



# Criando mundos utilizando caos

**Apresentado por:**

Anderson Pereira

# Quem sou eu



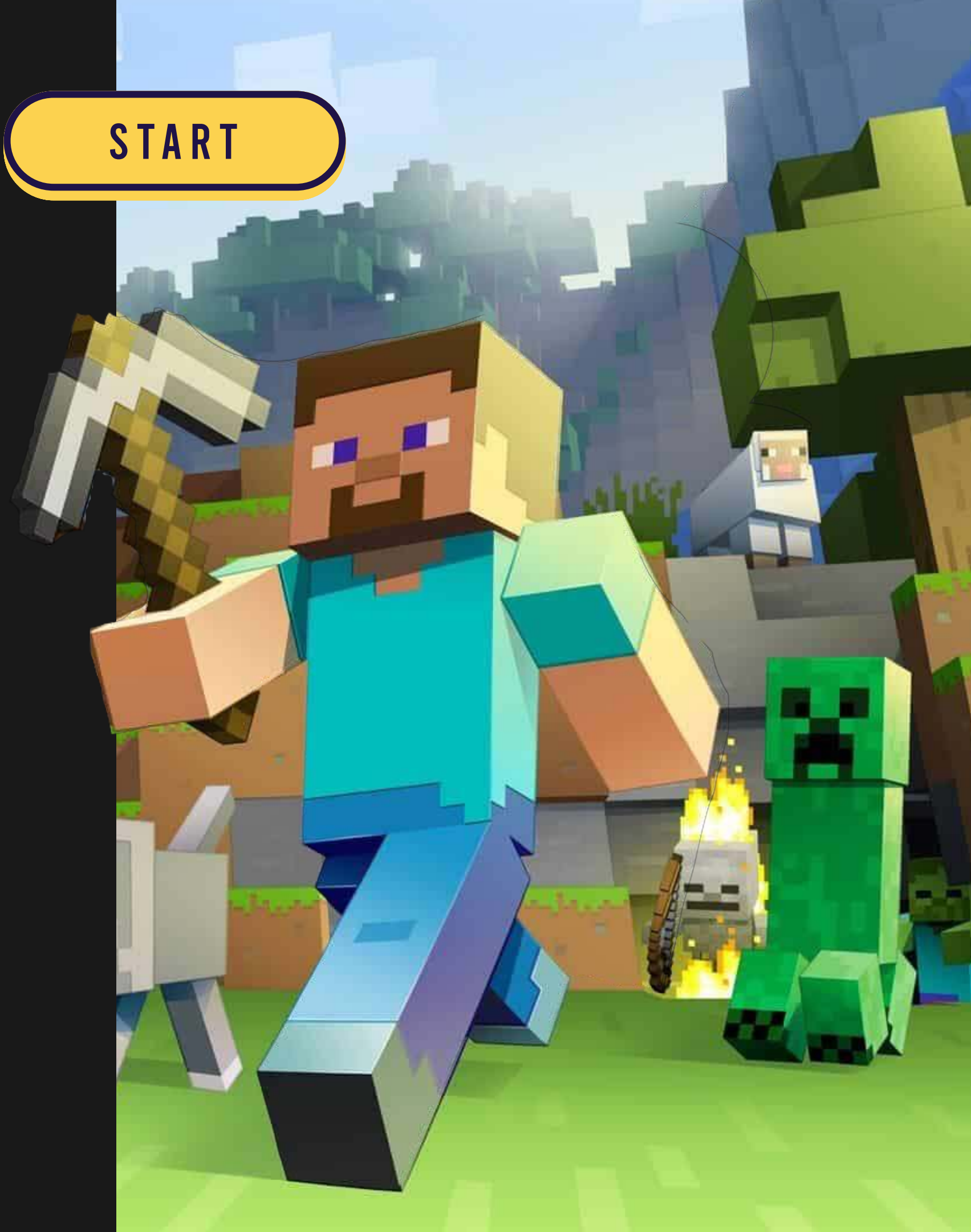
- 1 Graduando em Tecnologia da Informação pela UFRN;
- 2 Formado em Informática para internet pelo IFRN;
- 3 Técnico em Criação de jogos pela Metrópole Digital;
- 4 Aspirante a gamer.



# Por que falar de jogos?

- Entretenimento
- Cultura
- Inovação Tecnológica
- Comunidade
- Educação
- Arte
- Gosto pessoal

START





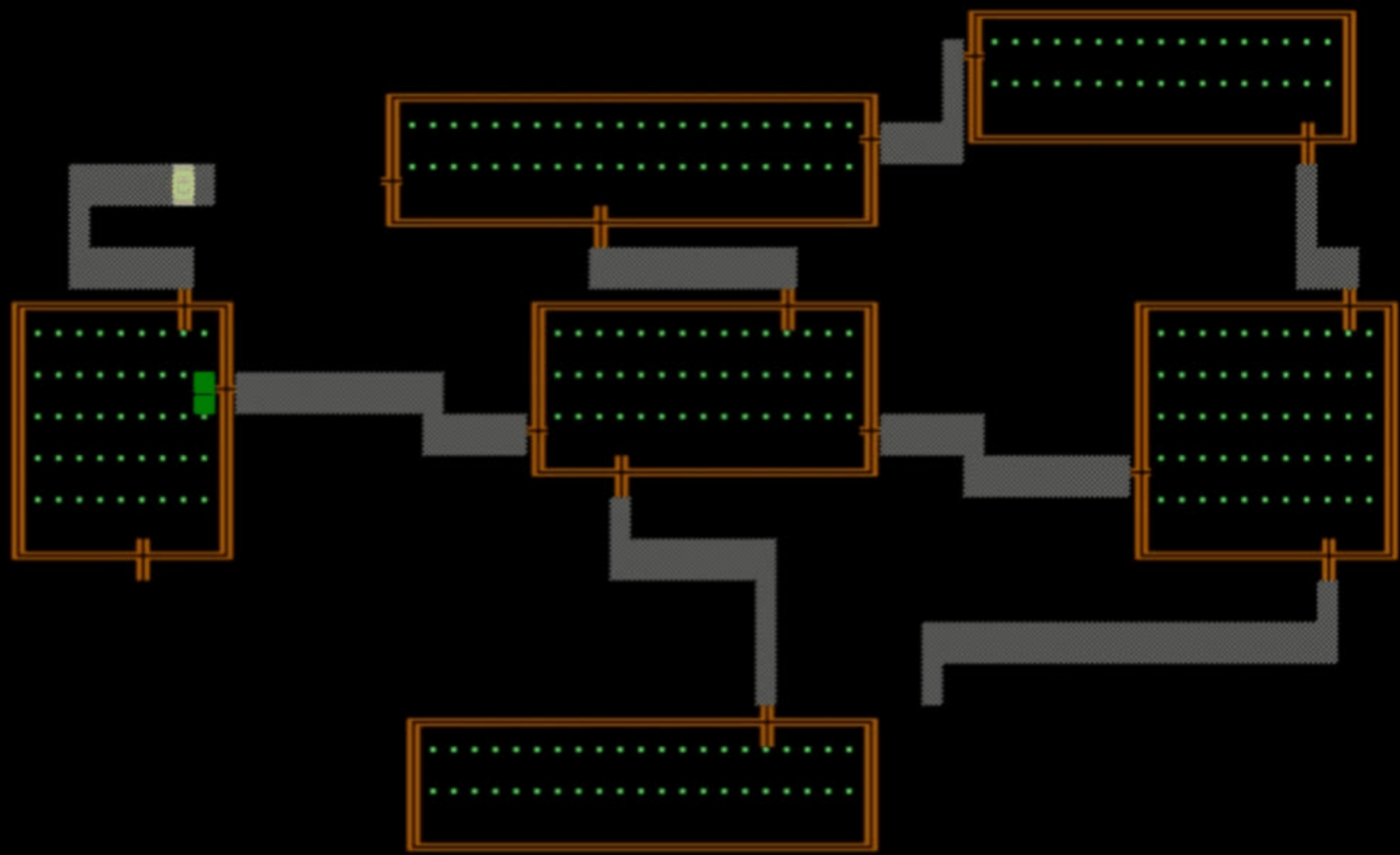
# HADES



# Roguelikes

- Um subgênero de jogos RPG
- Ladrilho
- Morte permanente
- Geração de nível aleatória ou procedural durante a partida

# Welcome to the Dungeons of Doom



Level:1 Hits:12(12) Str:16(16) Gold:75 Armor:5 Exp:1/5



# Geração Procedural

## GERAÇÃO

substantivo feminino

1. ação ou efeito de gerar(-se).

## PROCEDURAL

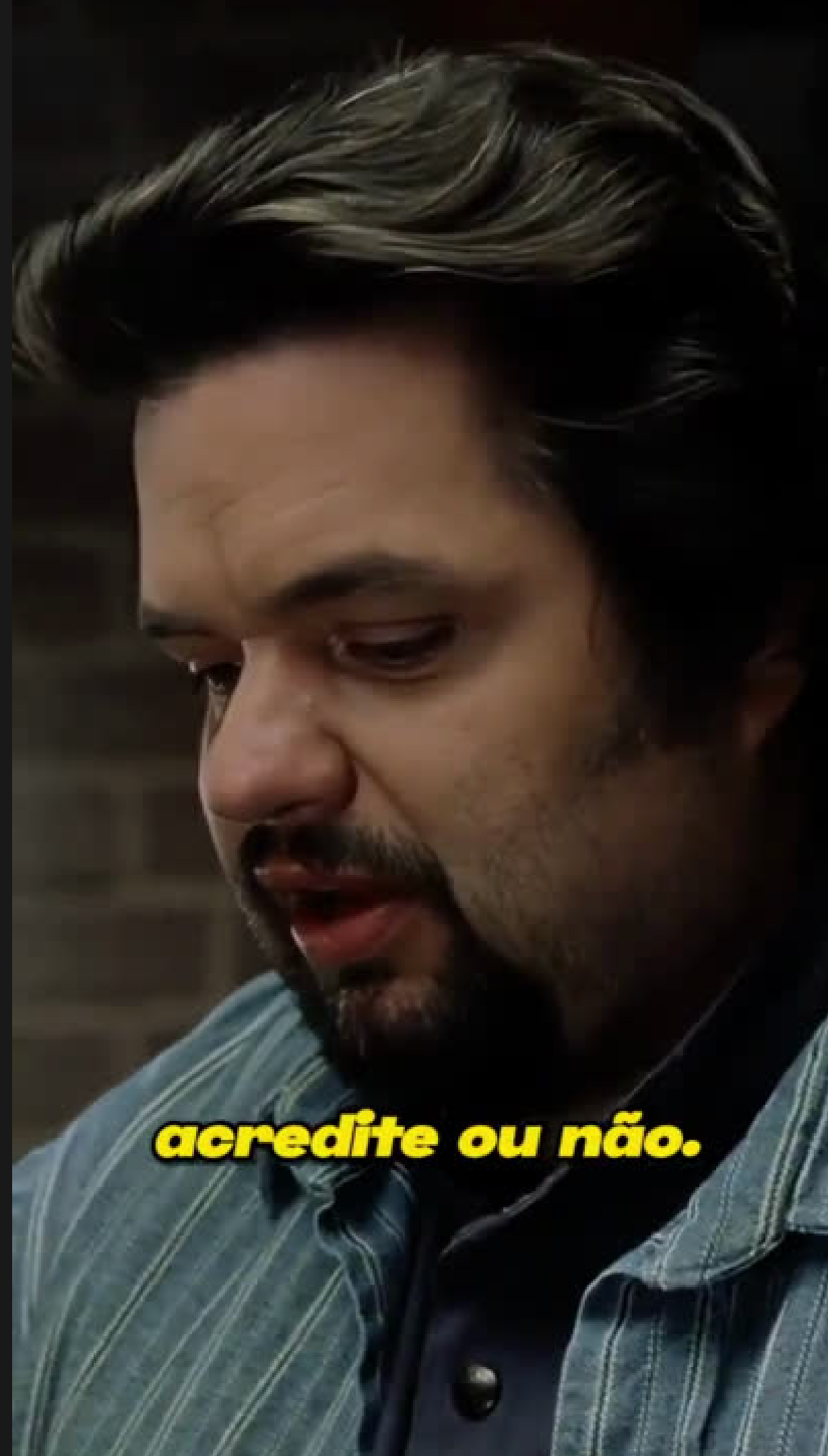
substantivo masculino

1. sequência contínua de fatos ou operações

2. modo de fazer alguma coisa; método, maneira, procedimento.



RUÍDOS



**acredite ou não.**

O Homem Bicentenário  
(1999)



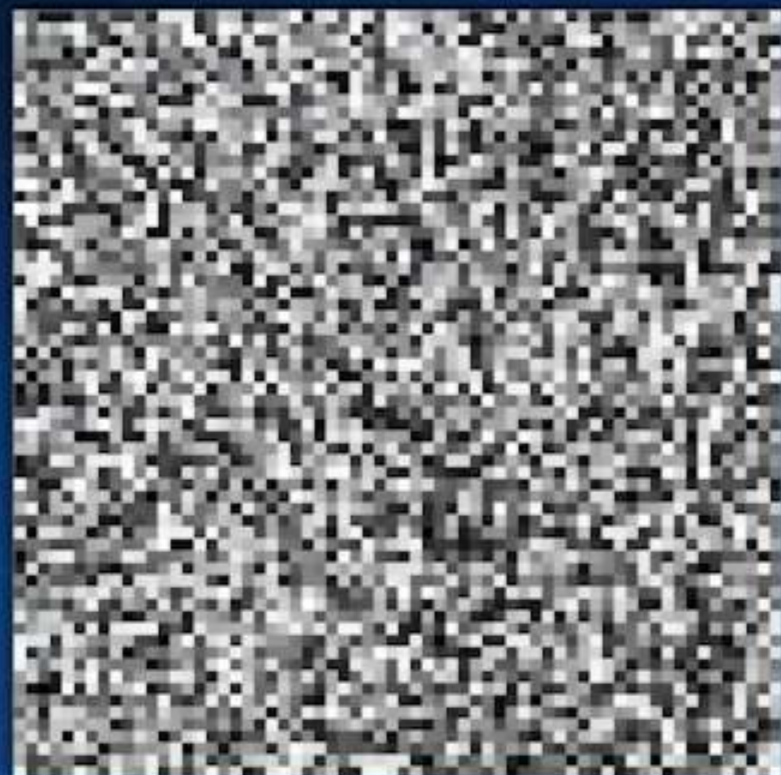
# Ruído

"Noise" (ruído) em matemática e computação se refere a uma espécie de "bagunça" ou "imprevisibilidade" que é adicionada a dados de propósito.

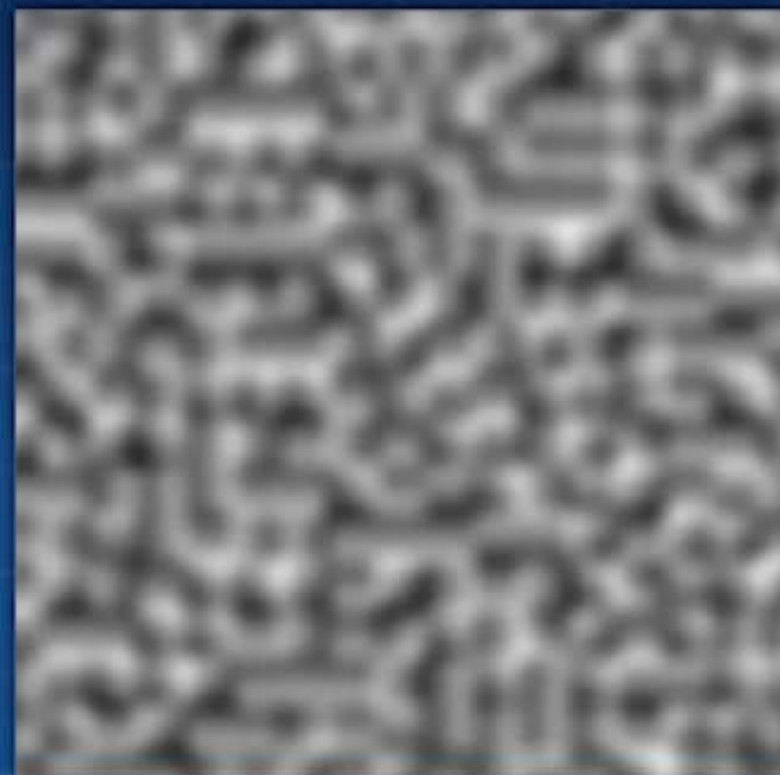
Isso é feito para tornar esses dados mais naturais, complexos ou visualmente interessantes.



