# Práctica 4 Diagramas de Voronoi

Denisse Leyva

Marzo 10, 2021

## 1. Introducción

El tema de esta práctica tiene importancia en las matemáticas puras y en las ciencias aplicadas como por ejemplo la ciencia de materiales. Tomamos de un espacio bidimensional una zona con medidas conocidas que contiene $k$ puntos semillas $p_i$ representados por sus coordenadas $(x_i, y_i)$ lo que se busca es dividir esa zona en regiones llamadas celdas de Voronoi de tal forma que todos los puntos que pertenecen a la región de $p_i$ estén más cerca de esa semilla que de cualquier otra.
El modelo matemático en si es continuo, es decir, las coordenadas son números reales, pero nosotros lo vamos a discretizar en esta práctica. Se va a representar la zona por una matriz n $x$ n y las coordenadas serán entonces números enteros en $[1, n]$ [2].

## 2. Objetivo

Examinar de manera sistemática el efecto del número de semillas y del tamaño de la zona en la distribución de las grietas que se forman en términos de la mayor distancia manhattan entre la pieza y el exterior de la pieza [2].

## 3. Código

Con el siguiente código se obtiene la mayor distancia Manhattan entre la grieta y el exterior de la pieza y ser realizó un arreglo para que este dato lo tomará de diferentes tamaños de zona y con números de semilla proporcionales al tamaño de la zona. Se utilizaron 5 diferentes tamaños de zona (20x20, 40x40, 60x60, 80x80 y 100x100) y la cantidad de semillas se determinó por los porcentajes de longitud de zona (50 %, 75 %, 100 %, 125 %, 150 %). El código base se obtuvo de Schaeffer [3]. El código completo se encuentra en el repositorio [1].

```
1   while True:
2         g[x, y] = negro
3         largo += 1
4         frontera, interior = [], []
5         for v in vecinos:
6             (dx, dy) = v
7             vx, vy = x + dx, y + dy
8             if vx >= 0 and vx < n and vy >= 0 and vy < n: # existe
9                if g[vx, vy] != negro: # no tiene grieta por el momento
10                    if vor[vx, vy] == vor[x, y]: # misma celda
11                        interior.append(v)
12                    else:
13                        frontera.append(v)
```

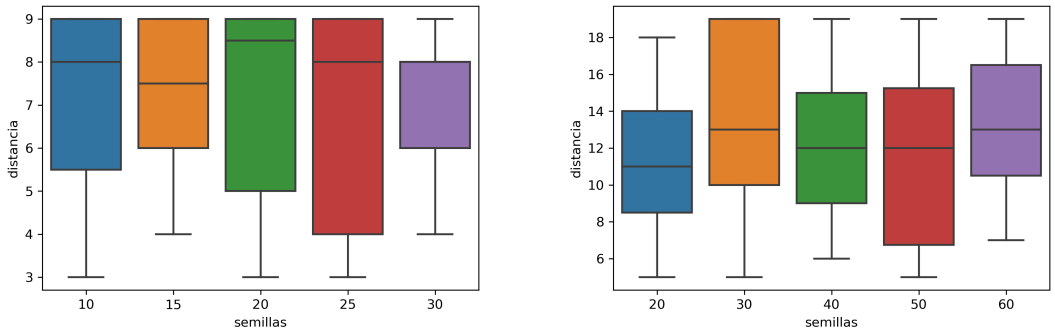
Código 1: Obtiene el mínimo − máximo de la distancia Manhattan de la grieta.

```
if __name__ == "__main__":
    replicas = 20
    for n in range(20, 101, 20):
        p_t = []
        porcentaje = []
        dimension = []
        for por in range(50, 151,25):

            profundidad = []
            k = int((por*n)/100)
            semillas =  []
            # semillas = []
            for s in range(k):
                while True:
                    x, y = randint(0, n - 1), randint(0, n - 1)
                    if (x, y) not in semillas:
                        semillas.append((x, y))
                        break
```

Código 2: Representa la automatización para variar el tamaño de la zona y el número de semillas que aparecen.

