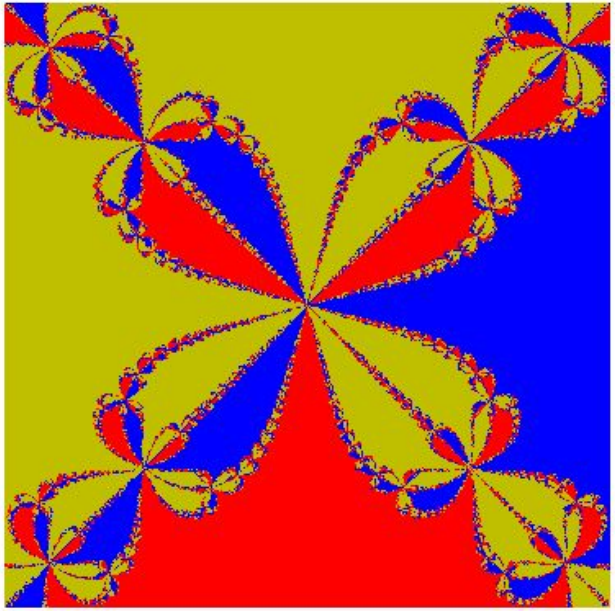
La curva de Hilber

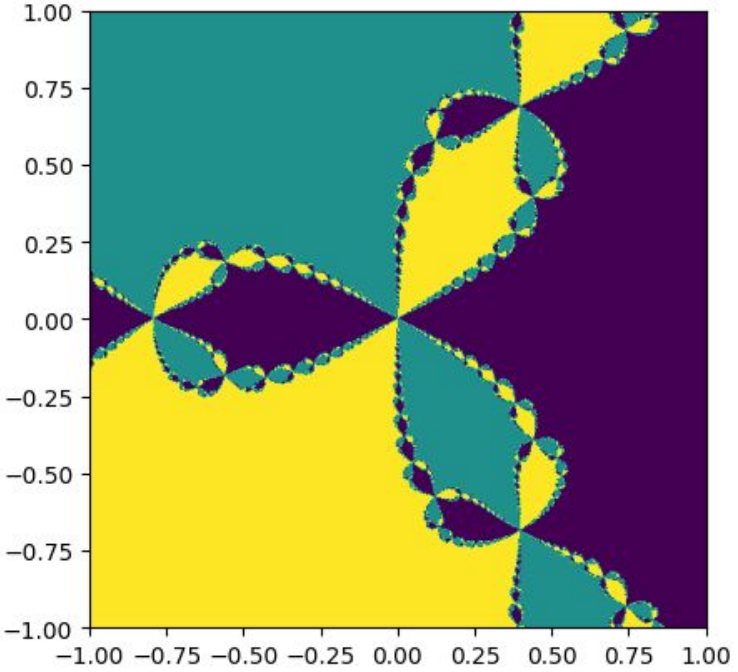


```python
import turtle

def hilbert_curve(order, angle, distance):
    # Caso base: order es 0
    if order == 0:
        return

    turtle.right(angle)
    hilbert_curve(order-1, -angle, distance)
    turtle.forward(distance)
    turtle.left(angle)
    hilbert_curve(order-1, angle, distance)
    turtle.forward(distance)
    hilbert_curve(order-1, angle, distance)
    turtle.left(angle)
    turtle.forward(distance)
    hilbert_curve(order-1, -angle, distance)
    turtle.right(angle)

# inicializar la tortuga
turtle.speed(0)

# cambiar el tamaño de la ventana de la tortuga
turtle.setup(800, 800)
```

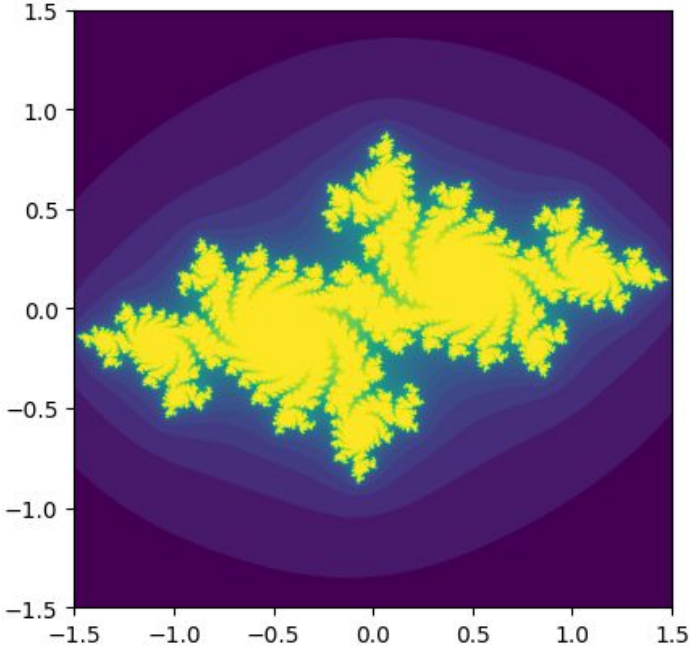
```python
# cambiar el tamaño de la ventana de la tortuga
turtle.setup(800, 800)

# posicionar la tortuga
turtle.penup()
turtle.goto(-200, -200)
turtle.pendown()

# dibujar la curva de Hilbert
hilbert_curve(5, 90, 10)

# ocultar la tortuga
turtle.hideturtle()

# mantener la ventana abierta
turtle.done()
```

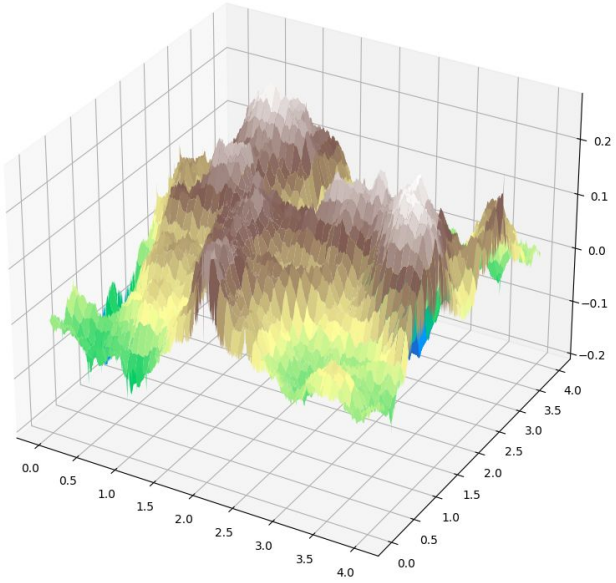
Fractal de Newton





Conjunto Julia - paisaje fractal





Fractales y la inteligencia artificial

GRACIAS