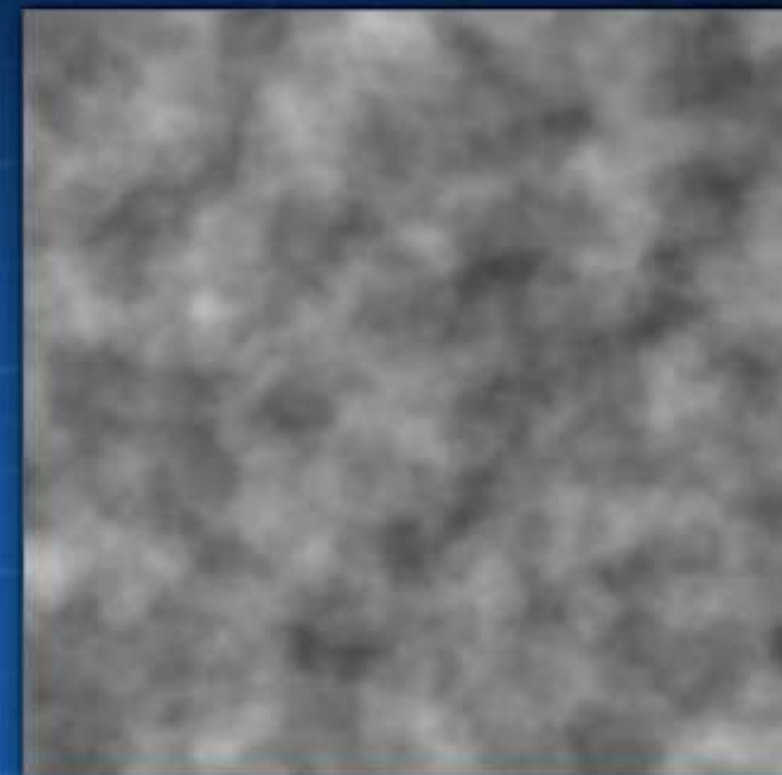




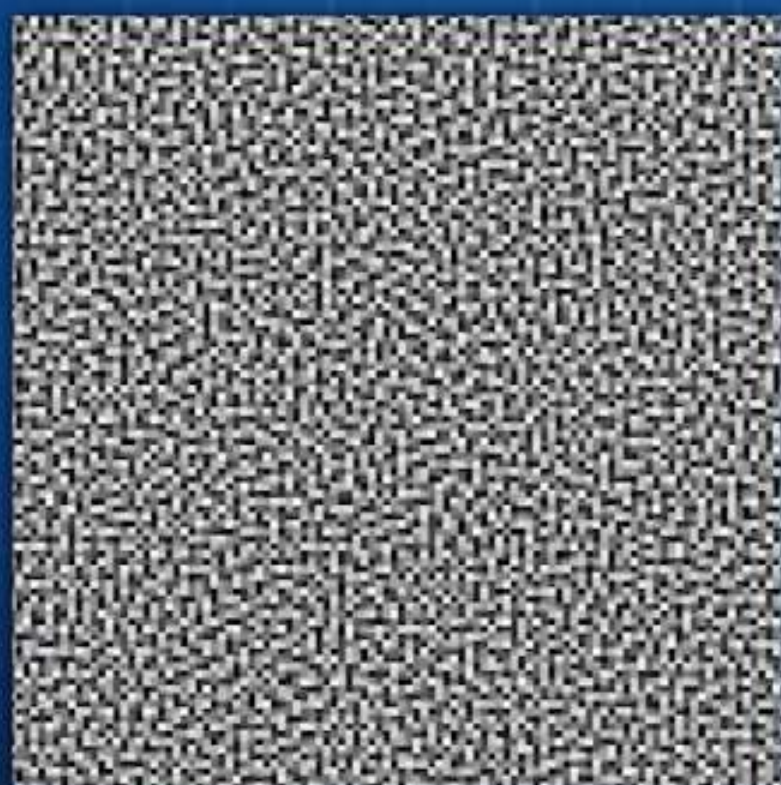
**White Noise**



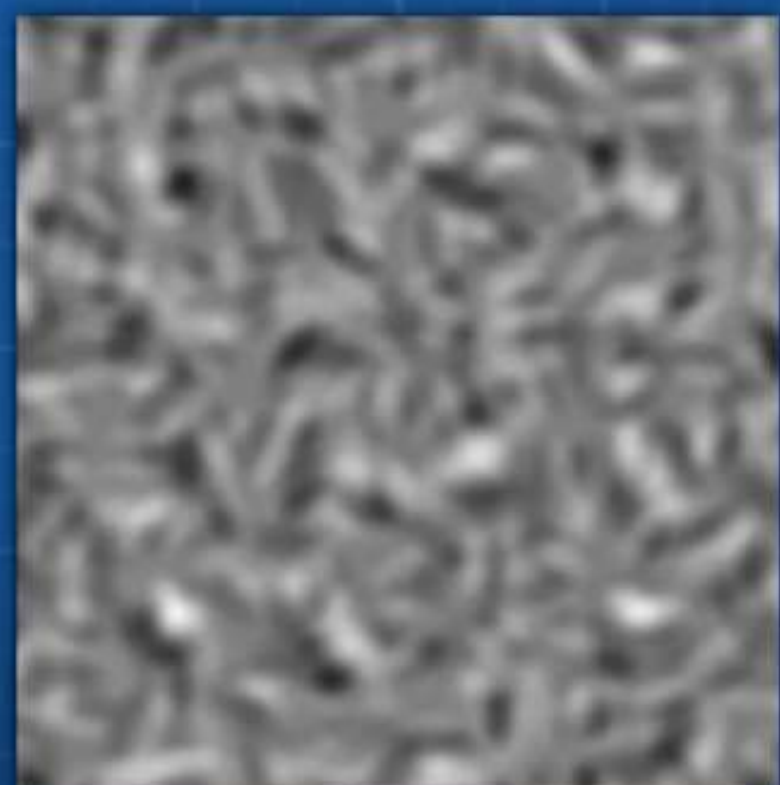
**Perlin Noise**



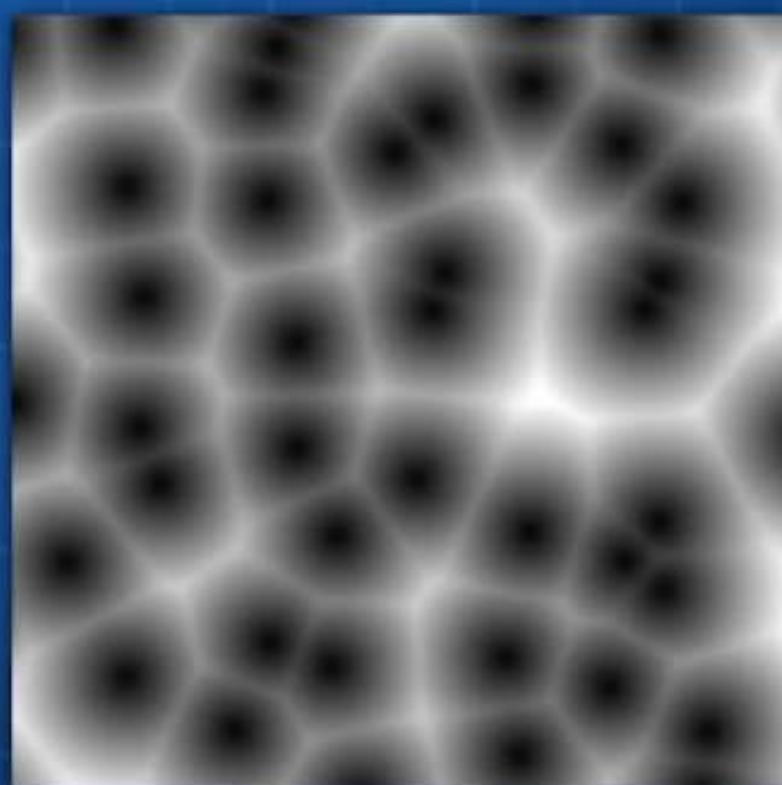
**Fractal Noise**



**Blue Noise**

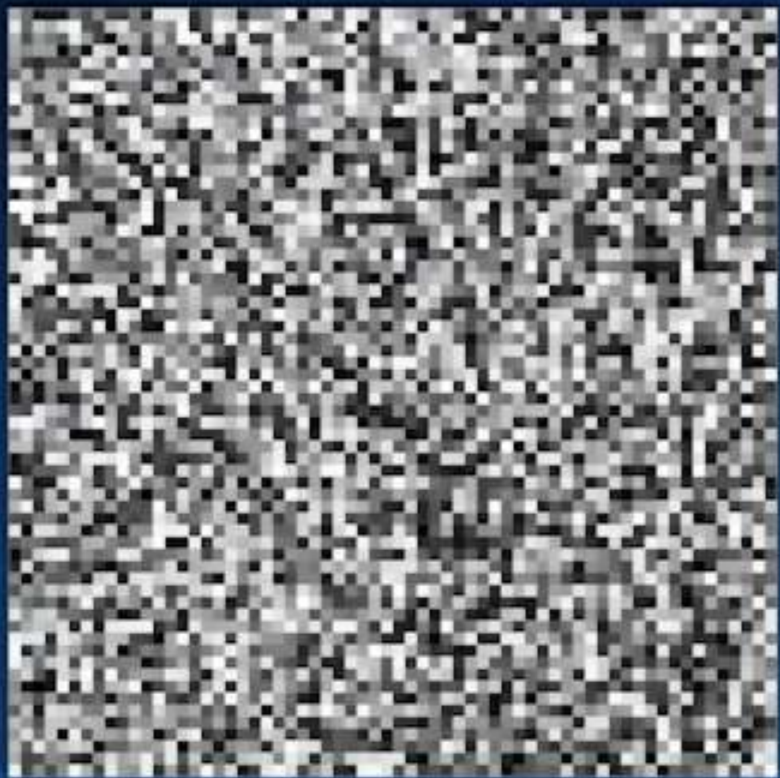


**Wavelet Noise**

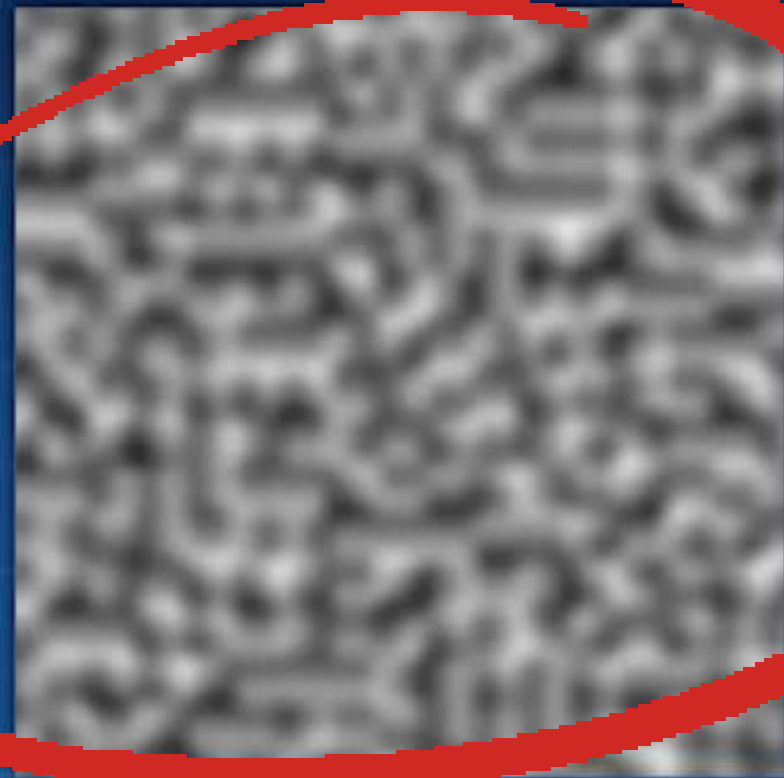


**Cellular Noise**

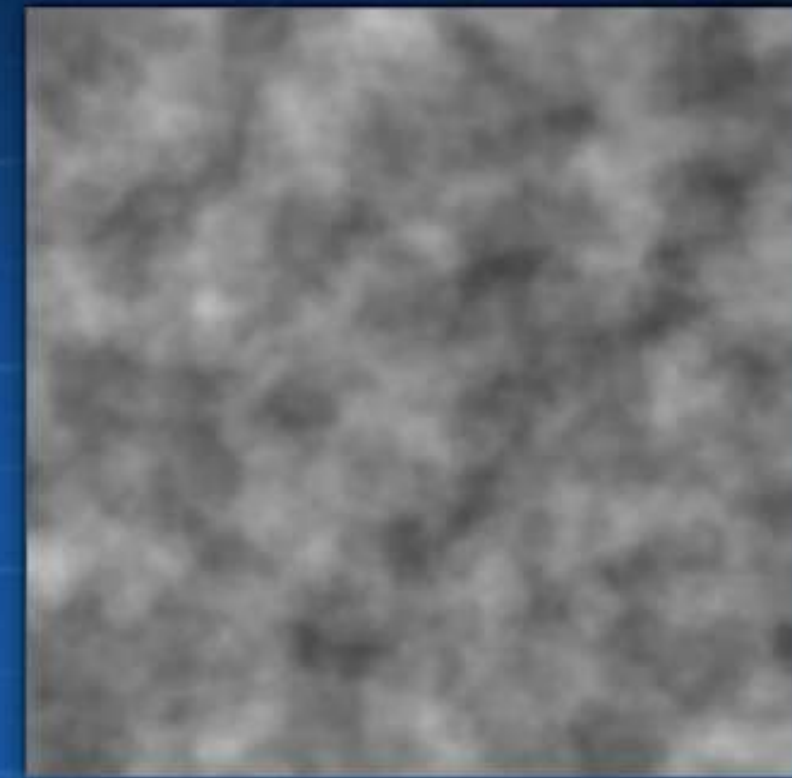




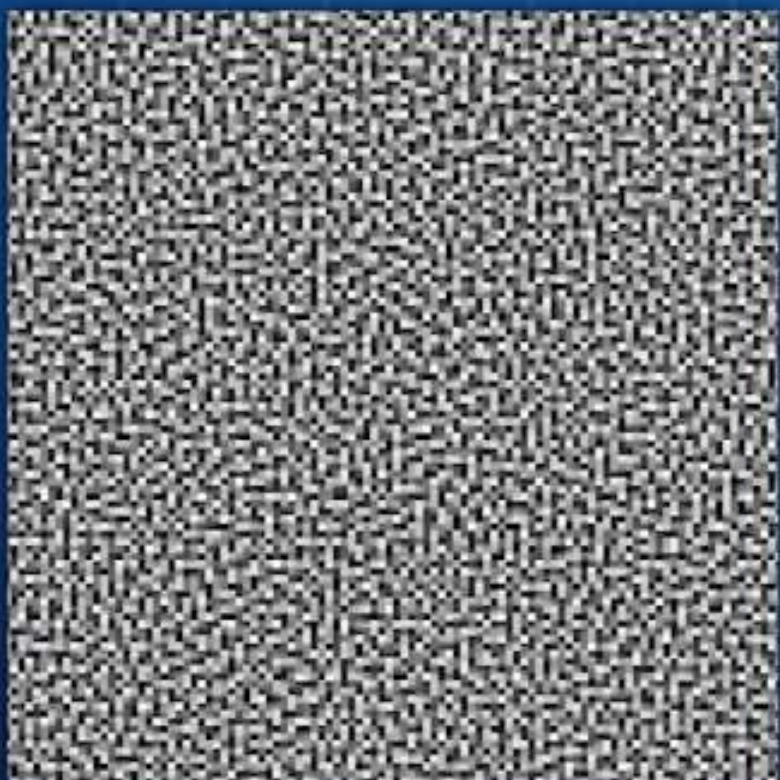
**White Noise**



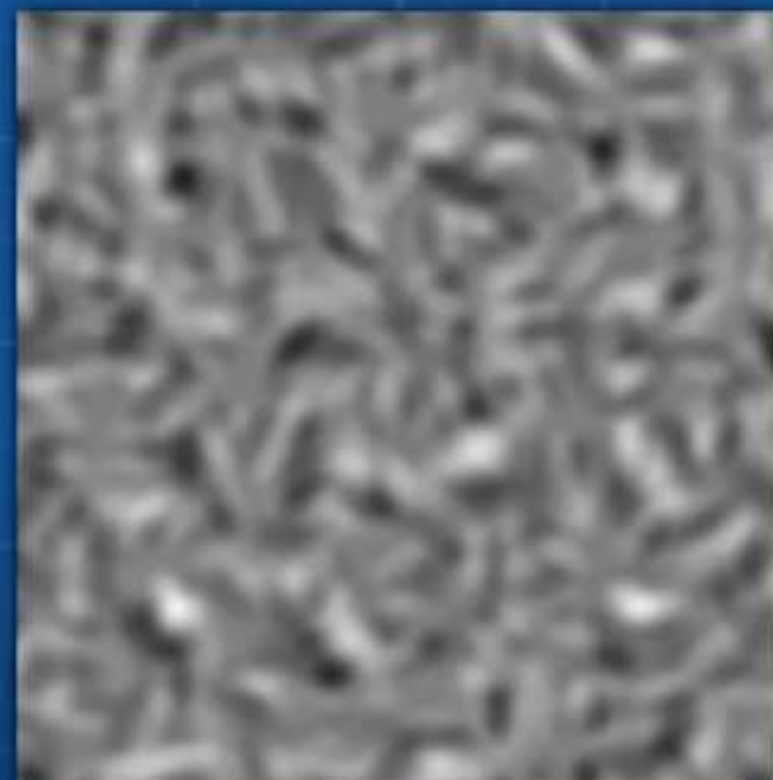
**Perlin Noise**



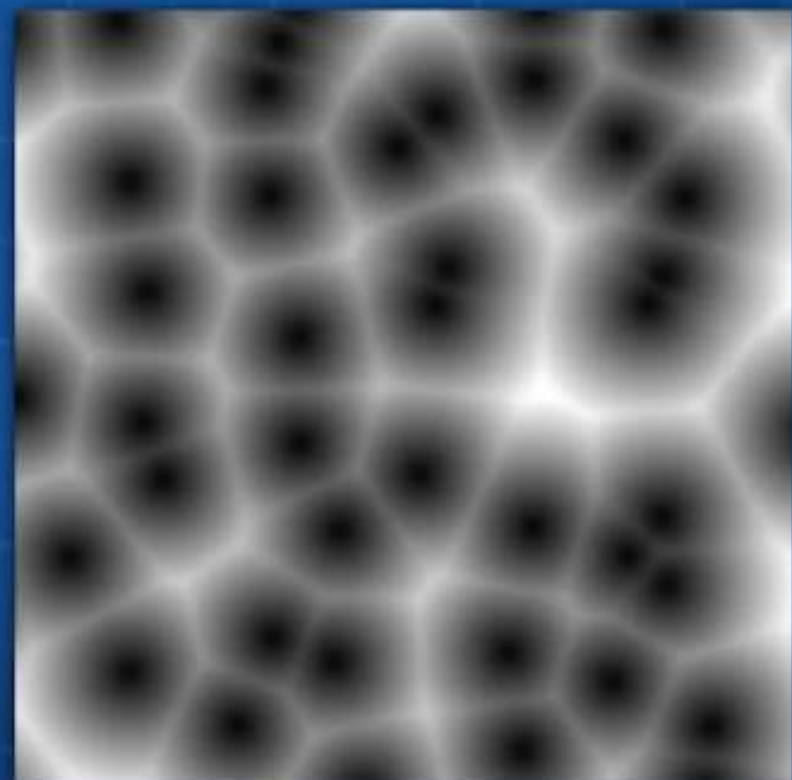
**Fractal Noise**



**Blue Noise**



**Wavelet Noise**



**Cellular Noise**

# Bibliotecas, linguagen e ferramentas utilizadas

## Python

- Noise
  - *pip install noise*
- Pygame
  - *pip install pygame*
- Numpy
  - *pip install numpy*
- Image
  - *pip install image*

## Sublime





```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.8_3.8.2800....
>>> import noise
>>> help(noise)
Help on package noise:

NAME
    noise - Noise functions for procedural generation of content

DESCRIPTION
    Contains native code implementations of Perlin improved noise (with
    fBm capabilities) and Perlin simplex noise. Also contains a fast
    "fake noise" implementation in GLSL for execution in shaders.

    Copyright (c) 2008, Casey Duncan (casey dot duncan at gmail dot com)

PACKAGE CONTENTS
    _perlin
    _simplex
    perlin
    shader
    shader_noise
    test

FUNCTIONS
    pnoise1 = noise1(...)
        noise1(x, octaves=1, persistence=0.5, lacunarity=2.0, repeat=1024
, base=0.0)

        1 dimensional perlin improved noise function (see noise3 for more
info)

    pnoise2 = noise2(...)
        noise2(x, y, octaves=1, persistence=0.5, lacunarity=2.0, repeatx=
1024, repeaty=1024, base=0.0)

        2 dimensional perlin improved noise function (see noise3 for more
info)

    pnoise3 = noise3(...)
        noise3(x, y, z, octaves=1, persistence=0.5, lacunarity=2.0, repea
tx=1024, repeaty=1024, repeatz=1024, base=0.0)
```

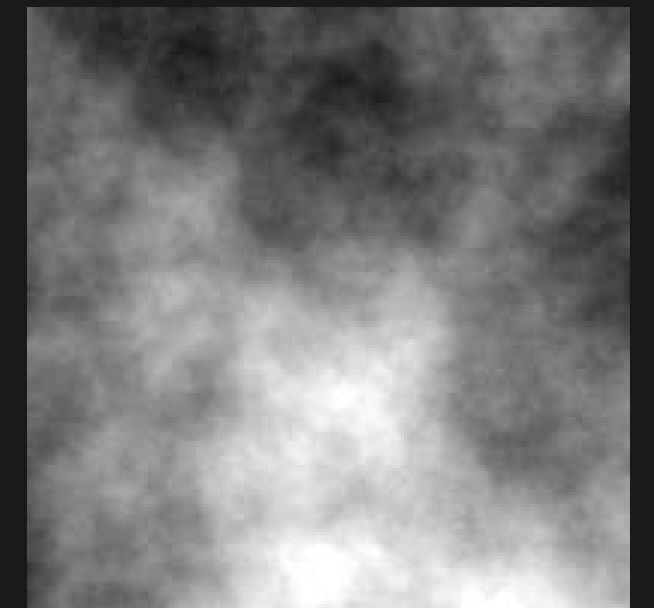
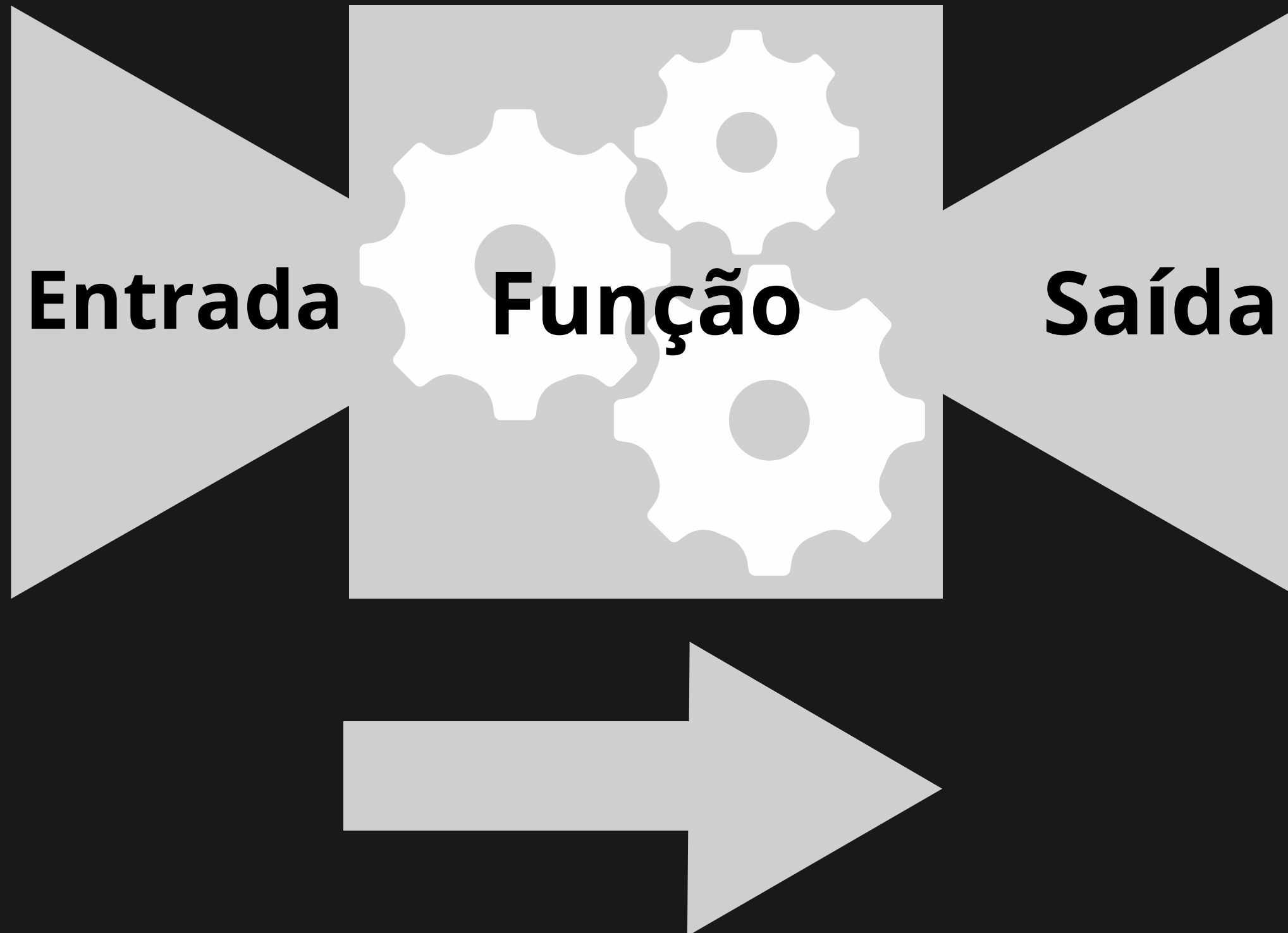
# Noise

*help(noise)*

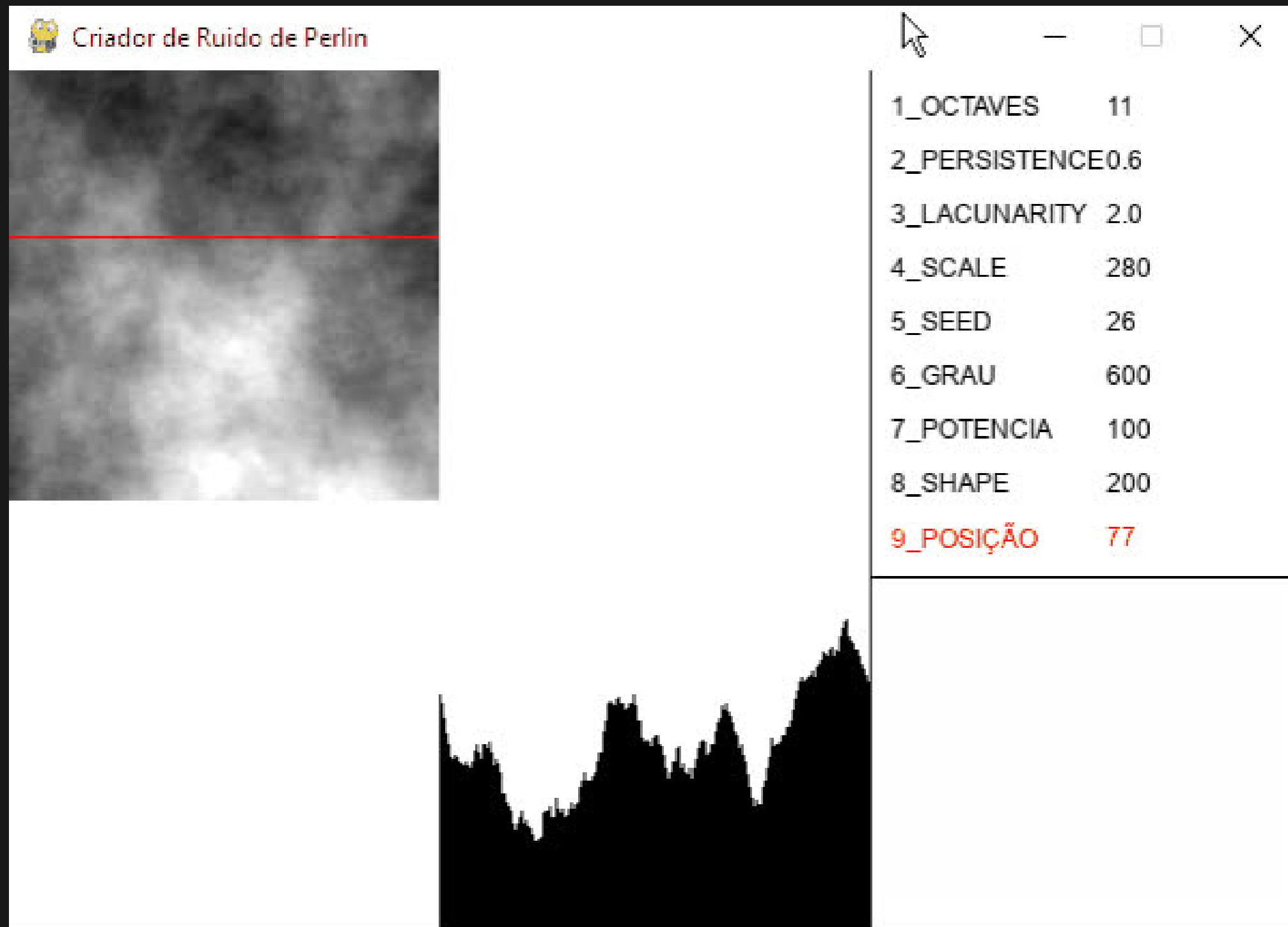
**Descreve as informações da biblioteca Noise utilizada, desde suas funções ate o parametro para as mesmas**