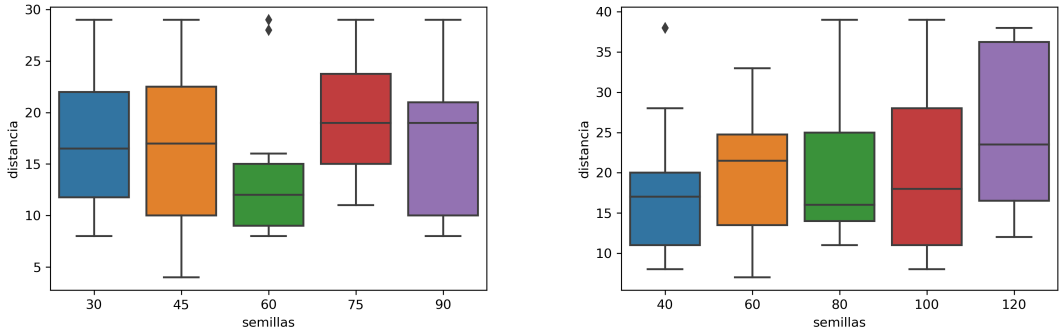
# 4. Resultados




(a) Matriz 20x20

(b) Matriz 40x40

Figura 1: Gráfica de distancia Manhattan vs número de semillas.




(a) Matriz 60x60 (b) Matriz 80x80

Figura 2: Gráfica de distancia Manhattan vs número de semillas.

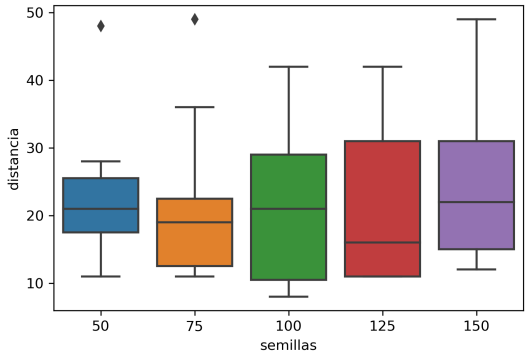
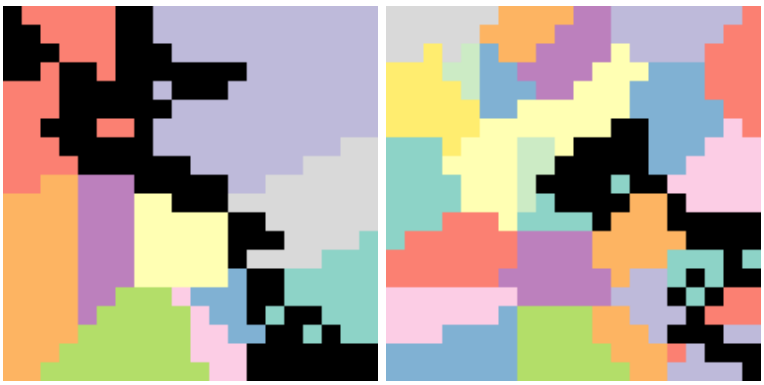
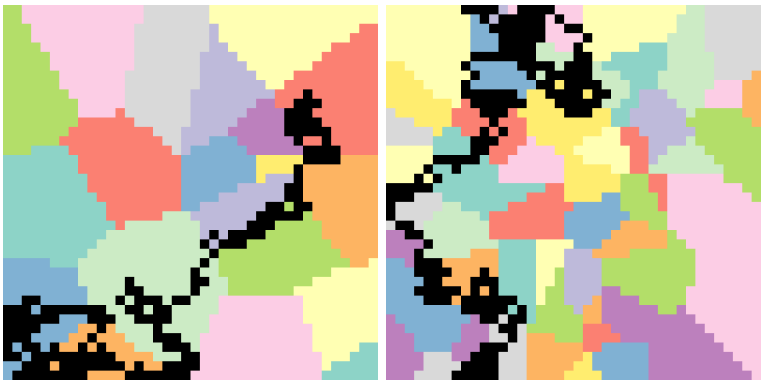



Figura 3: Matriz 100x100 distancia Manhattan vs numero de semillas.



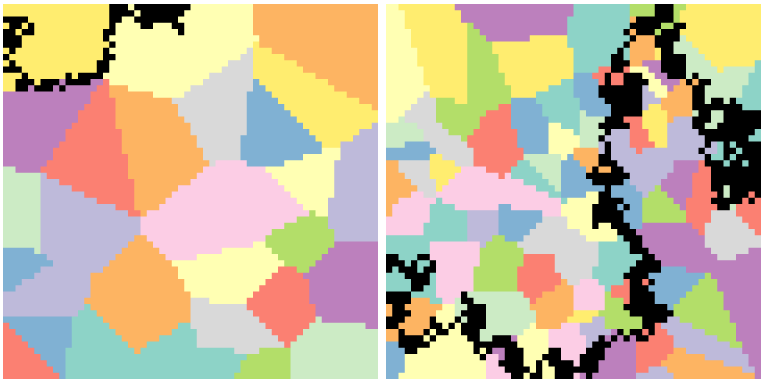
(a) Matriz 20x20 10 semillas (b) Matriz 20x20 30 semillas

Figura 4: Imágenes de la grieta con 10 y 30 semillas.