# Noise

octaves  
persistence  
lacunarity  
scale  
seed  
grau  
potencia  
shape



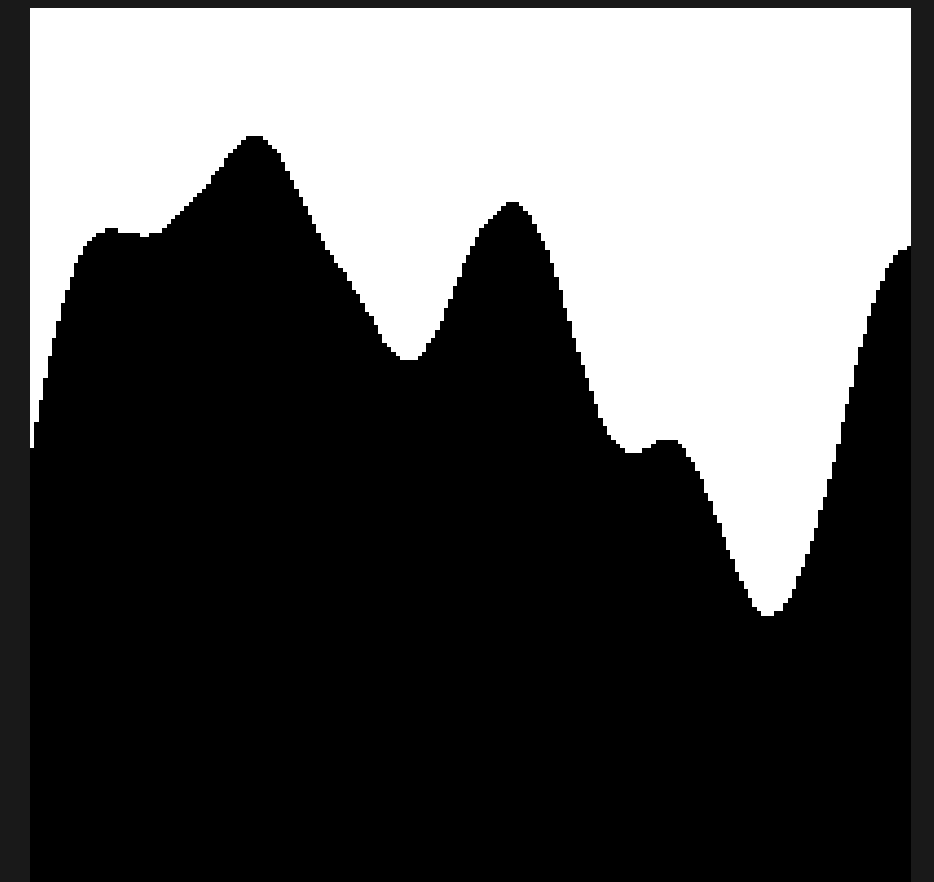
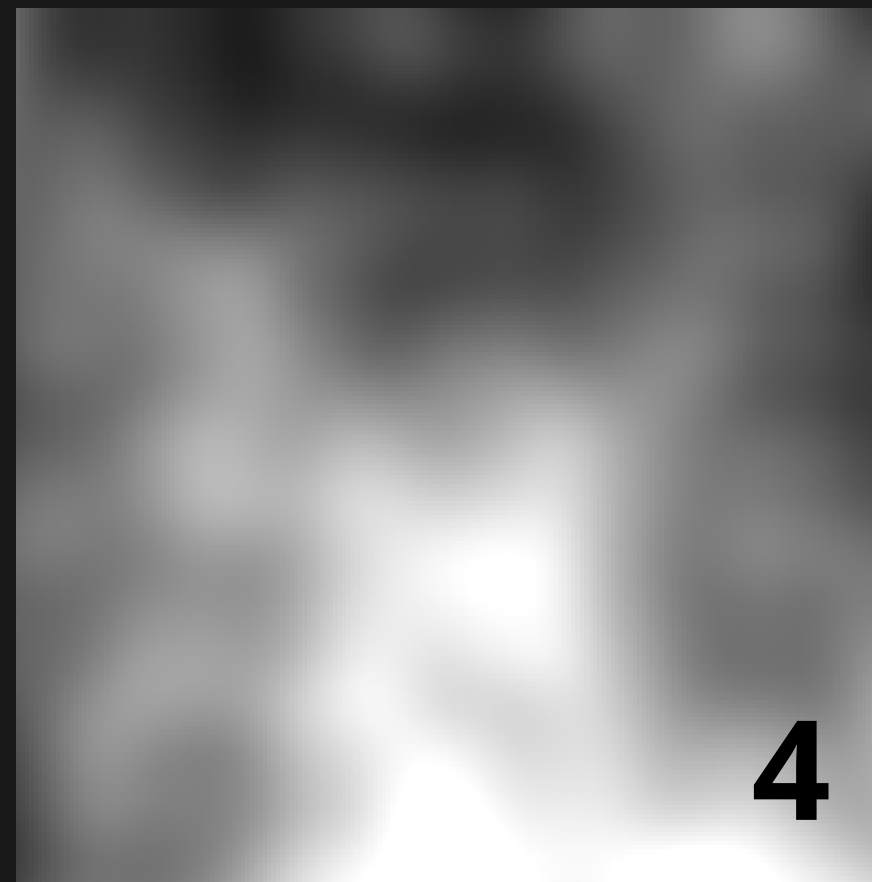
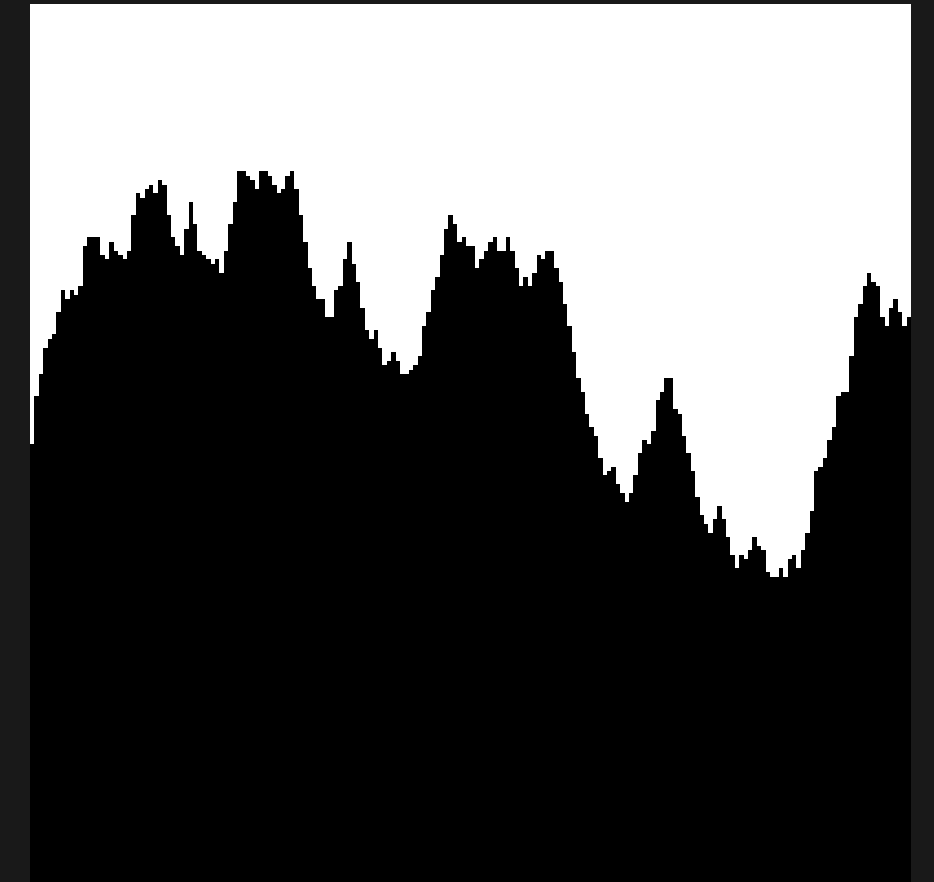
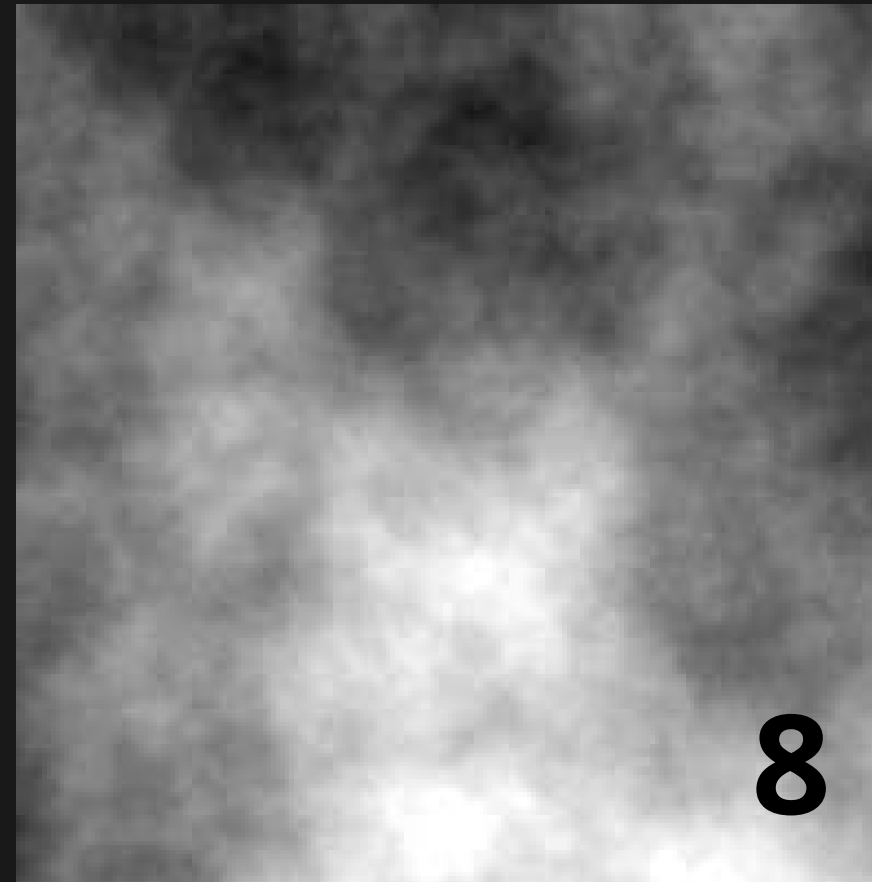
# Criação do ruído 2D



**Vamos observar as variações que ocorrem de acordo com as entradas**

# Octaves

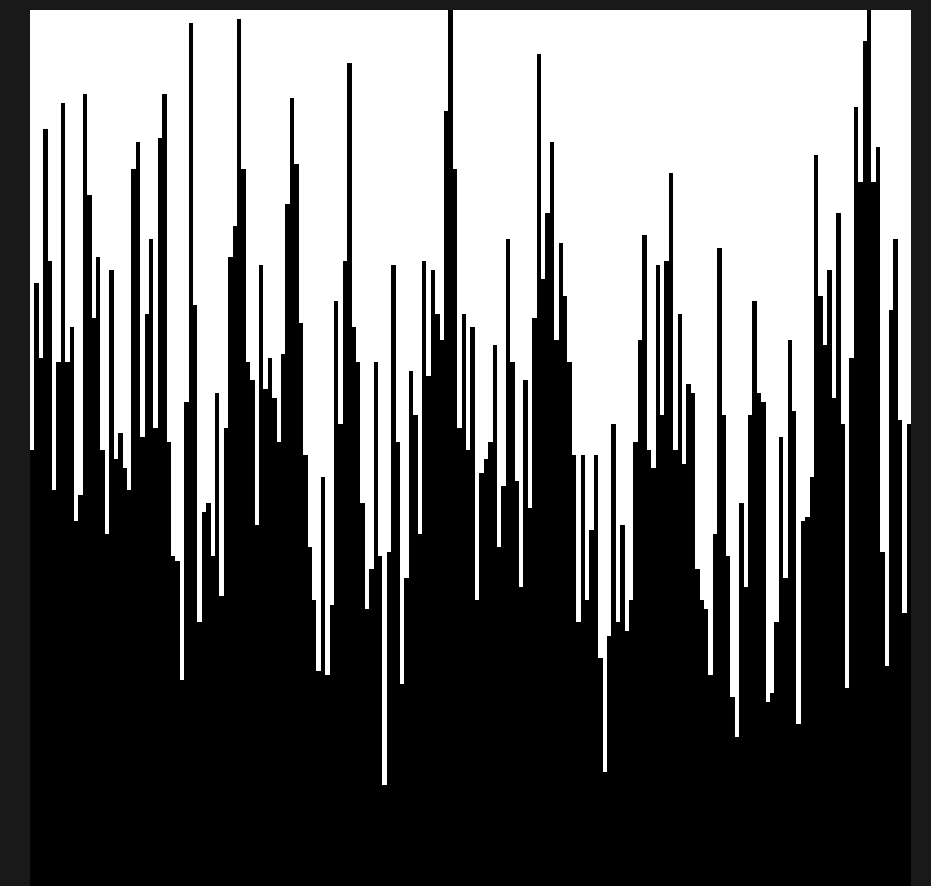
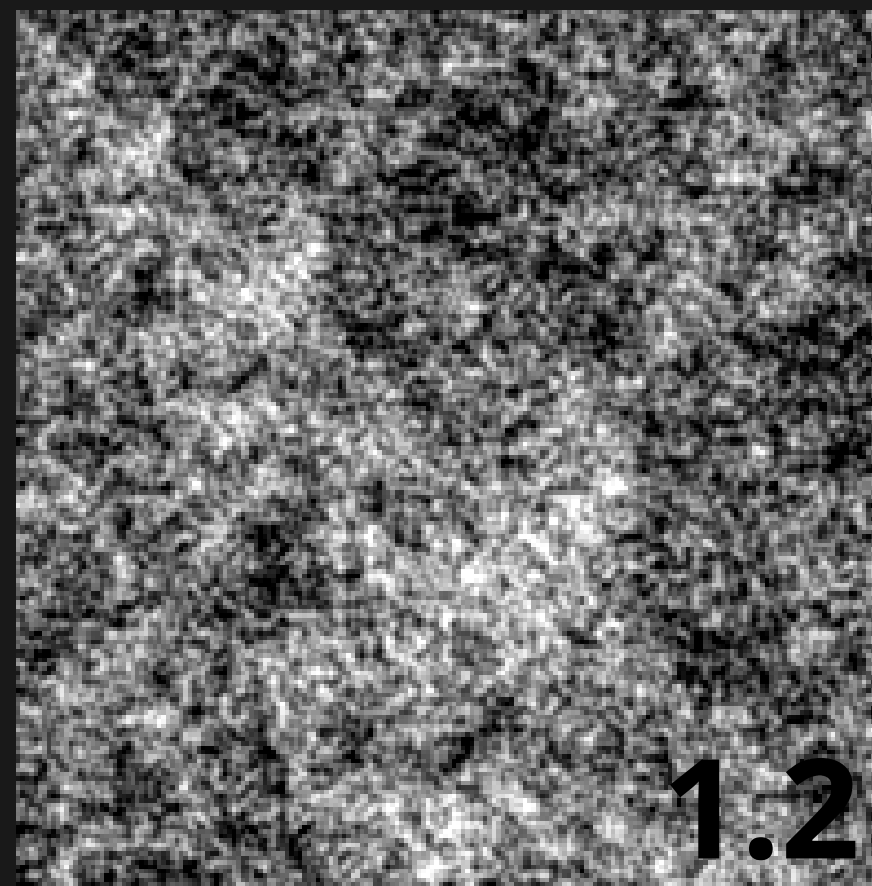
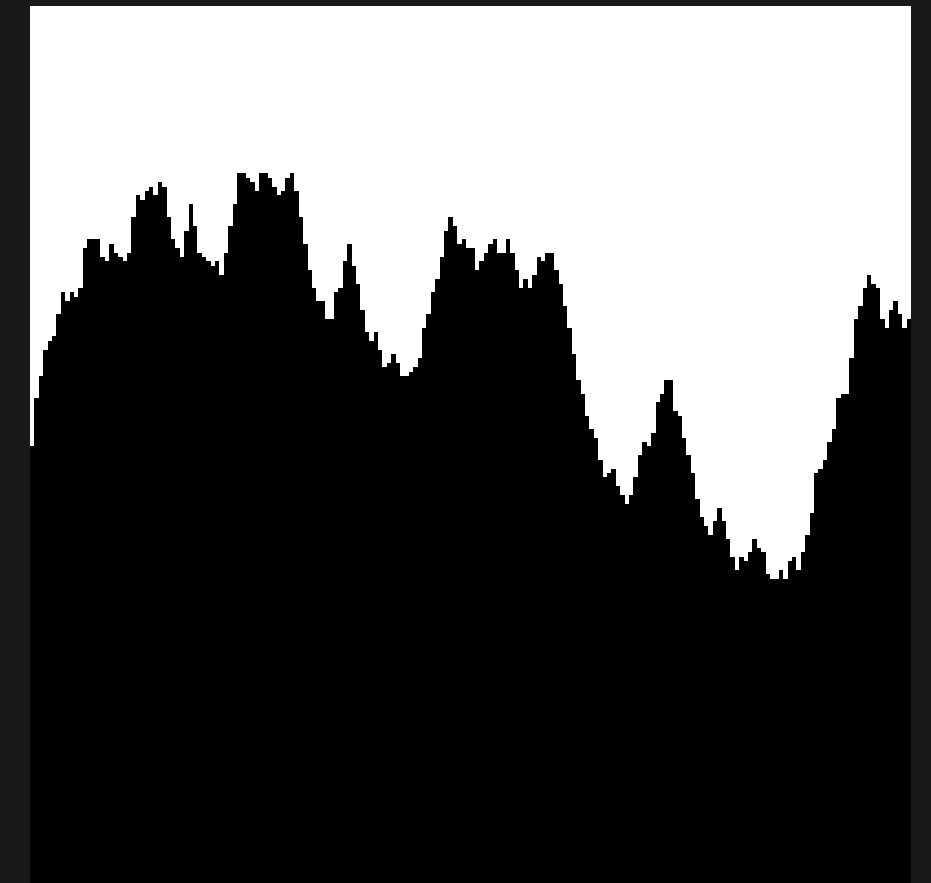
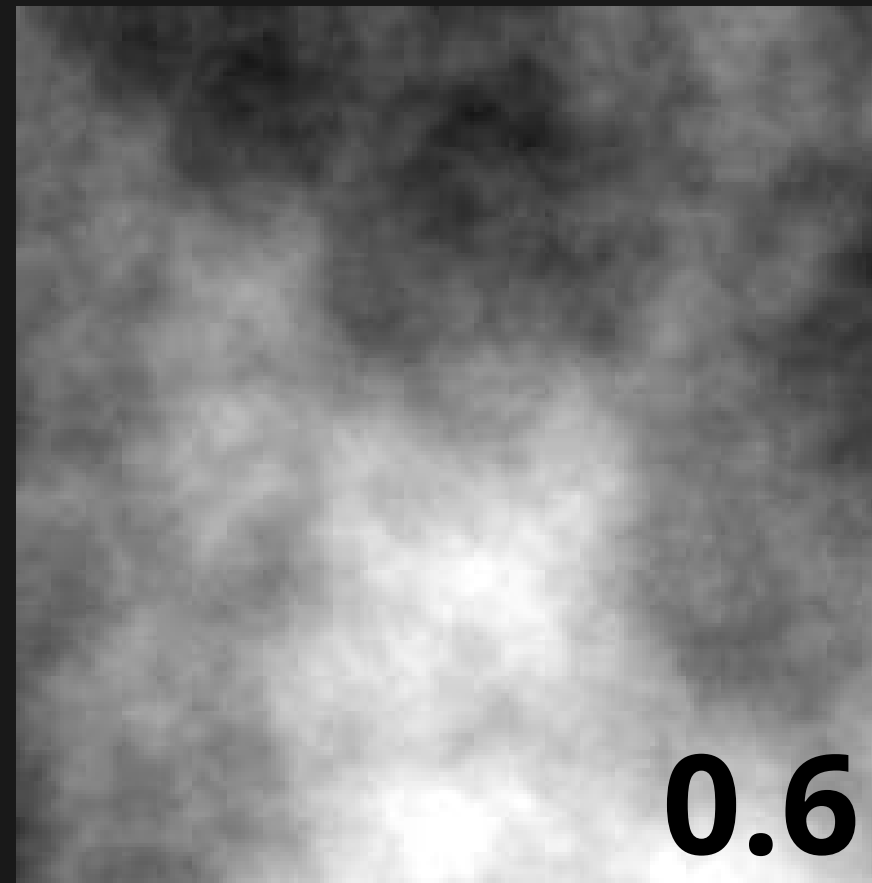
Significa o número de passos/camadas do algoritmo. Cada passe adiciona mais detalhes.





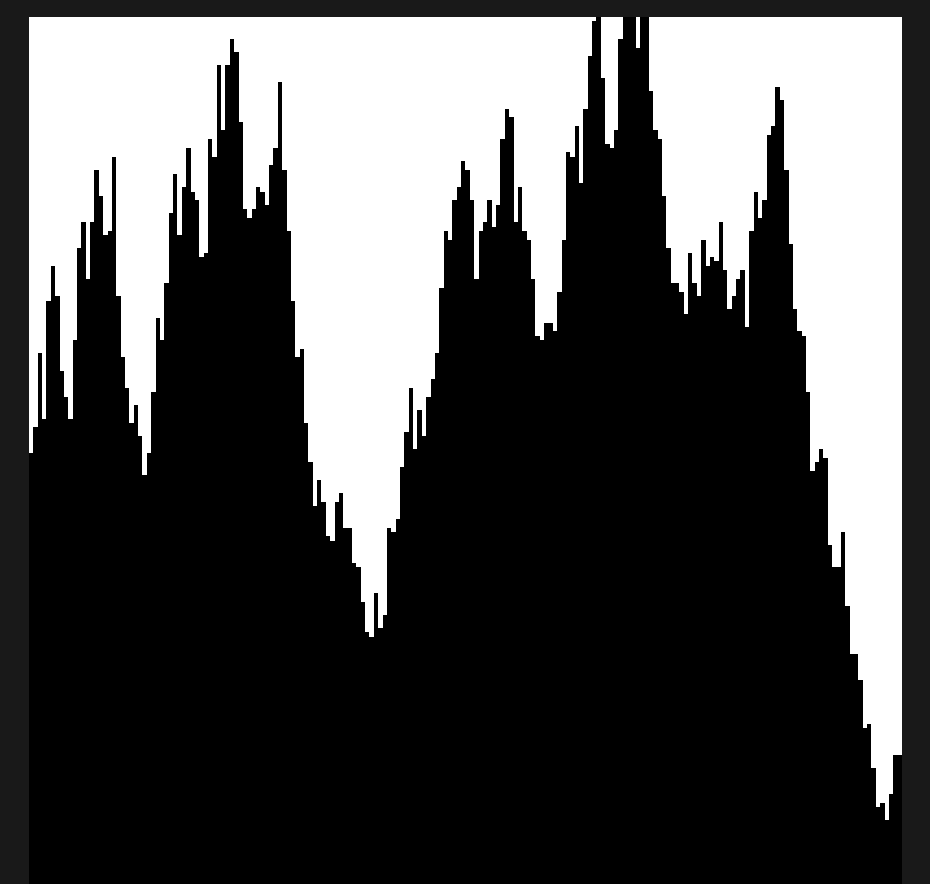
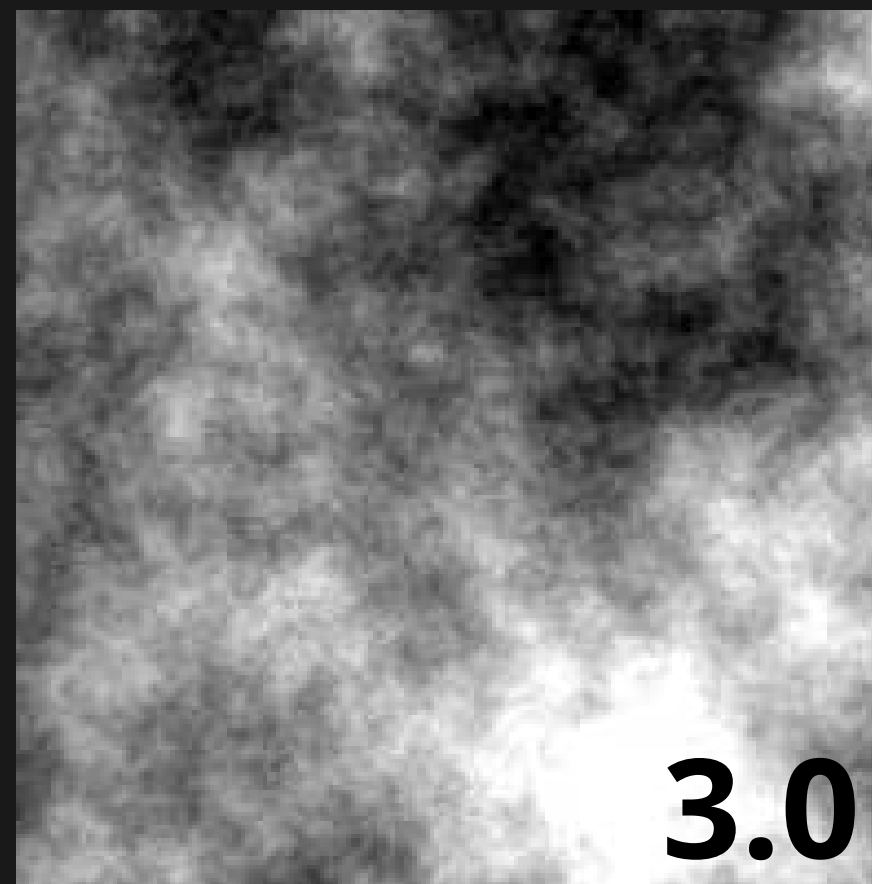
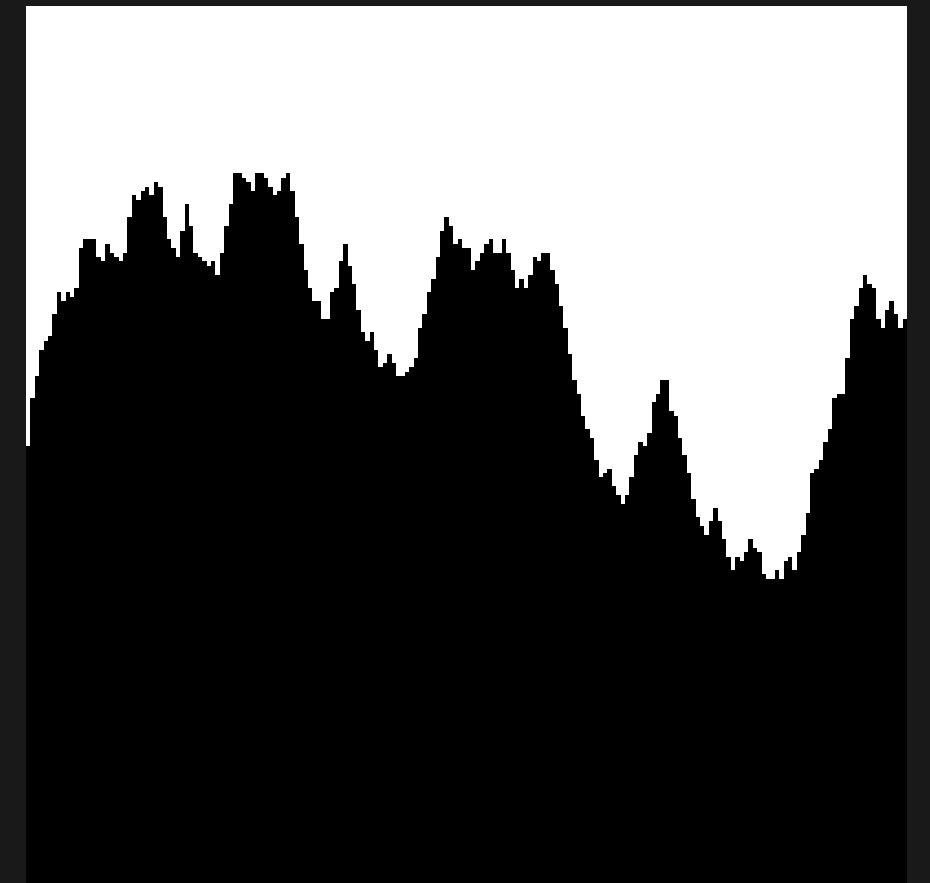
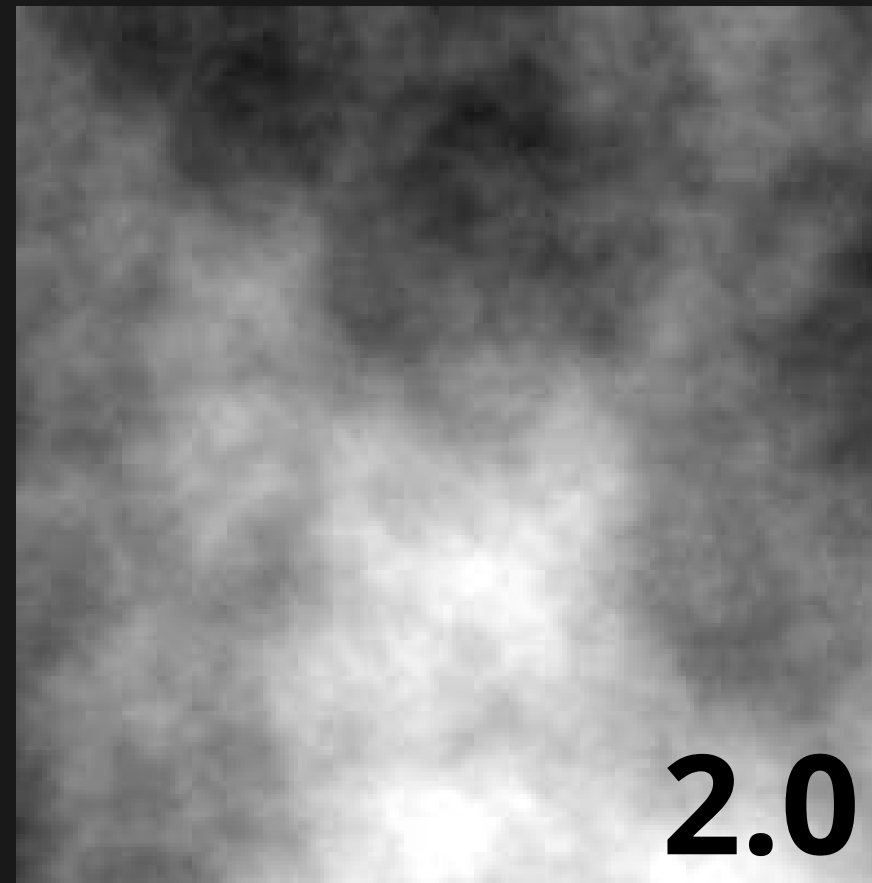
# Persistence

Número na qual  
determina o  
quanto cada oitava  
contribui para a  
forma geral, ajusta  
a amplitude.



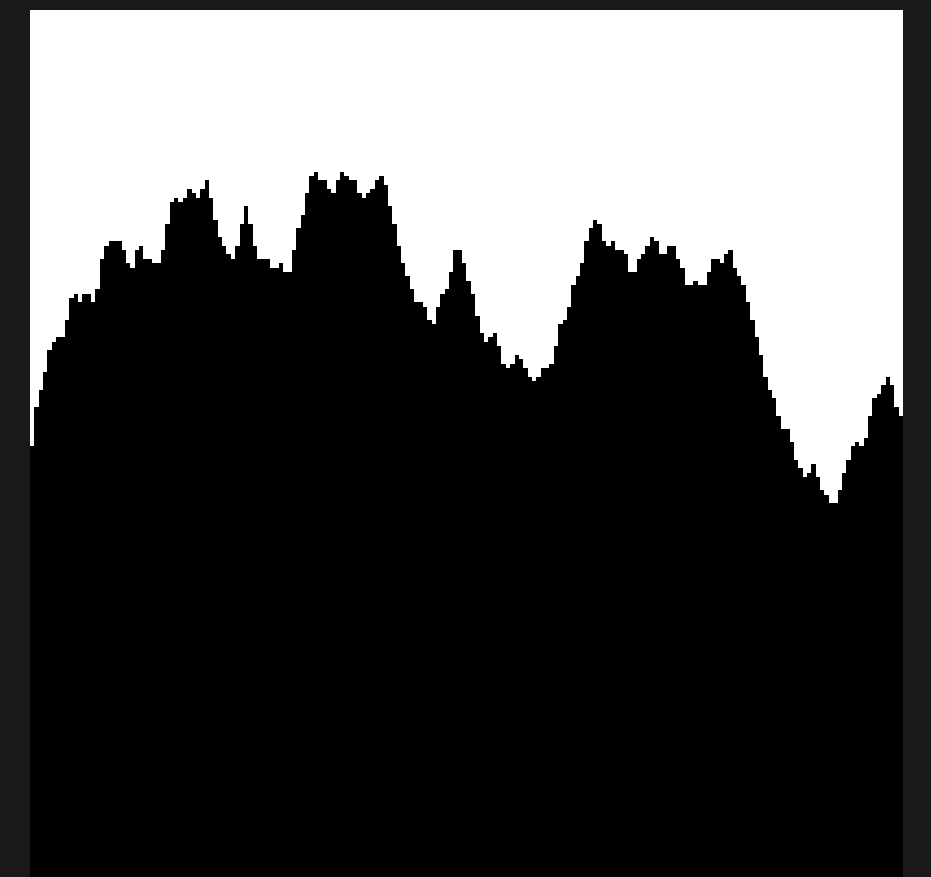
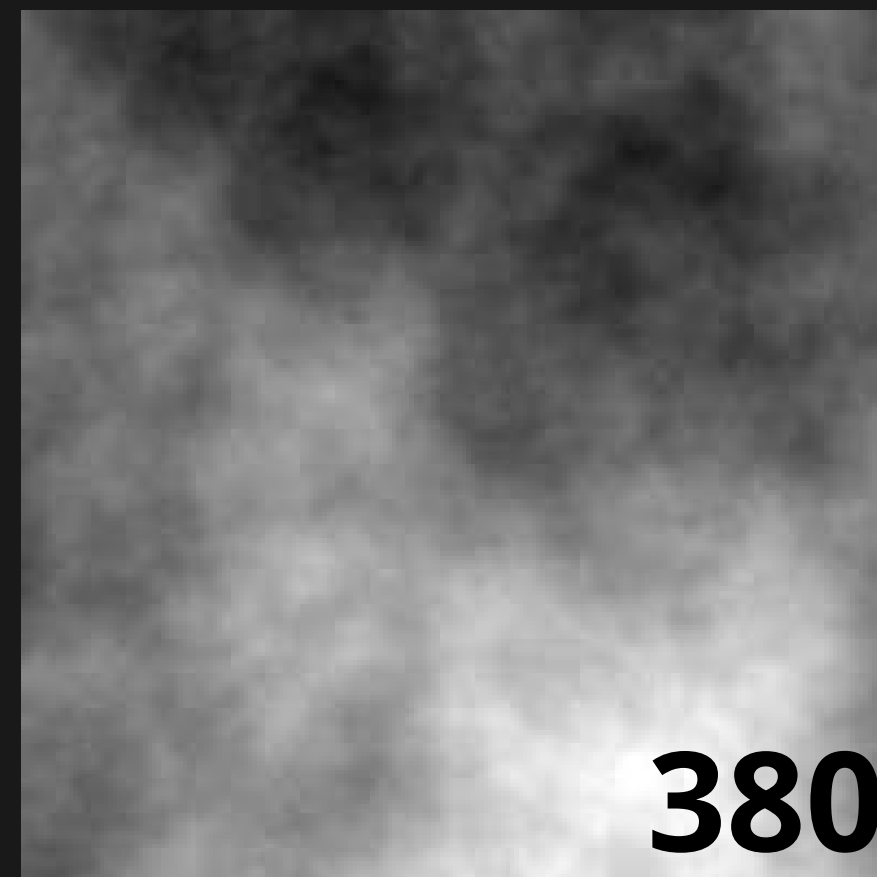
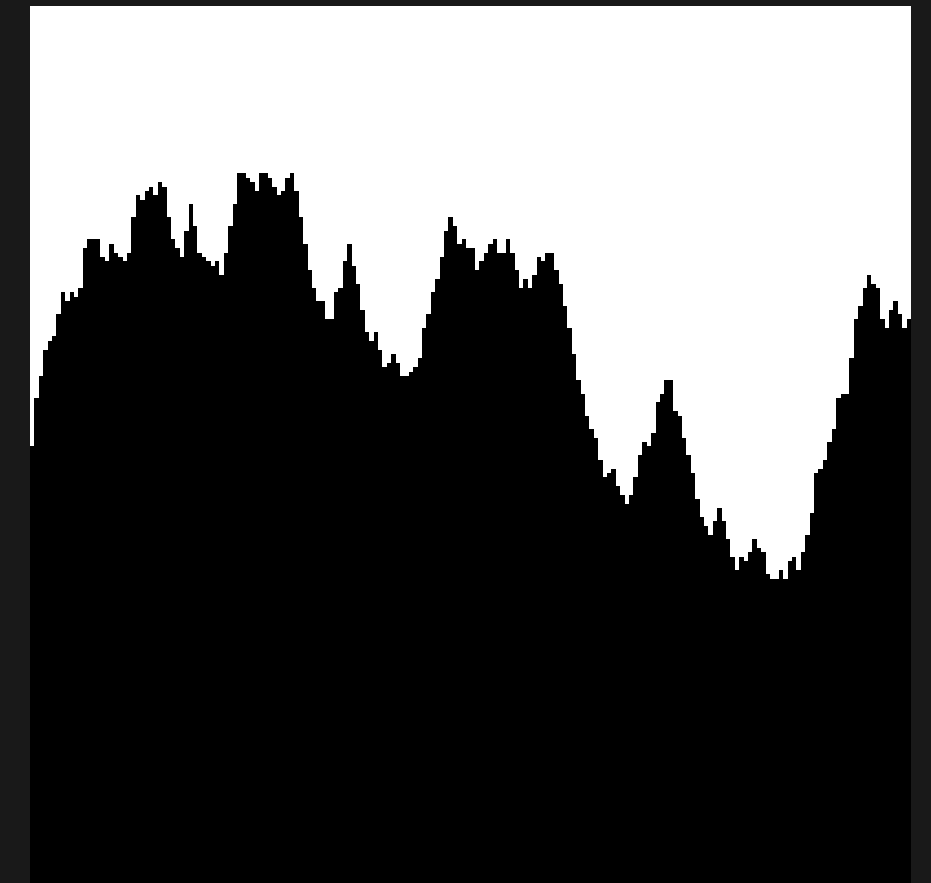
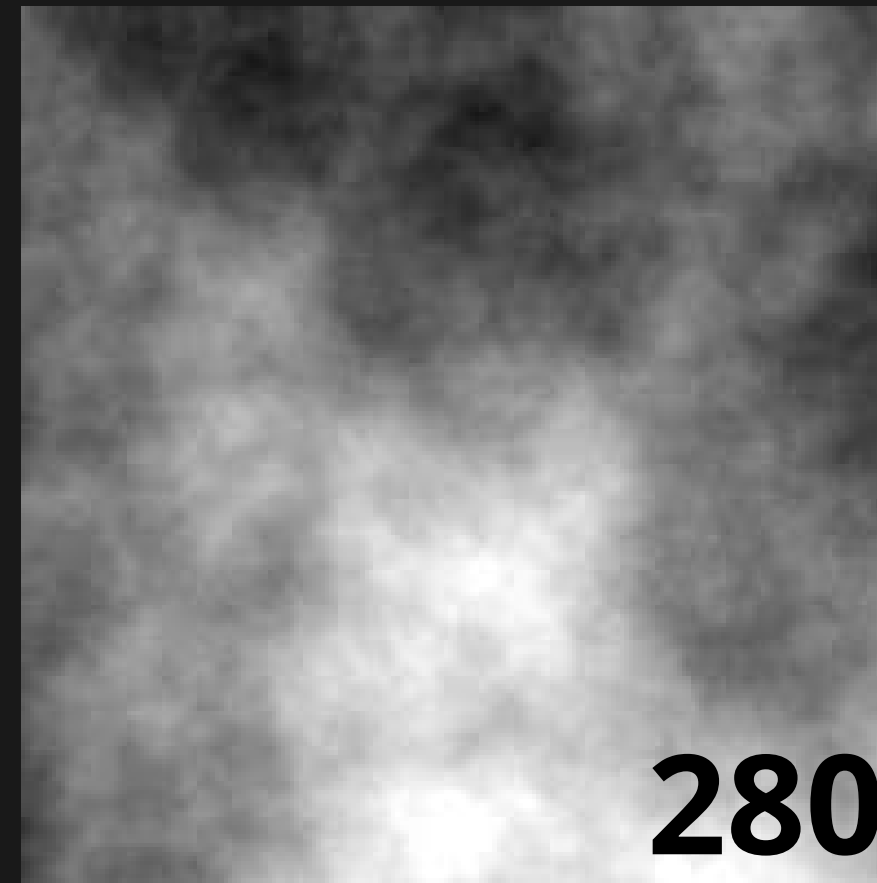
# Lacunarity

Número na qual  
determina quantos  
detalhes são  
adicionados ou  
removidos a cada  
oitava, ajusta a  
frequência.



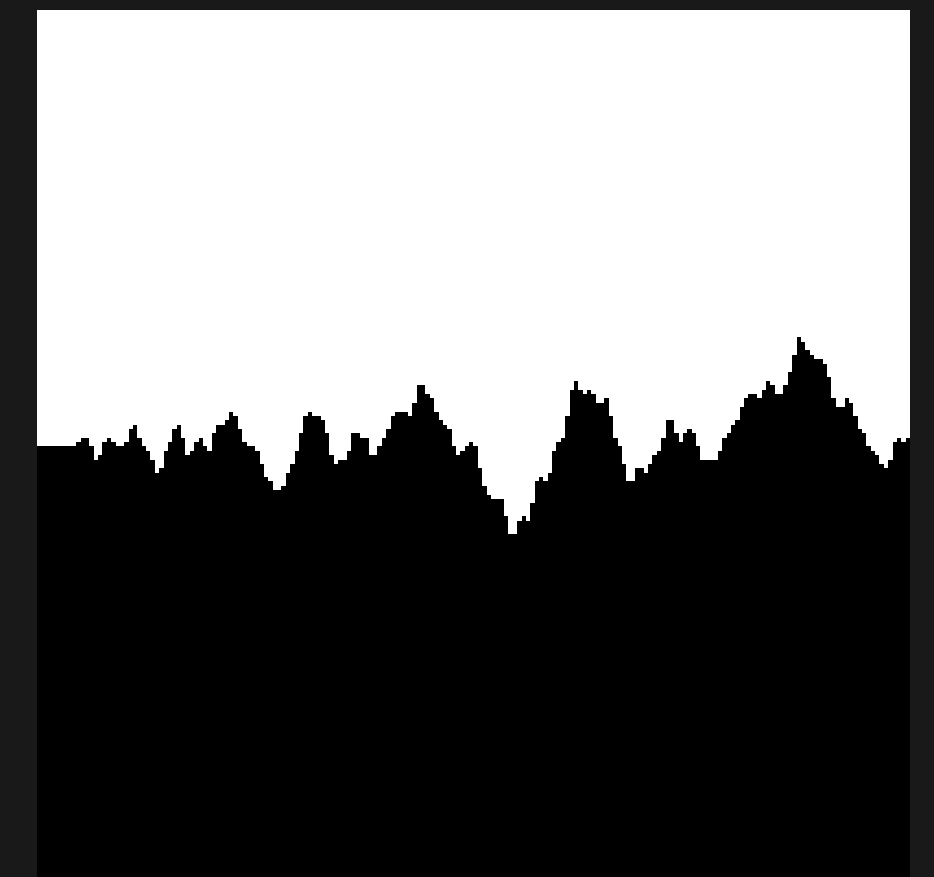
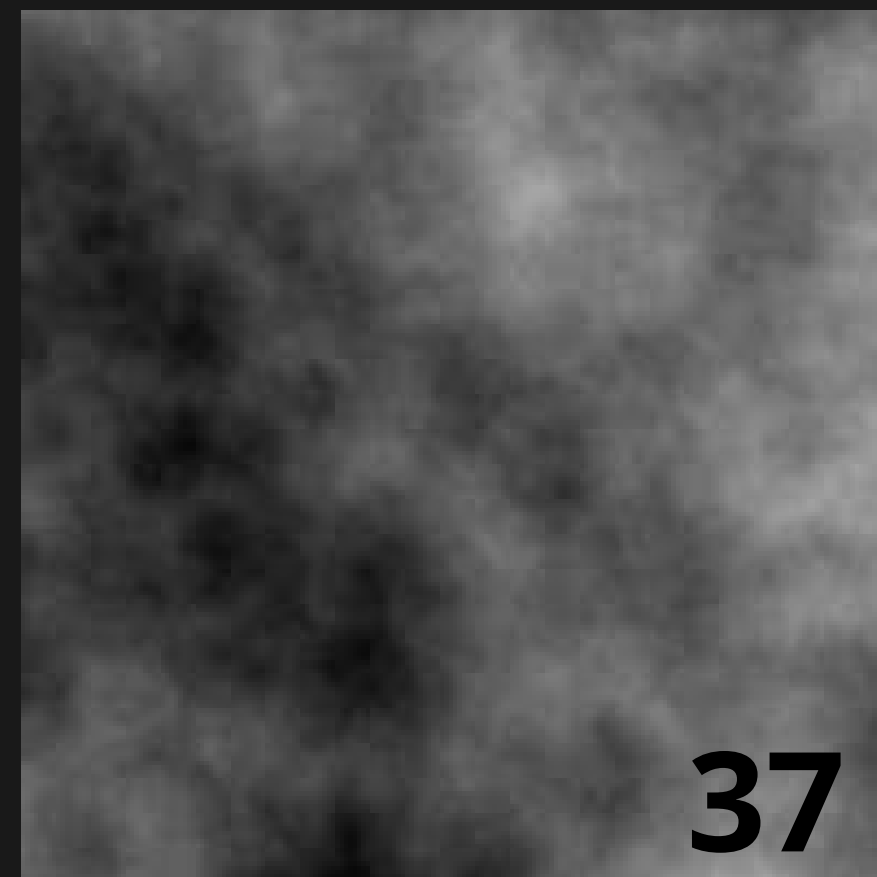
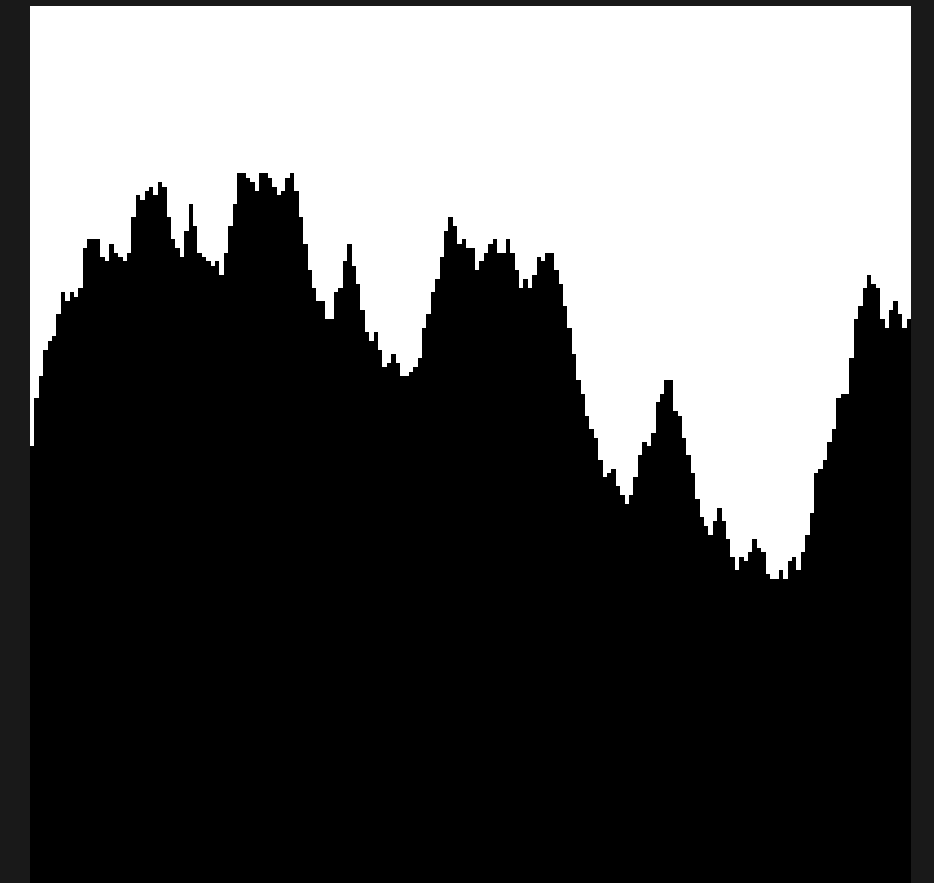
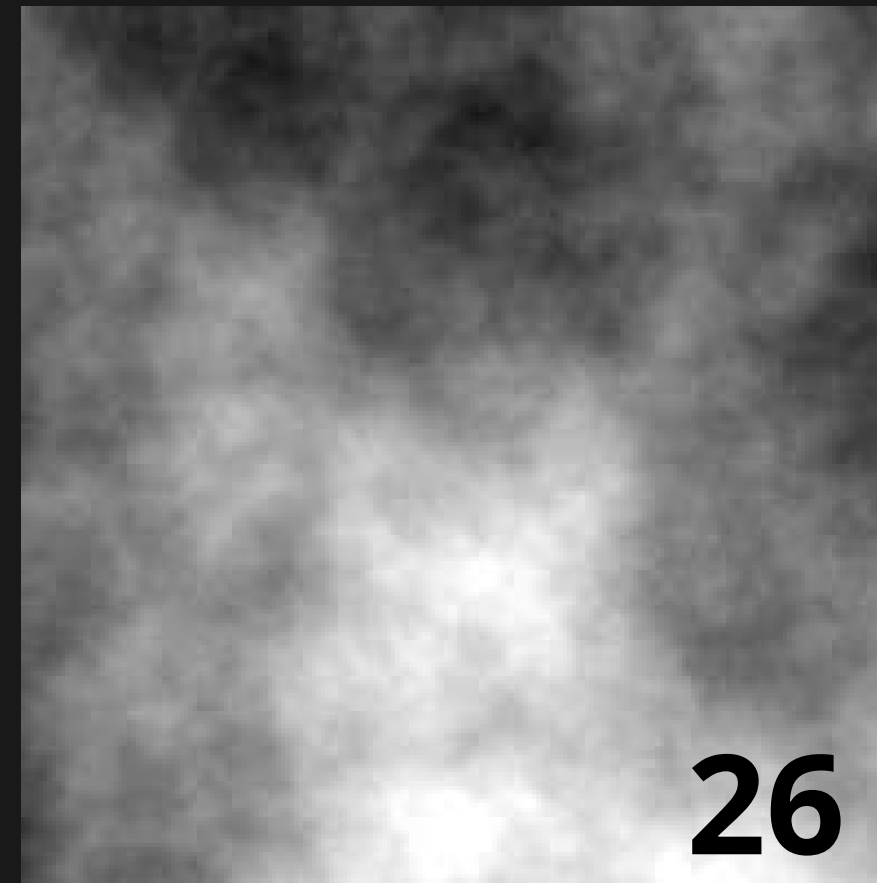
# Scale