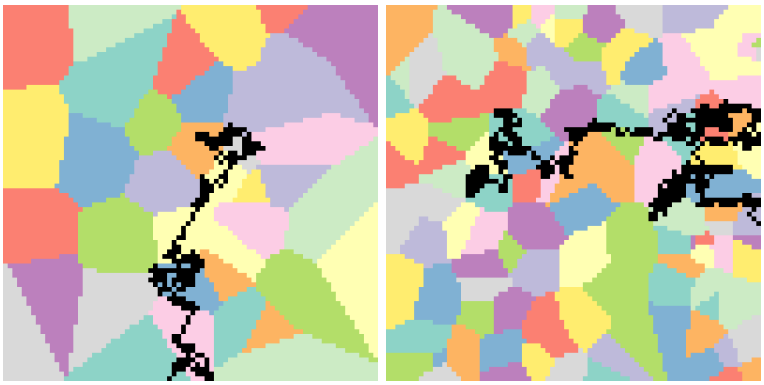
(a) Matriz 40x40 20 semillas (b) Matriz 40x40 60 semillas

Figura 5: Imágenes de la grieta con 20 y 60 semillas.



(a) Matriz 60x60 30 semillas (b) Matriz 60x60 90 semillas

Figura 6: Imágenes de la grieta con 30 y 90 semillas.



(a) Matriz 80x80 40 semillas (b) Matriz 80x80 120 semillas

Figura 7: Imágenes de la grieta con 40 y 120 semillas.

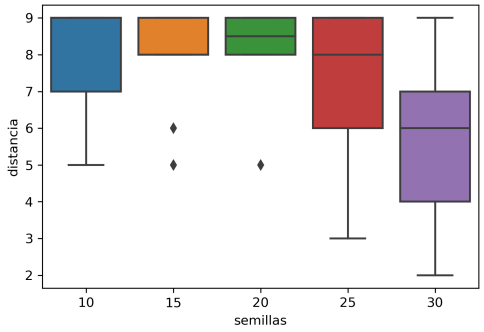
## 5. Reto 1

En este reto se deben hacer crecer las celdas dinámicamente alrededor de semillas de tal forma que las semillas aparezcan al azar en distintas iteraciones y crecen con una tasa exponencialmente distribuida (variable entre núcleos, pero constante para un núcleo específico) hasta toparse con las demás celdas [2].
El código completo se encuentra en el repositorio, así como el gif para poder observar el crecimiento de las semillas [1].

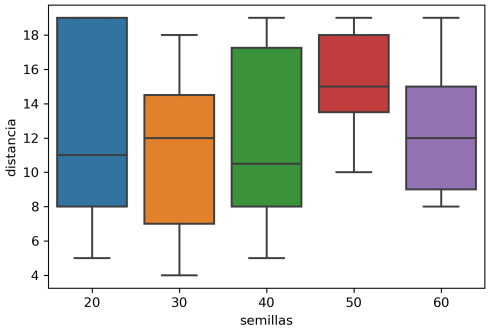
```
while True:
   ac = []
   for s in range(len(semillas)):
       for pos in range(n*n):
           fil = pos // n
           col = pos % n
           if celda[fil, col] == fondo:
               ac.append(1)
           else:
               ac.append(0)
           (xs, ys) = semillas[s]
           dx, dy = fil - xs, col - ys
           dist = sqrt(dx**2 + dy**2)
           if dist <= radio[s]:
               if celda[fil, col] == fondo:
                   celda[fil, col] = ImageColor.getrgb(color[sc[s]])
       radio[s] += 1

   if semilla <= k-1:
       conteo = 0
       while True:
           if ac == []:
               ac.append(1)
           if len(sc) == k:
               break
           columna = randint(0, n - 1)
           fila = randint(0, n - 1)
           if celda[columna, fila] == fondo:
               celda[columna, fila] =  ImageColor.getrgb(color[semilla])
               sc.append(semilla)
               semillas.append((columna,fila))
               break
           if conteo == 10:
               break
           conteo += 1
   visual = zona.resize((10 * n,  10 * n), resample=Image.BOX)
   # visual.save("p4p_{:d}.png".format(ciclo))
   visual.save("p4p_{1:d}_{2:d}_{0:d}.png".format(ciclo, n, k))
   ciclo += 1
   semilla += 1
   acc = max(ac)
   if acc == 0:
       print('no hay fondo')
       break
```

Código 3: Representa la aparición de la semilla en diferentes iteraciones en forma radial.