Muda a escala do  
ruído.



# Seed

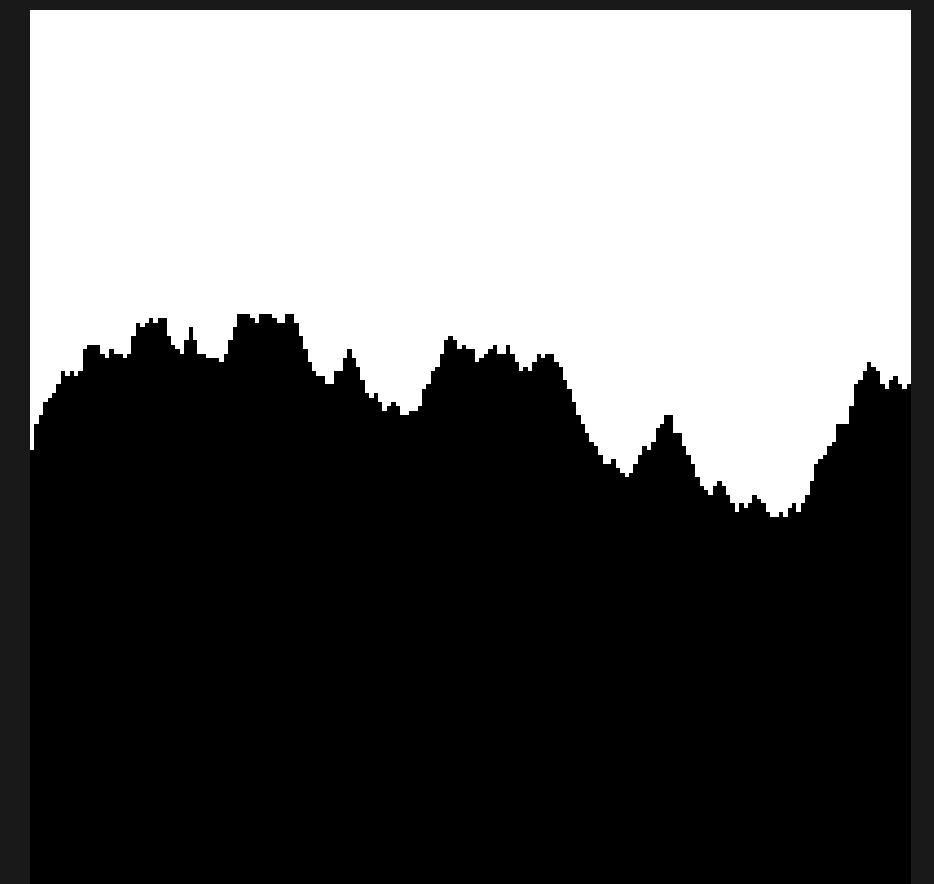
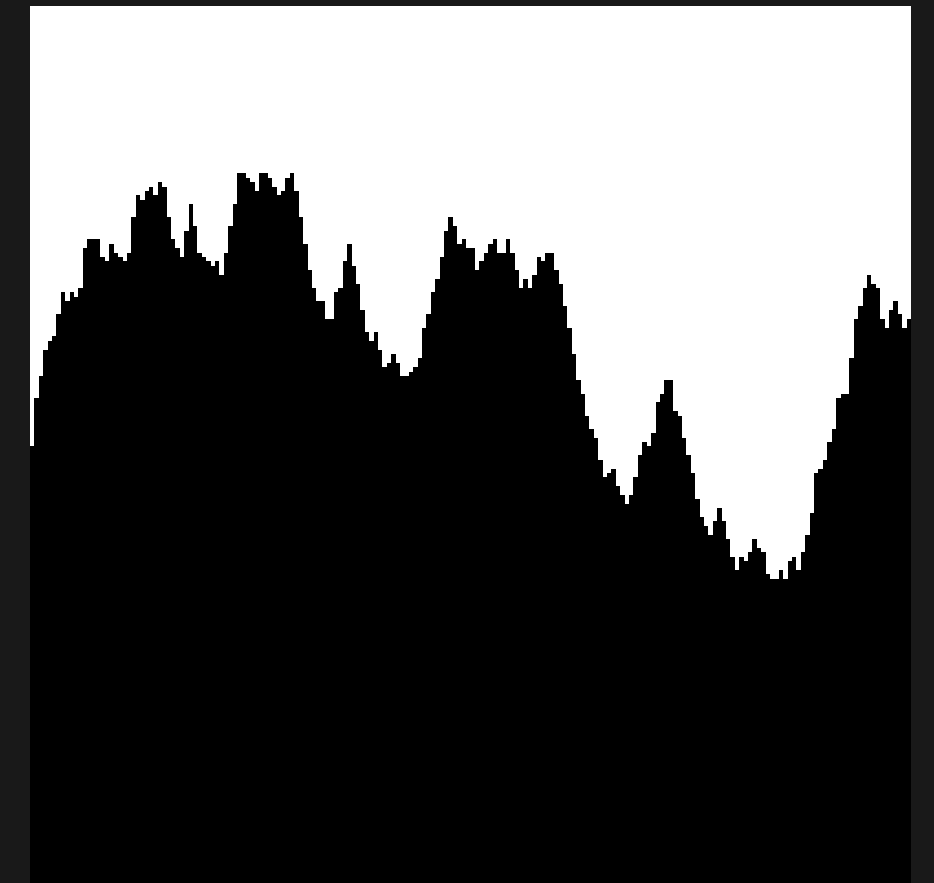
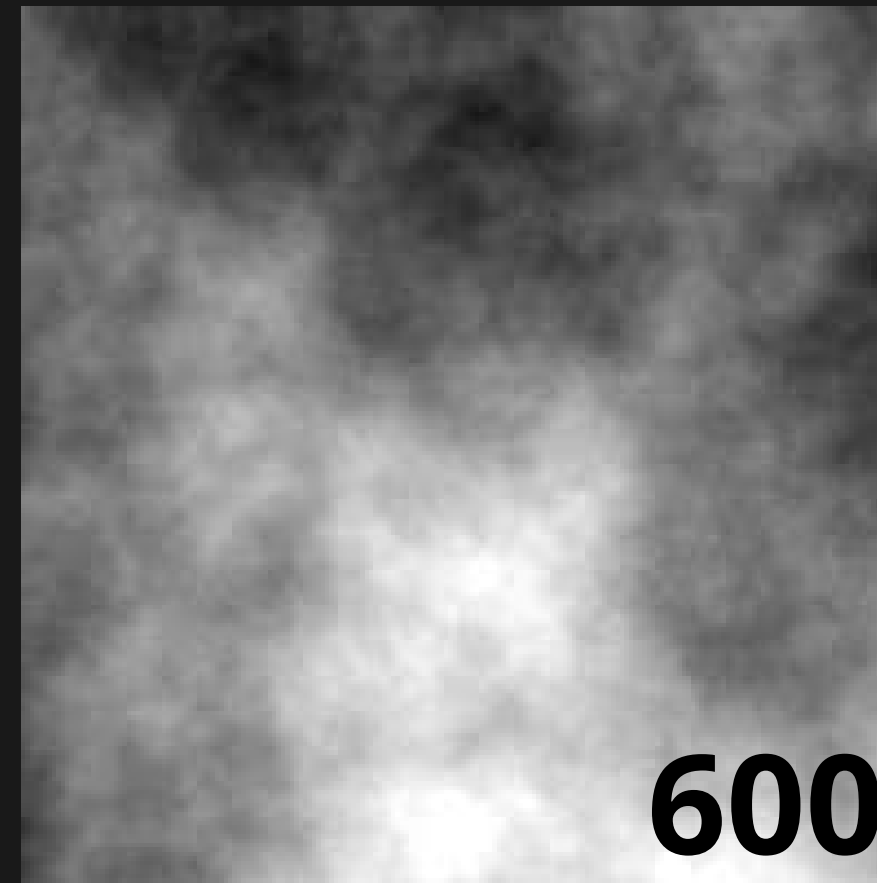
Id único para cada  
imagem gerada.





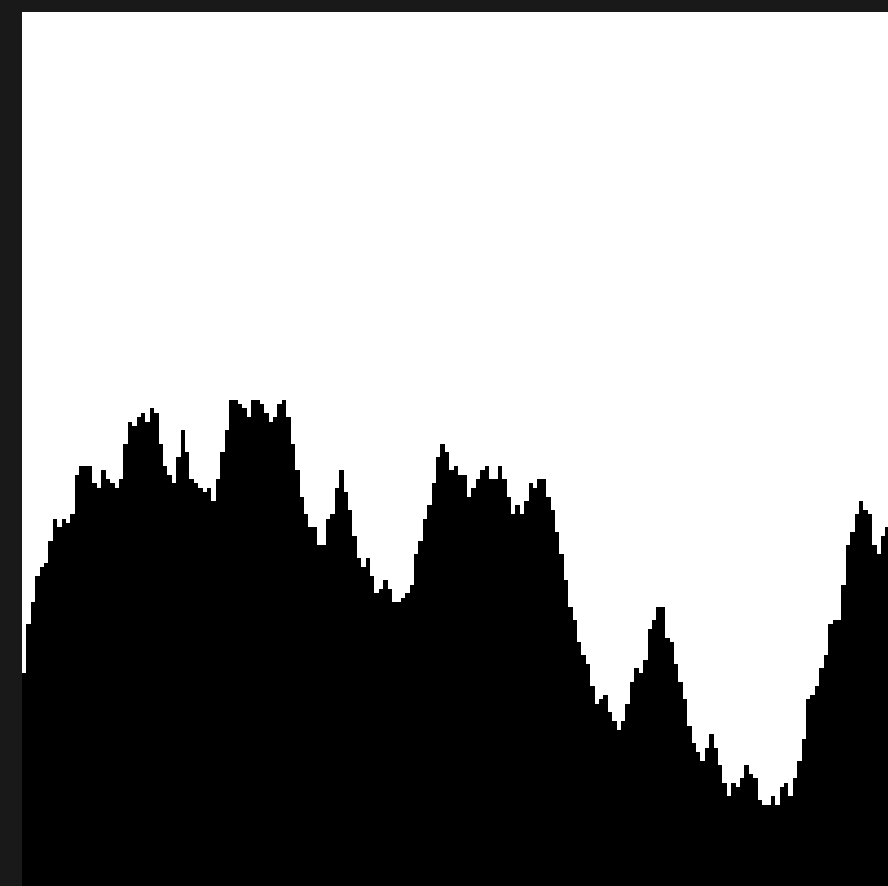
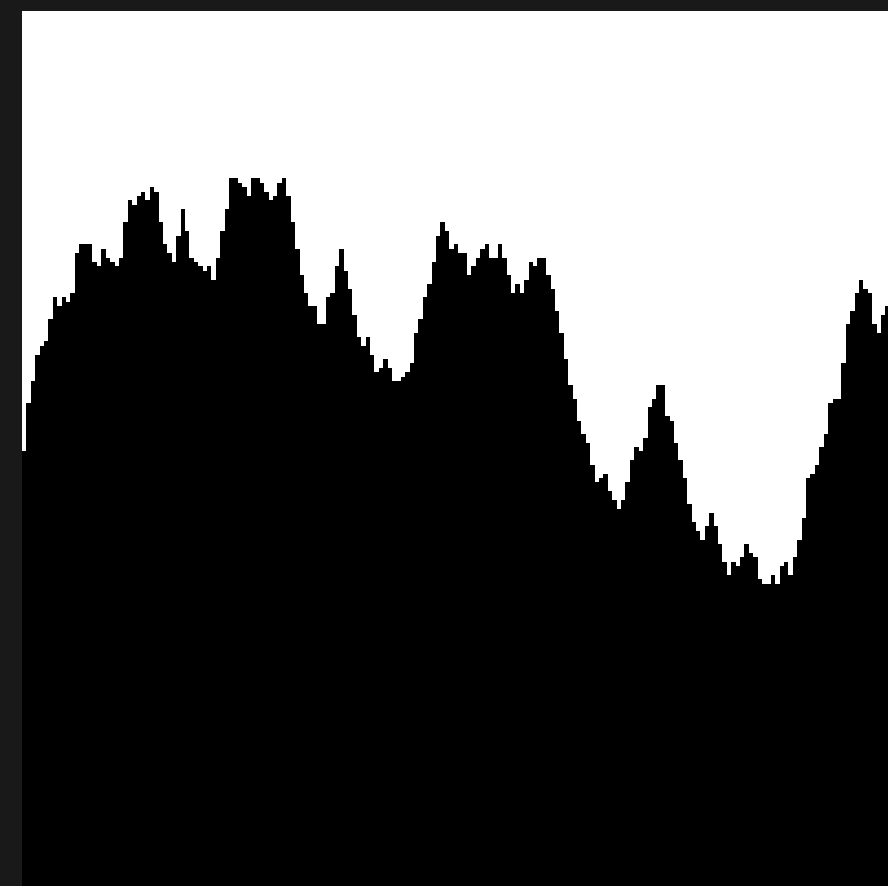
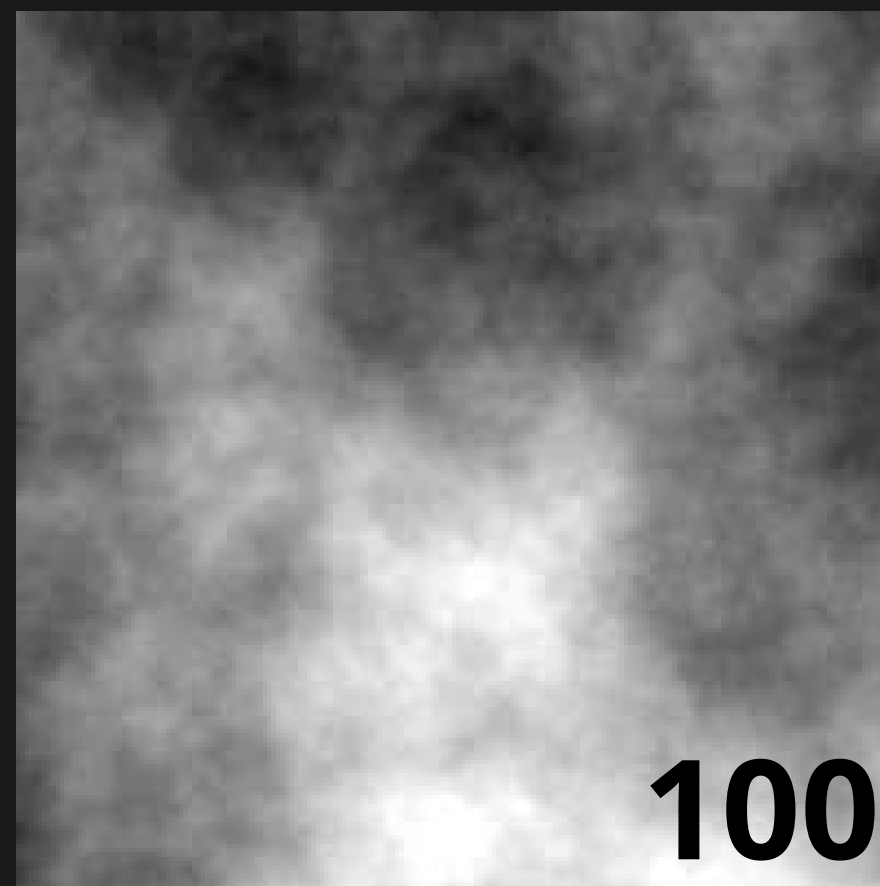
# Grau

Muda o valor do pixel através de um multiplicador.



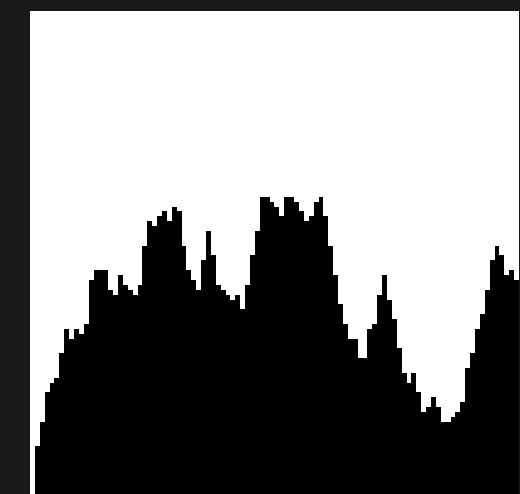
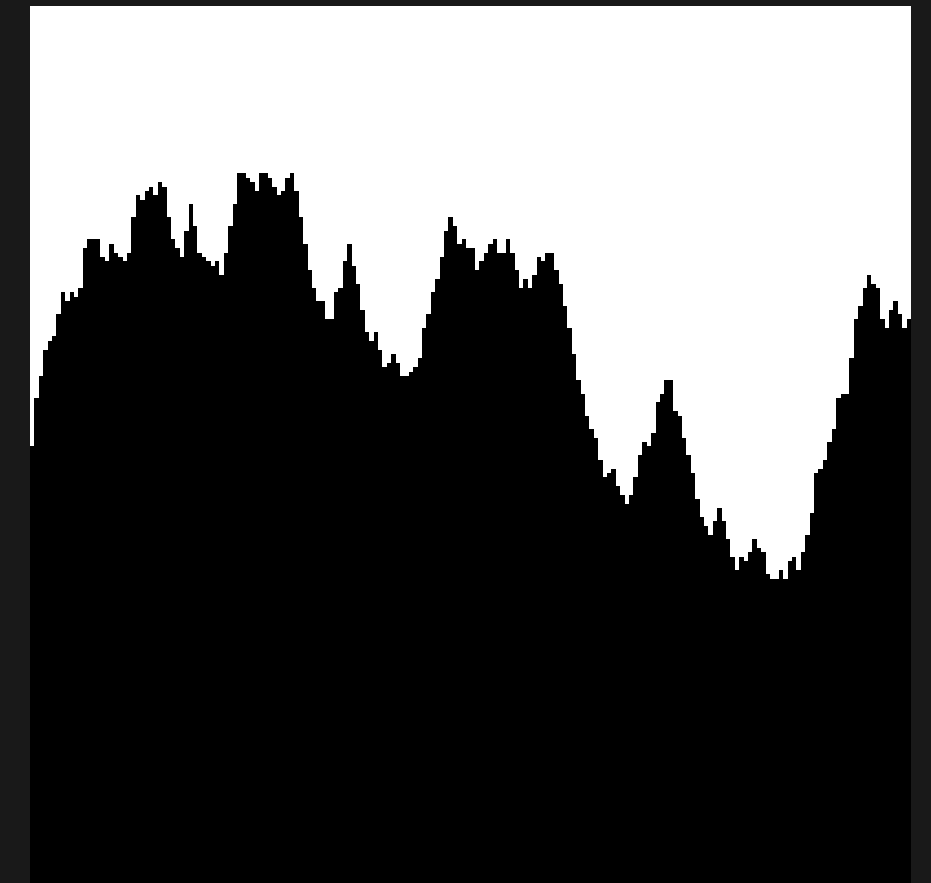
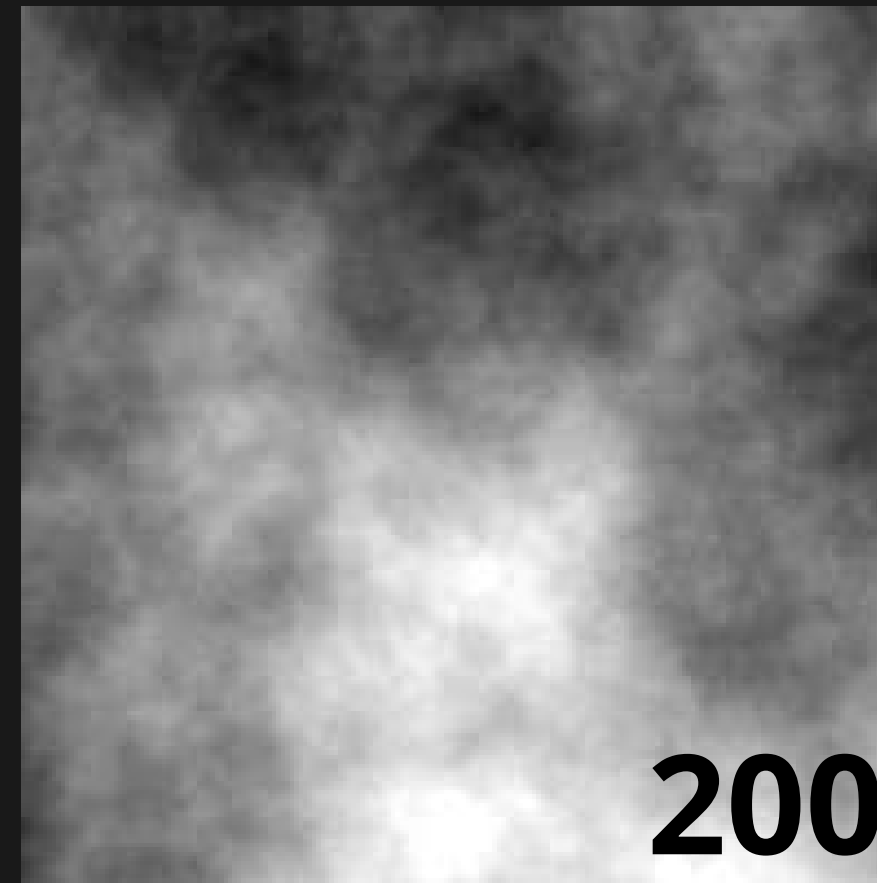
# Potencia

Muda o valor do pixel através de uma adição.