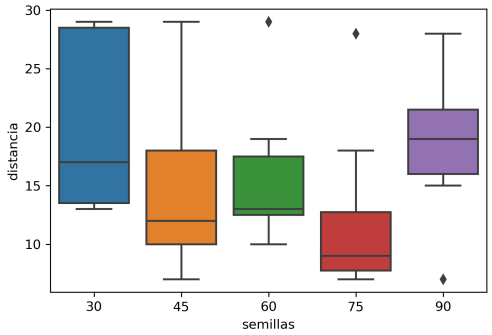



(a) Matriz 20x20

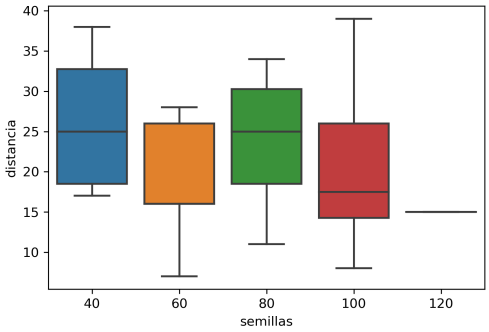



(b) Matriz 40x40

Figura 8: Gráfica de distancia Manhattan vs número de semillas.




(a) Matriz 20x20




(b) Matriz 40x40

Figura 9: Gráfica de distancia Manhattan vs número de semillas.

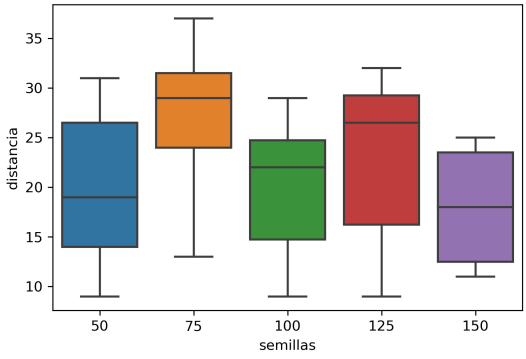
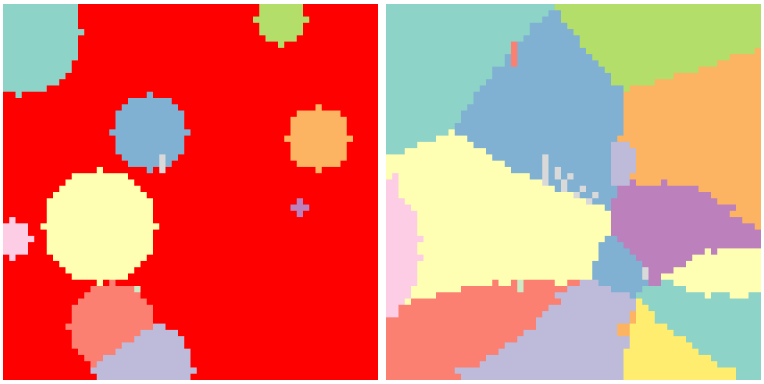


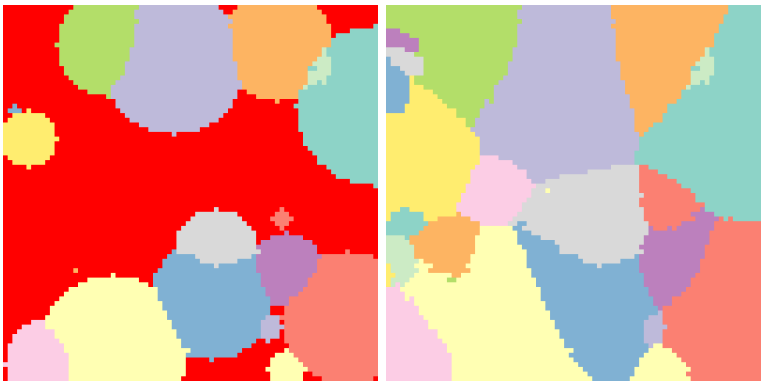

Figura 10: Matriz 100x100 de distancia Manhattan vs número de semillas.