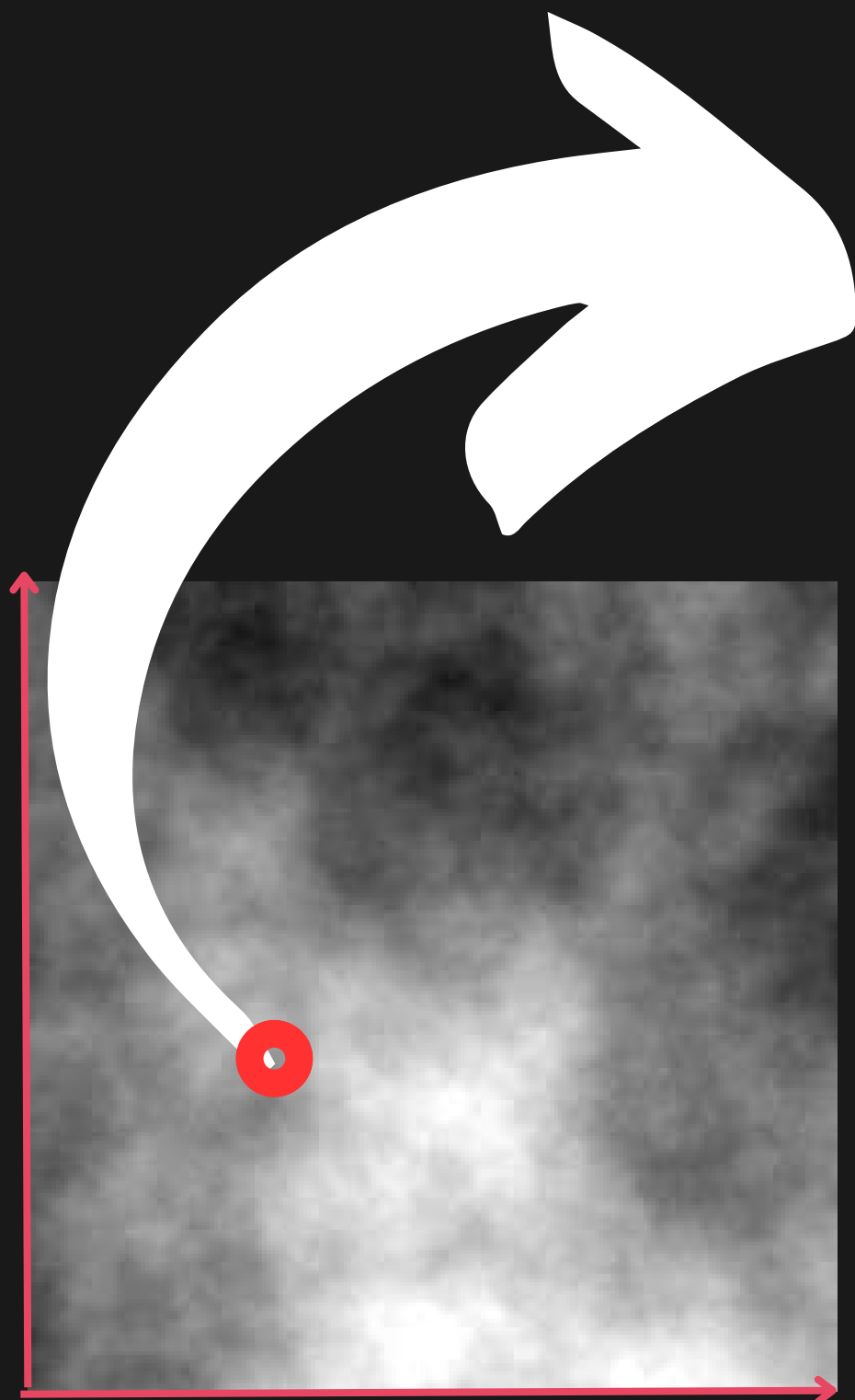


# Shape

Muda a escala da  
imagem.



# Mudando a escala de cores



$(x,y)$  ,  $(80,80,80, 255)$

posição do  
pixel na  
imagem

Cores Red,  
Green, Blue  
e camada  
Alpha de  
0 - 255

0



255

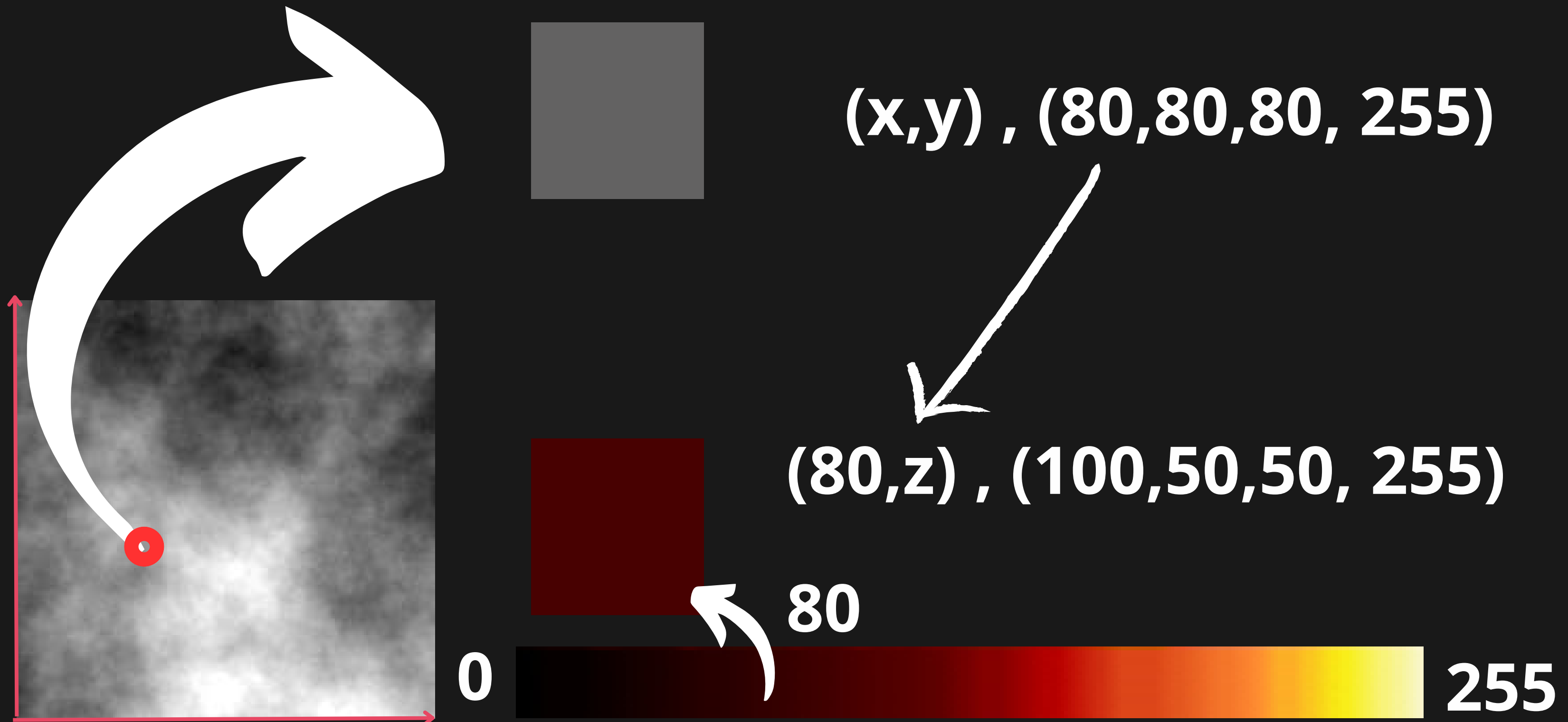
0



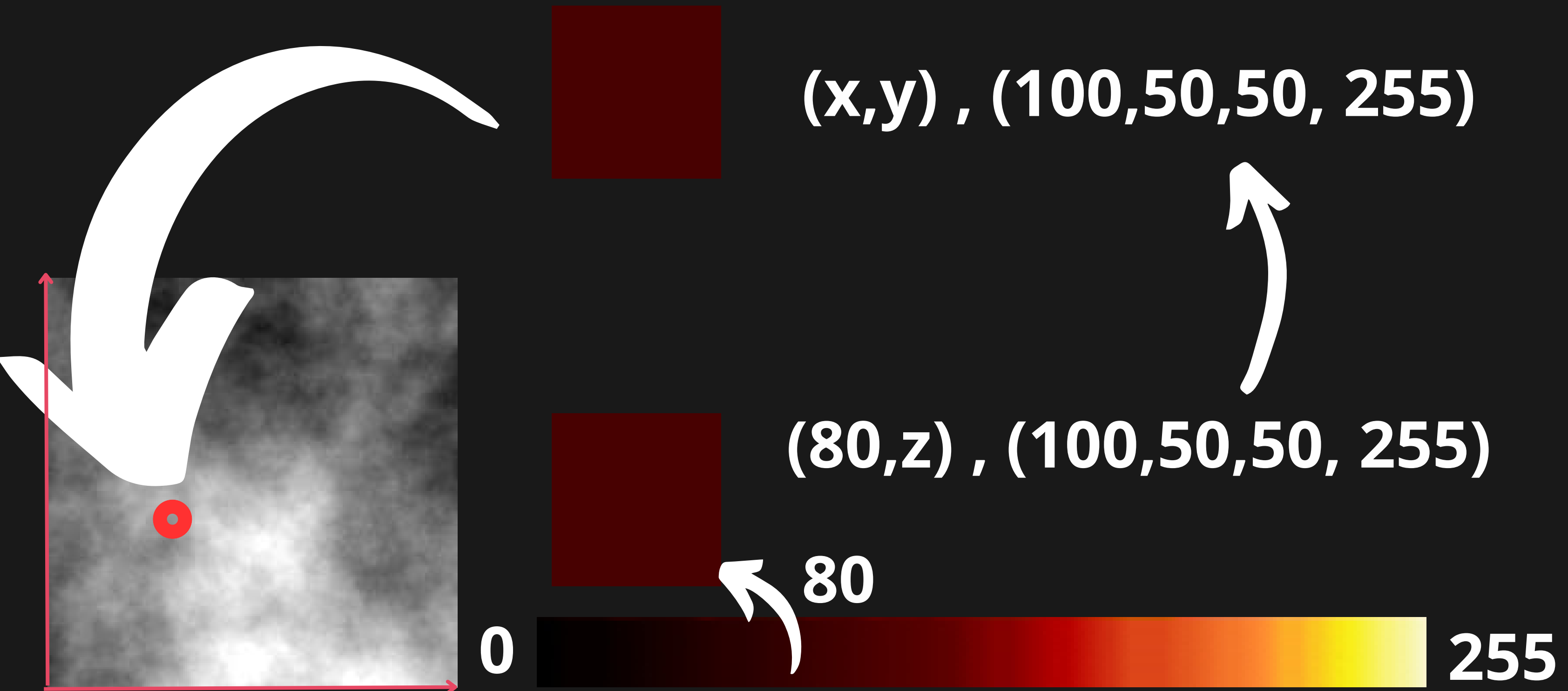
255



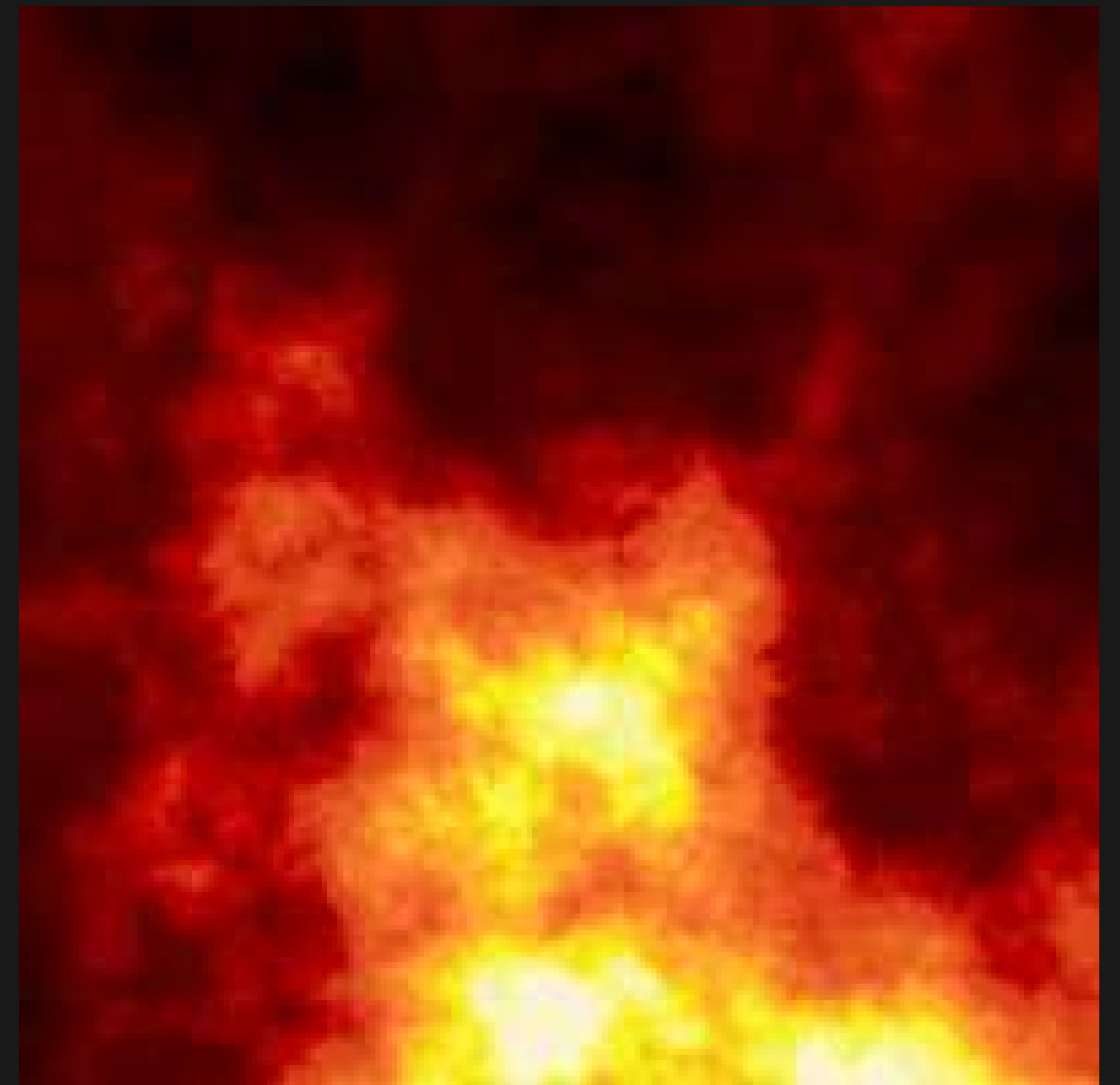
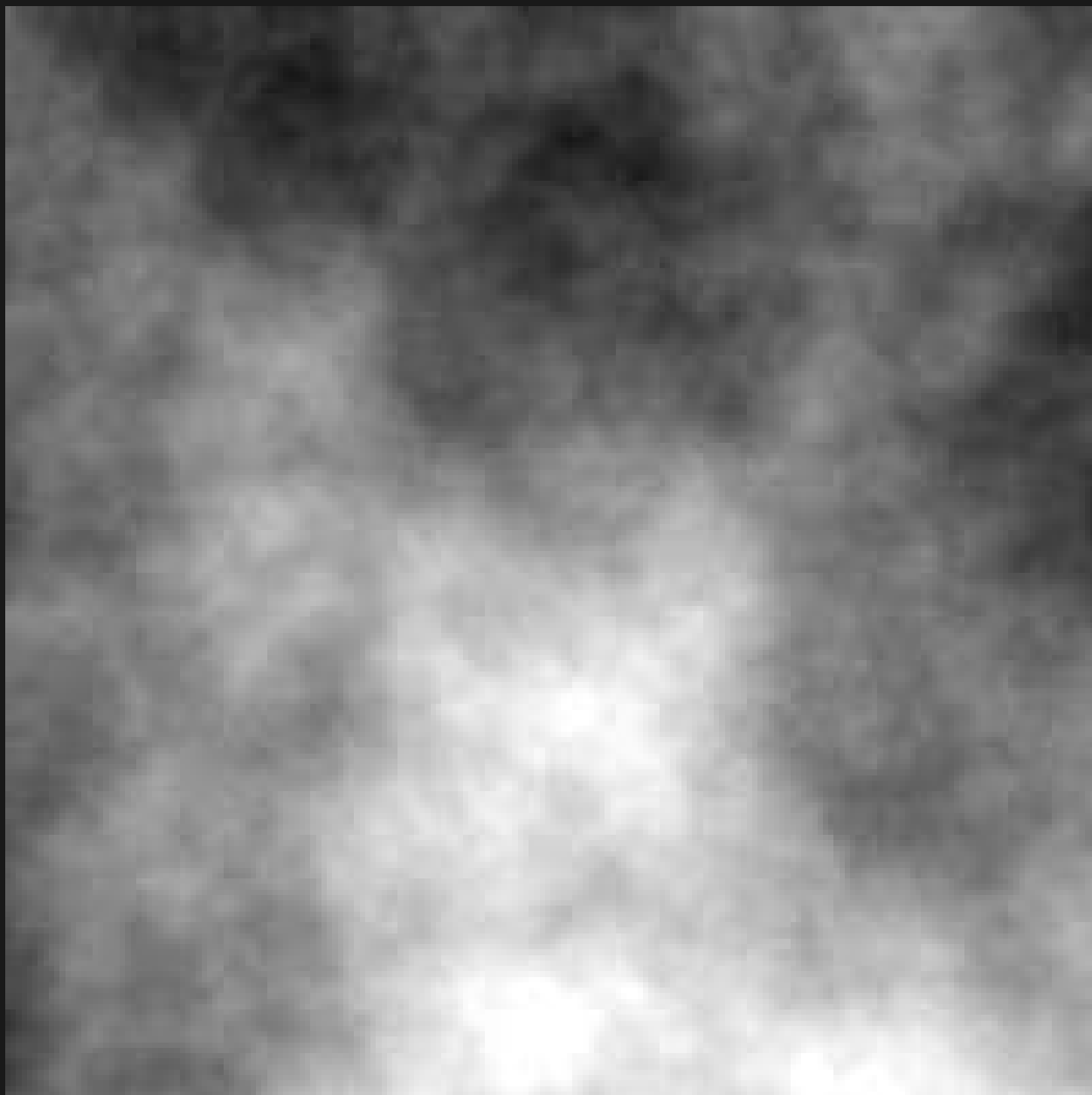
# Mudando a escala de cores



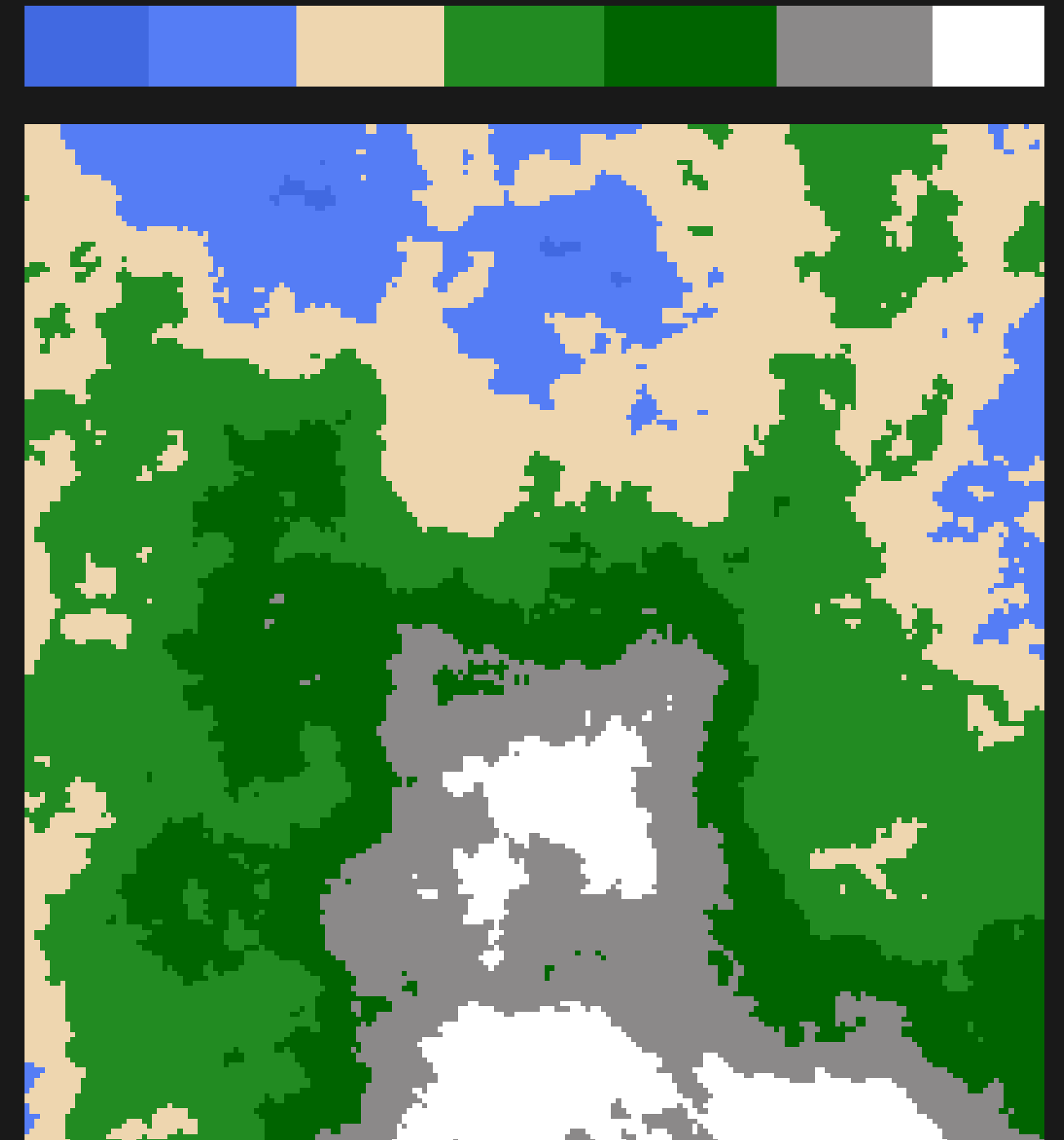
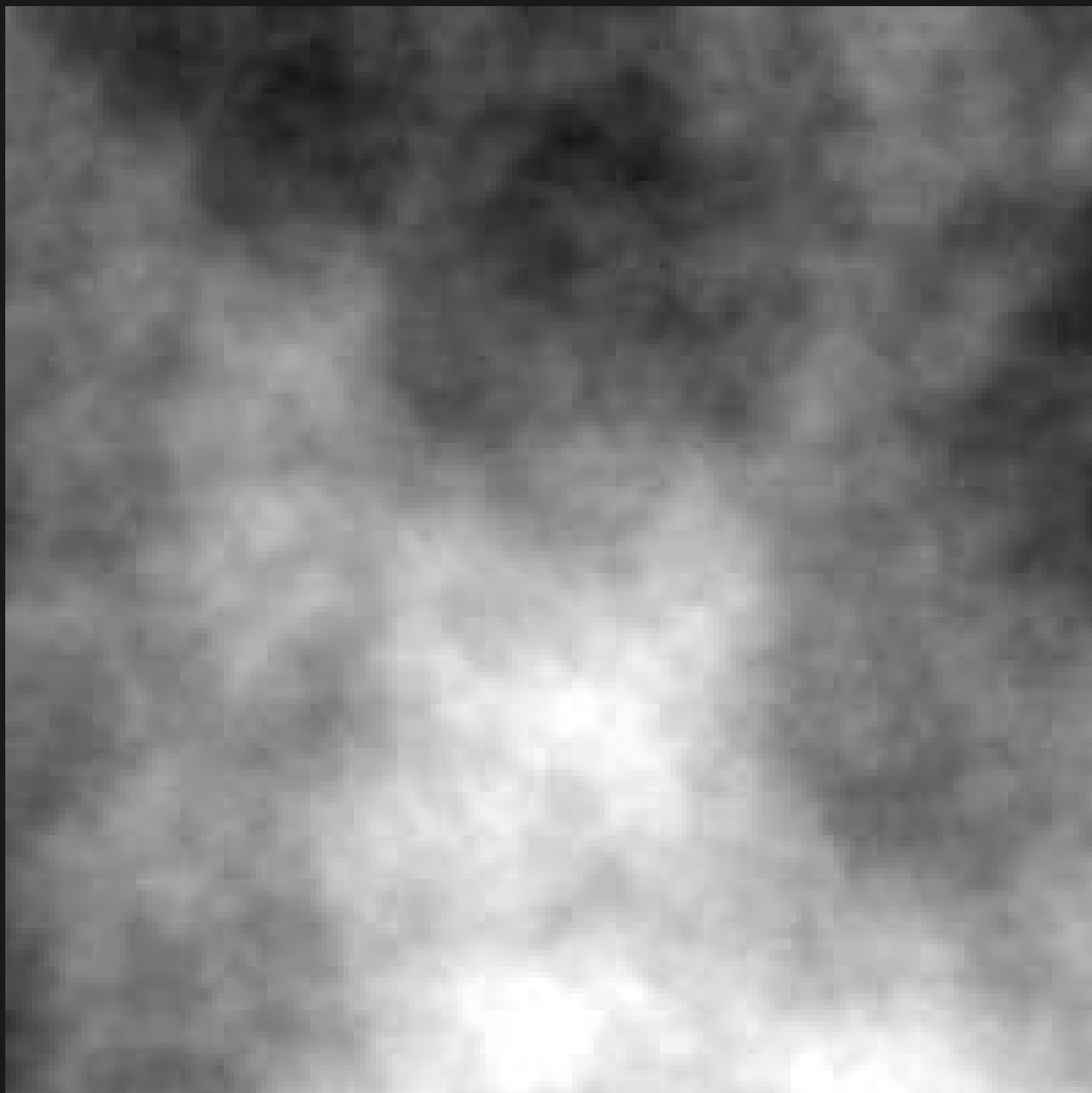
# Mudando a escala de cores




# Mudando a escala de cores



# Mudando a escala de cores





**Um mundo de  
possibilidades acaba de se  
abrir.**

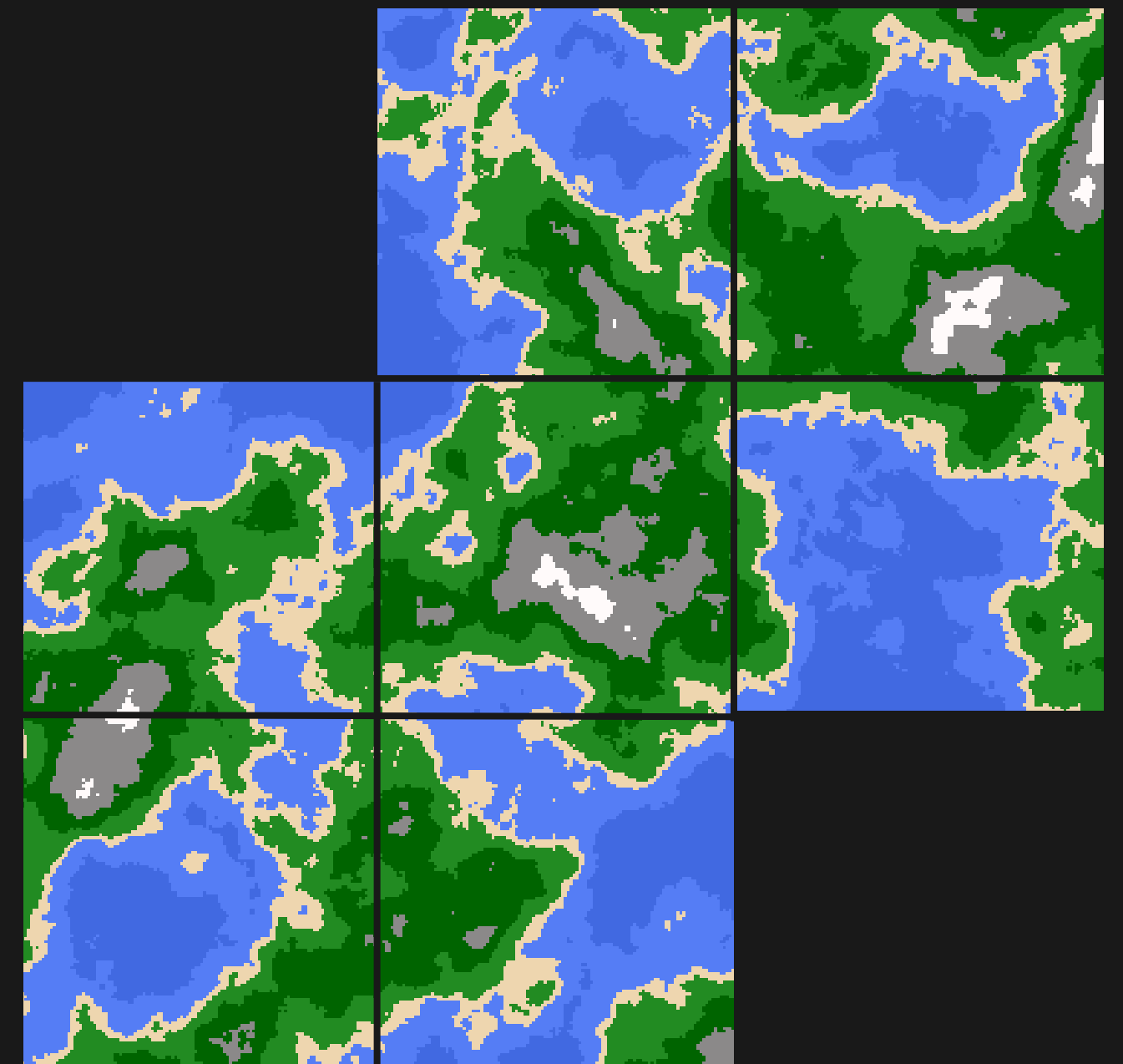
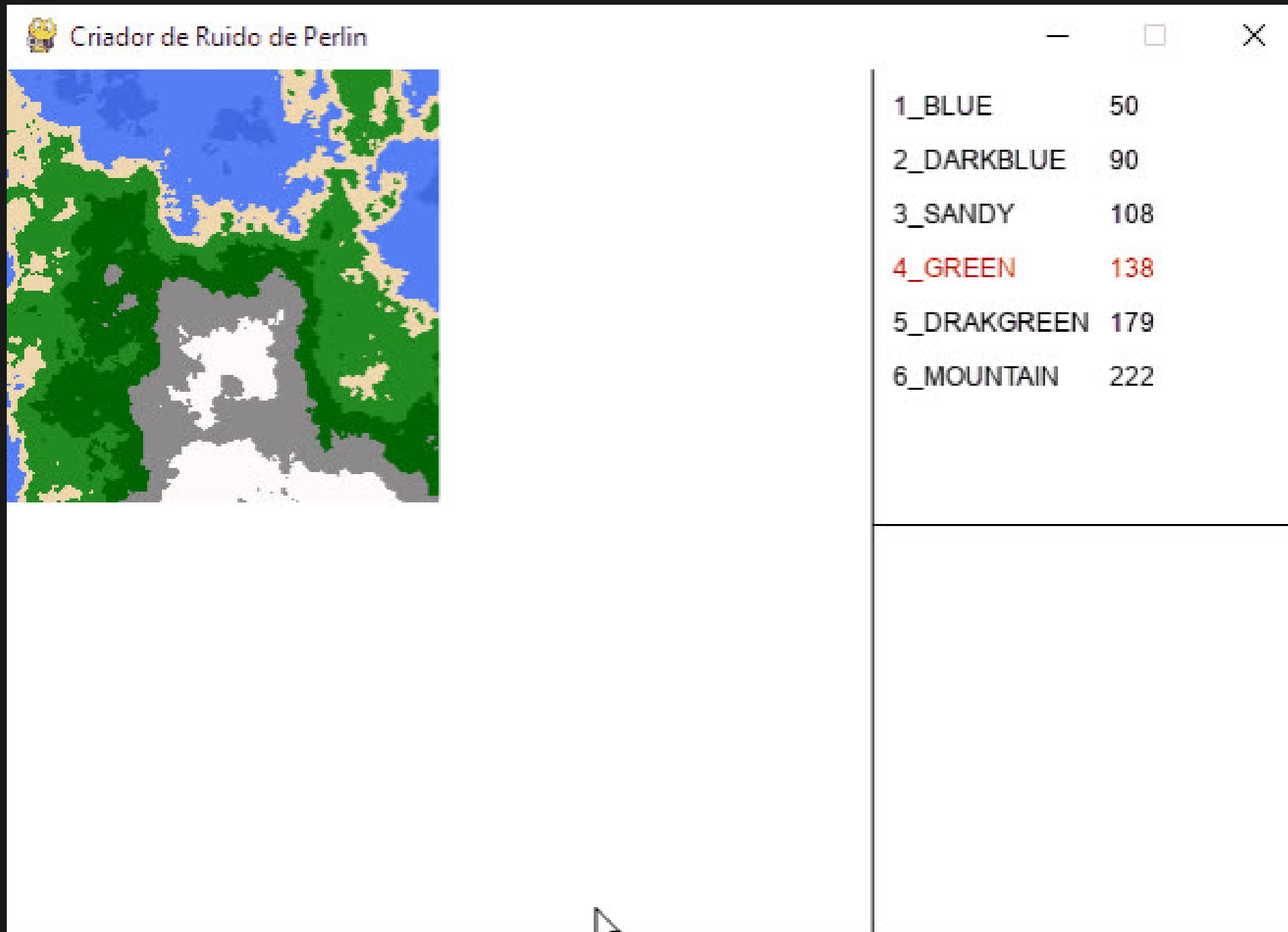


# Don't Starve Together

Jogo 2D em visão top down

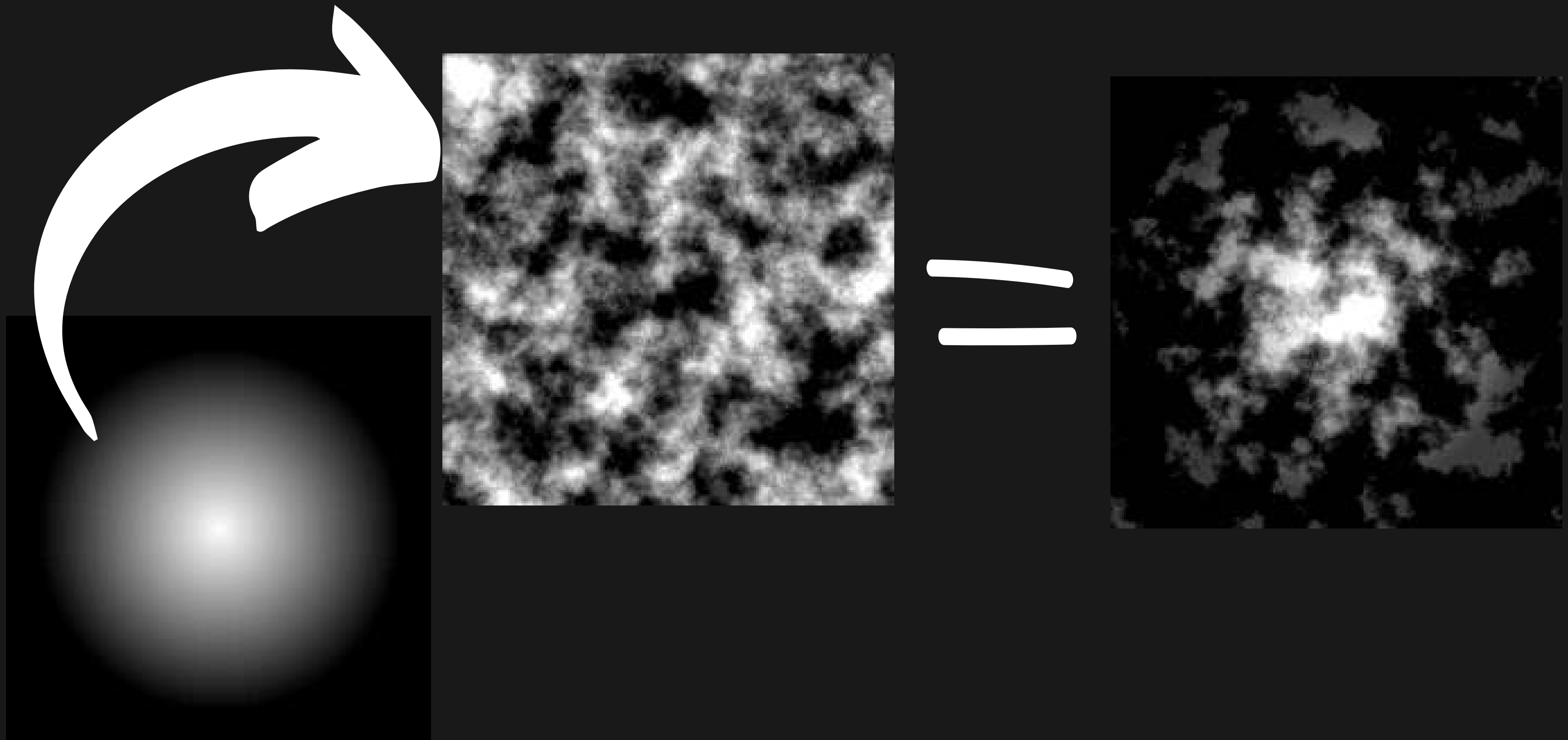


# DESAFIO: Como criar um arquipelago?





# Criando um filtro



# Desafio cumprido!



# Minecraft

Jogo 3D estilo sand box



**DESAFIO: Como criar e  
visualizar um ruído 3D?**



# Criando ruído 3D

## help(noise)

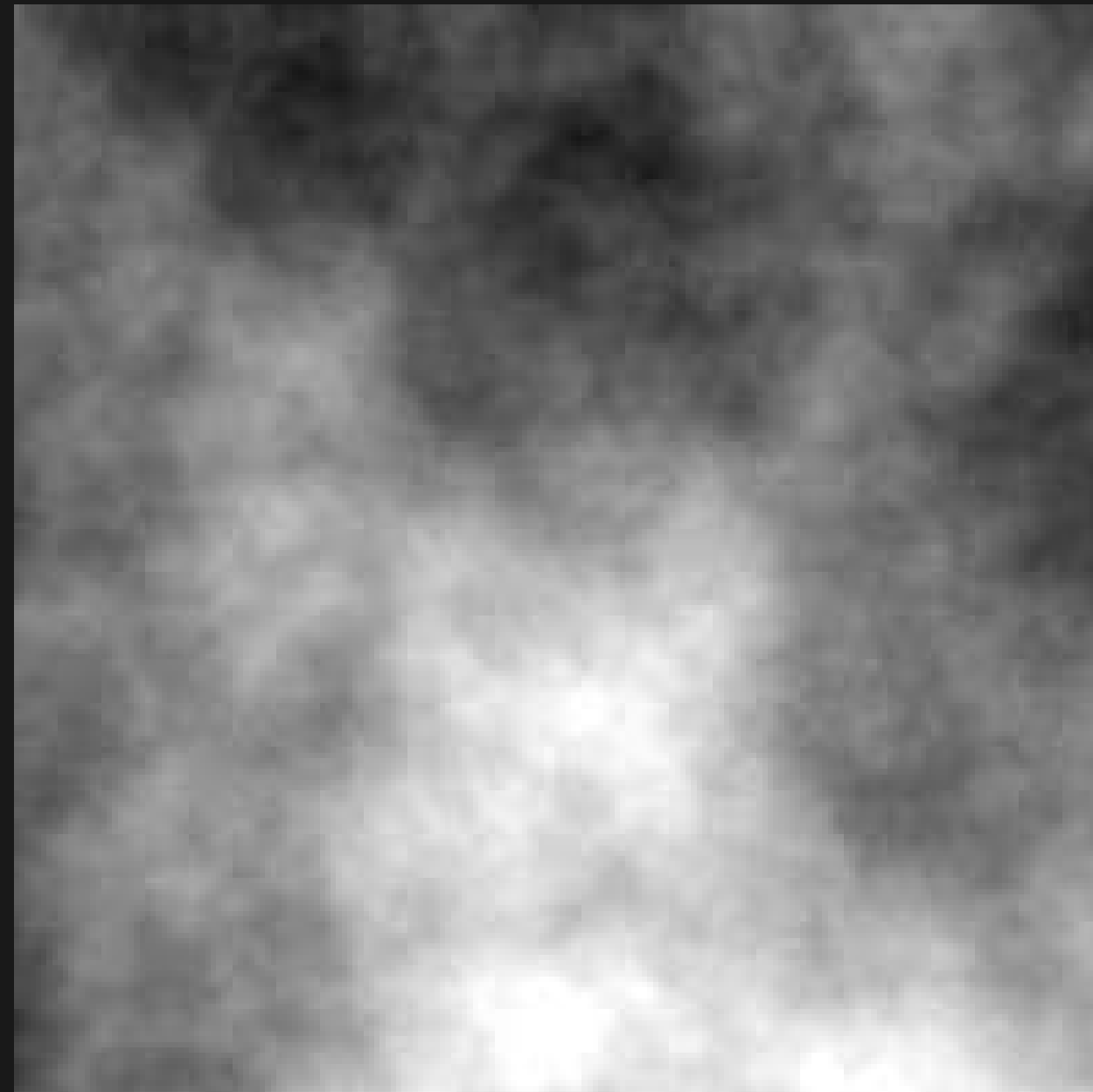
*pnoise2 = noise2(...)*

*noise2(x, y, octaves=1, persistence=0.5, lacunarity=2.0, repeatx=1024,  
repeaty=1024, base=0.0)*

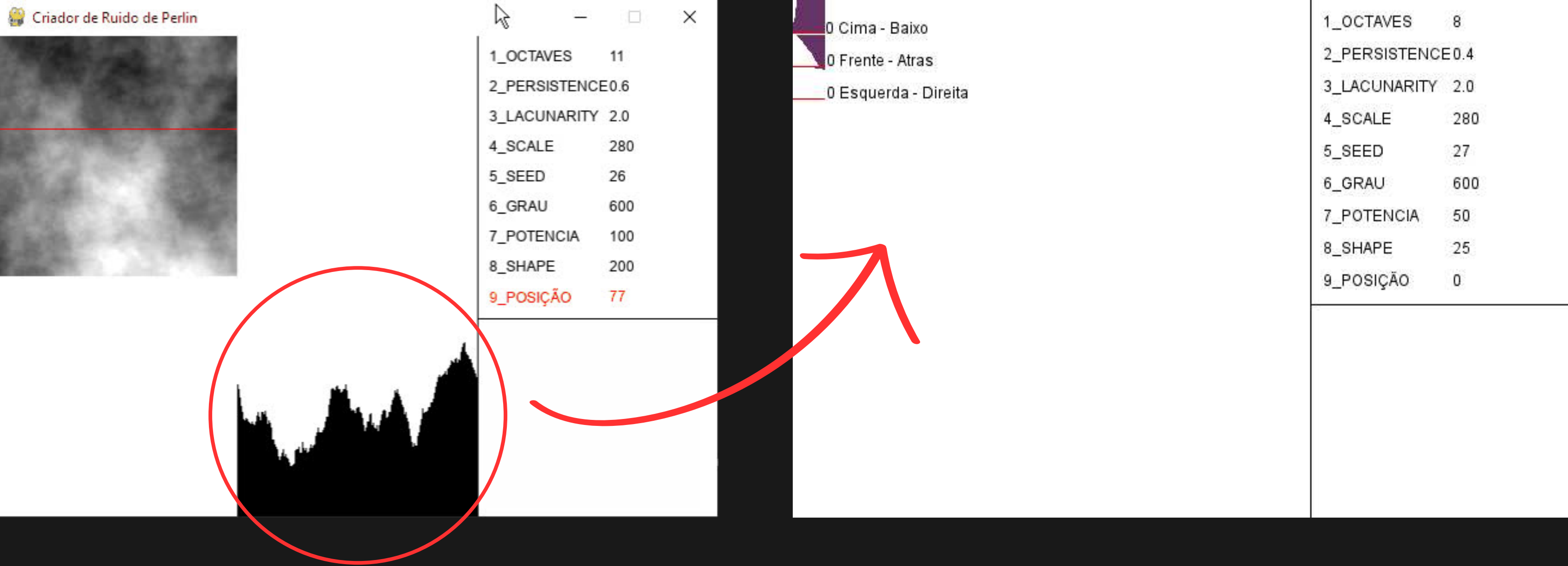
*pnoise3 = noise3(...)*

*noise3(x, y, **z**, octaves=1, persistence=0.5, lacunarity=2.0, repeatx=1024,  
repeaty=1024, repeatz=1024, base=0.0)*

# Criando ruído 3D



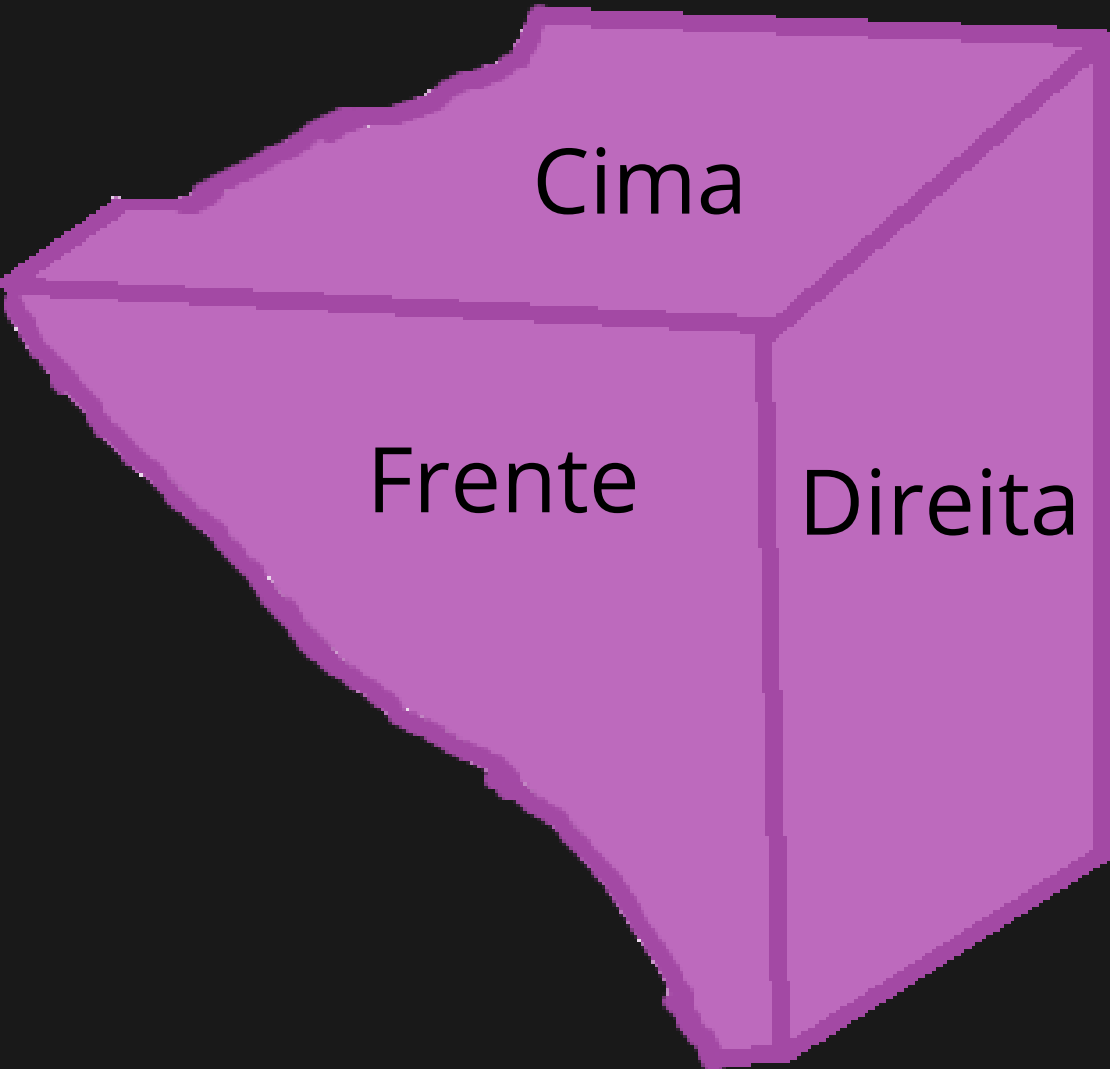
# Visualizando ruído 3D



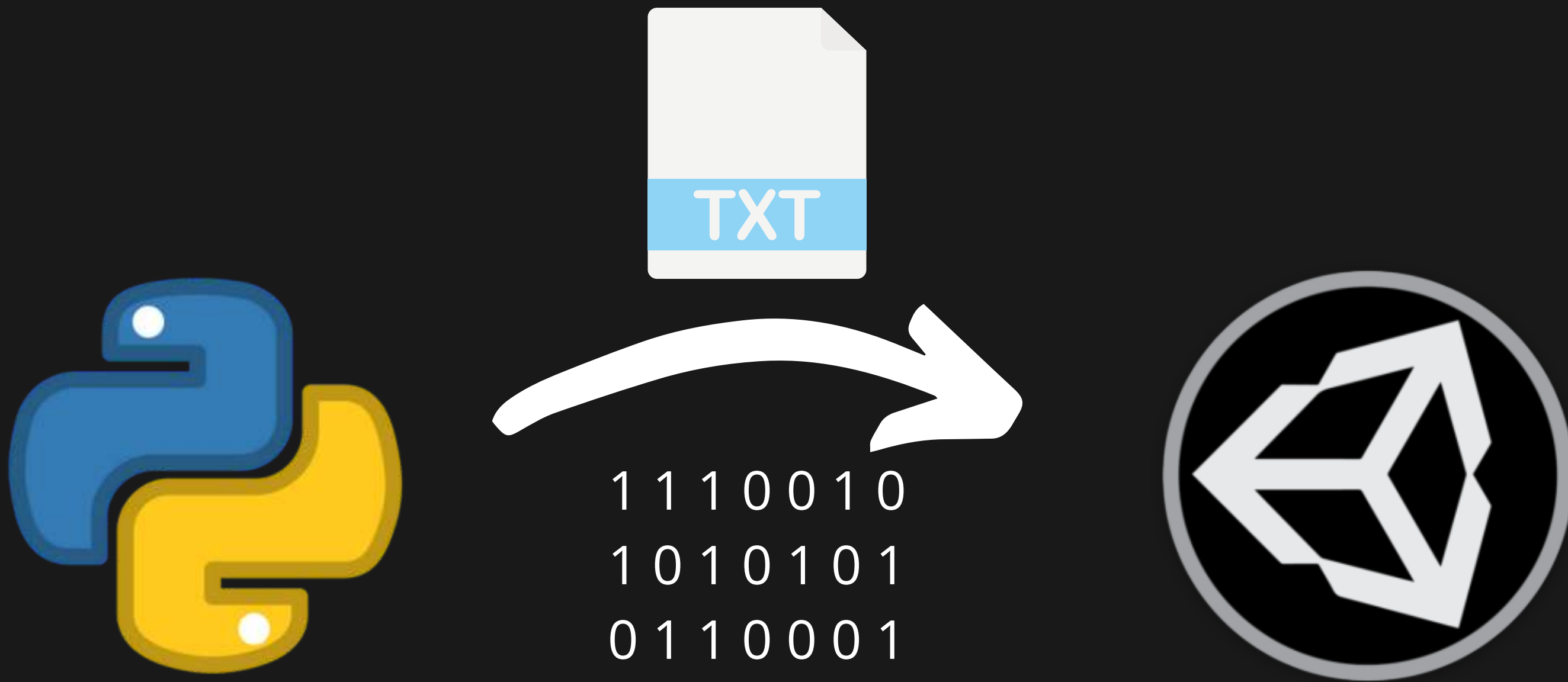


# Visualizando ruído 3D

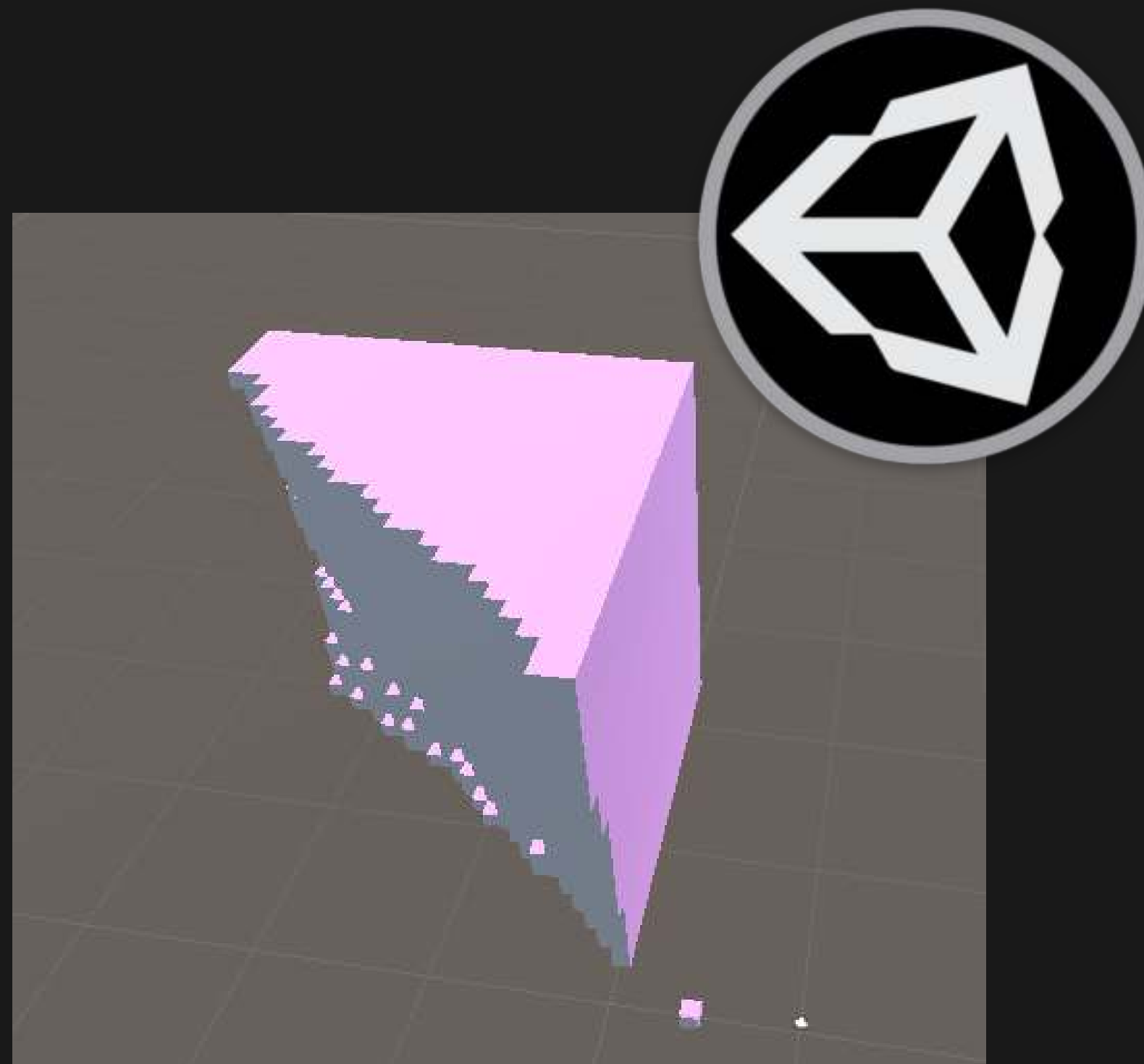
|   |               |     |
|---|---------------|-----|
|  A vertical sidebar on the left side of the interface. It contains three radio button controls for viewing the 3D noise from different perspectives: '0 Cima - Baixo' (Top - Bottom), '0 Frente - Atras' (Front - Back), and '0 Esquerda - Direita' (Left - Right). Each option is preceded by a small square icon with a diagonal line. | 1_OCTAVES     | 8   |
|   | 2_PERSISTENCE | 0.4 |
|   | 3_LACUNARITY  | 2.0 |
|   | 4_SCALE       | 280 |
|   | 5_SEED        | 27  |
|   | 6_GRAU        | 600 |
|   | 7_POTENCIA    | 50  |
|   | 8_SHAPE       | 25  |
|   | 9_POSIÇÃO     | 0   |
|   |               |     |



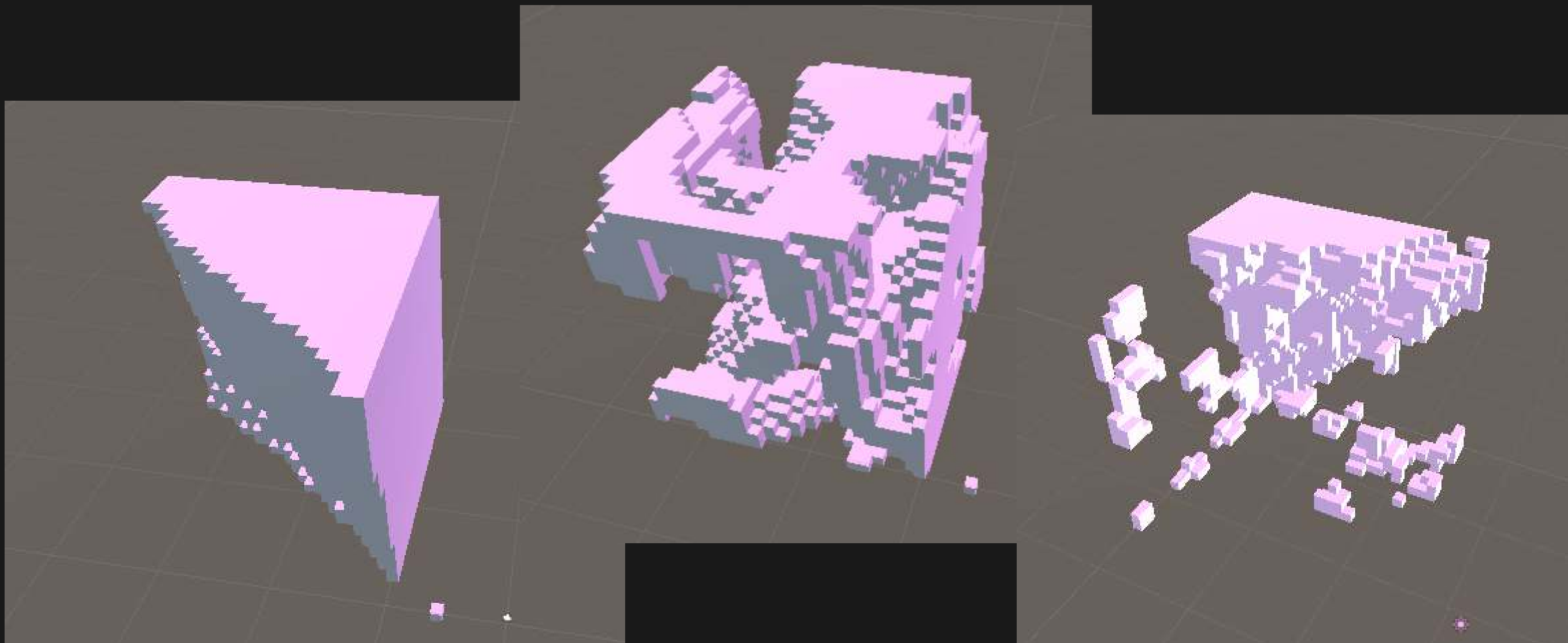
# Passando o ruído do Python para o Unity



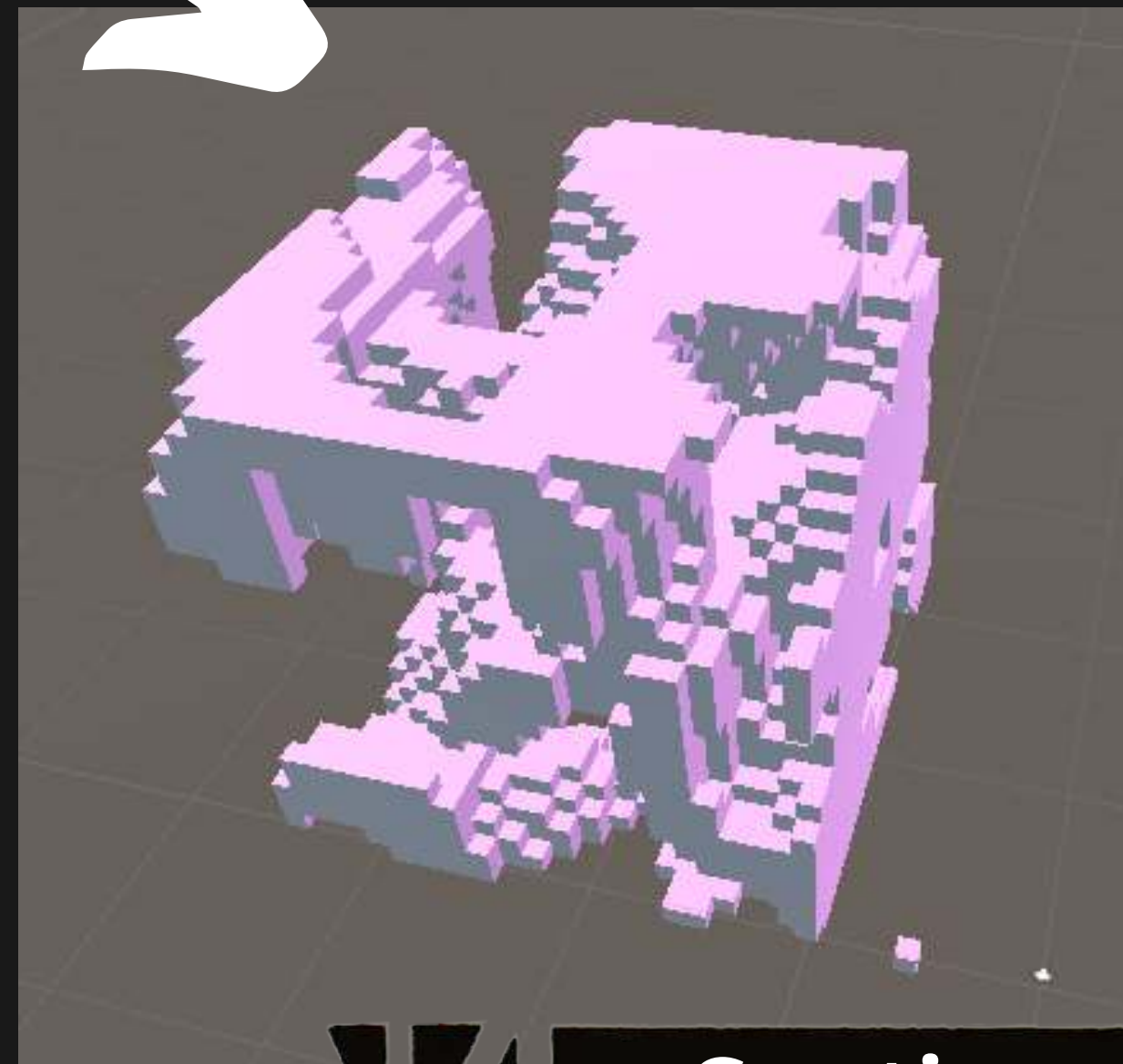
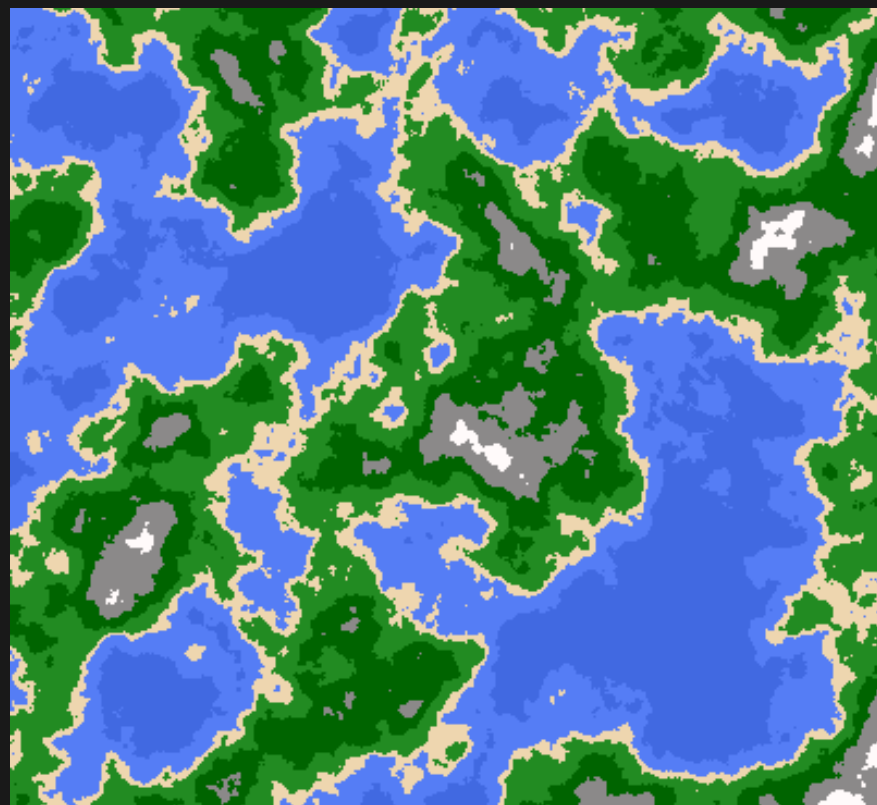
# Renderização de ruído 3D



# Desafio cumprido!

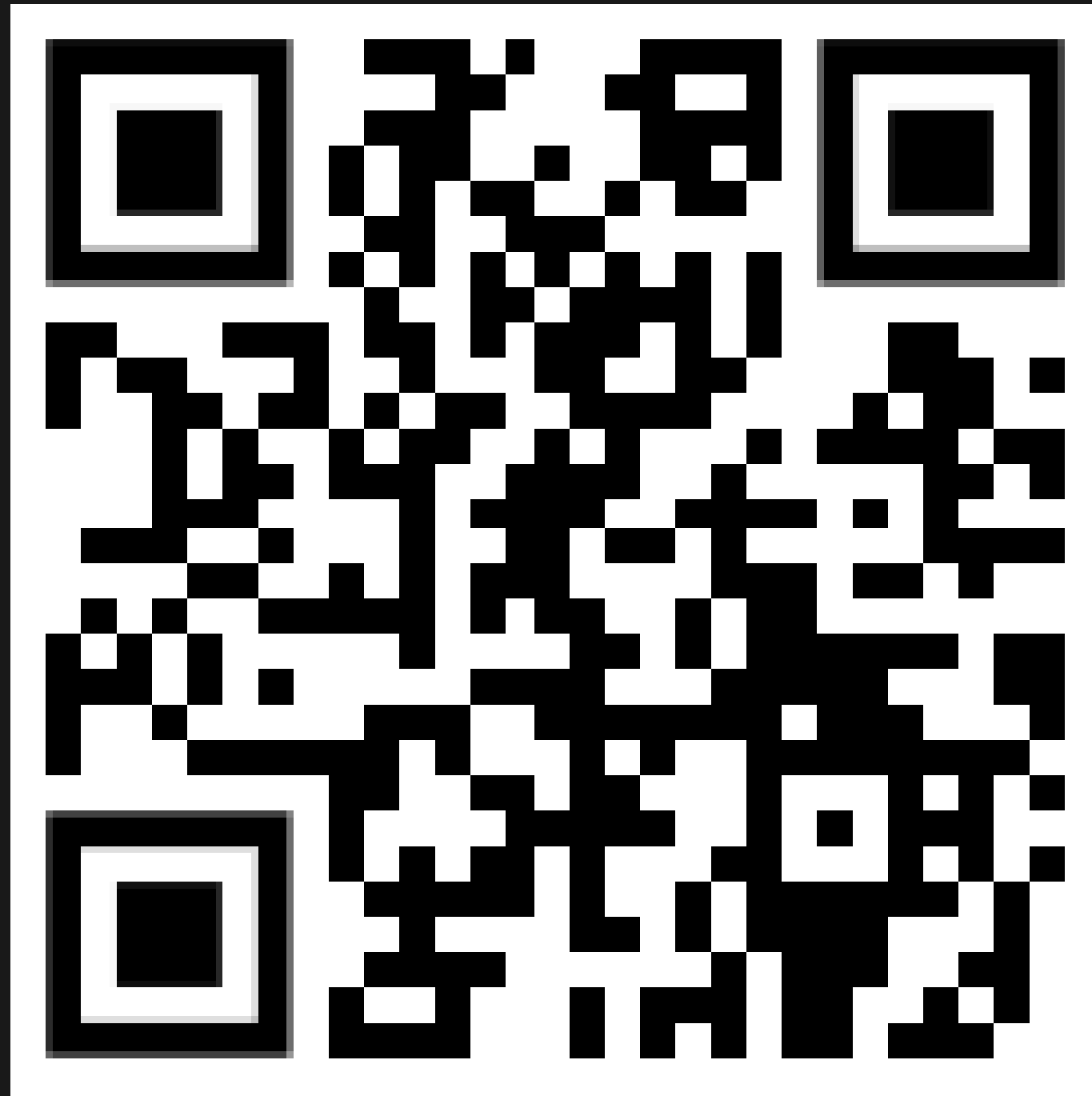


# Desafio pendente



Continua . . .

# Obrigado!



<https://github.com/AndersonPTSN>



<https://www.linkedin.com/in/andersonptsn>



[andersonptsn@yahoo.com.br](mailto:andersonptsn@yahoo.com.br)