(a) Matriz 60x60 60 semillas iteración 10

(b) Matriz 60x60 60 semillas iteración 26

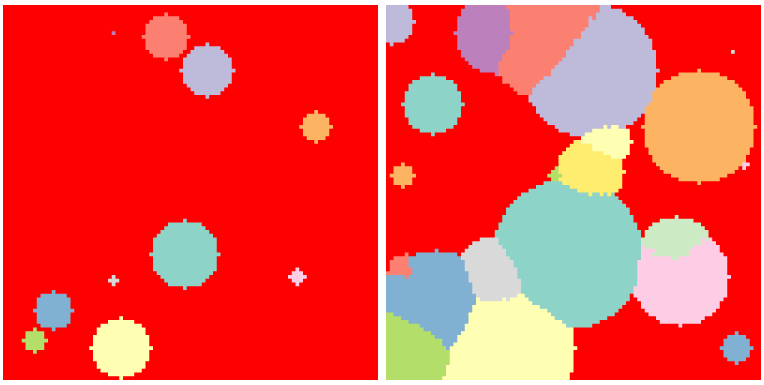
Figura 11: Imágenes del comportamiento con 60 semillas con diferentes iteraciones.



(a) Matriz 80x80 80 semillas iteración 17

(b) Matriz 80x80 80 semillas iteración 31

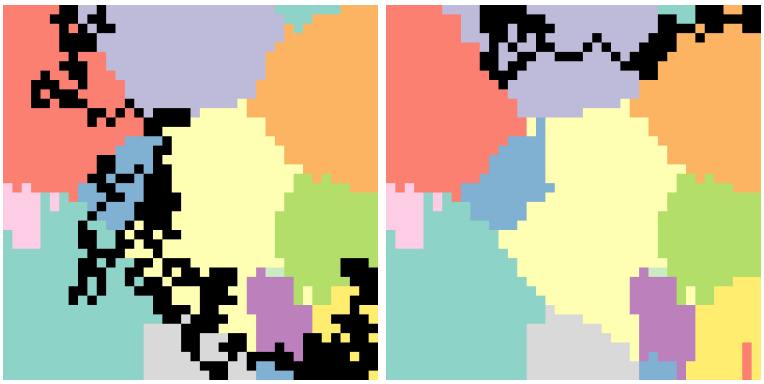
Figura 12: Imágenes del comportamiento con 80 semillas con diferentes iteraciones.



(a) Matriz 100x100 100 semillas iteración 9

(b) Matriz 100x100 100 semillas iteración 20

Figura 13: Imágenes del comportamiento con 100 semillas con diferentes iteraciones.



(a) Matriz 40x40 40 semillas con grieta.

(b) Matriz 40x40 40 semillas con grieta.

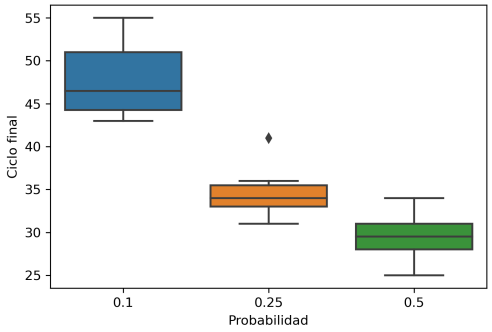
Figura 14: Imágenes del comportamiento de la grieta con 40 semillas.

## 6. Reto 2

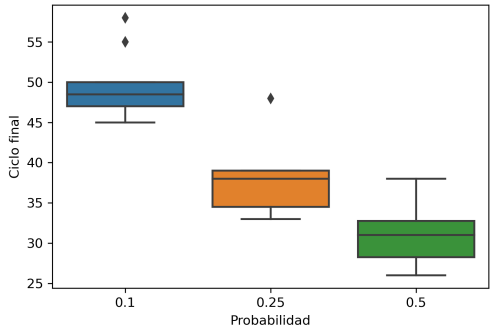
Para este segundo reto el crecimiento ya no es determinista, sino probabilista, como para modelar fenómenos como cáncer: un núcleo propaga a vecinos desocupados con una probabilidad $p_v$ si un núcleo no logra crecer, se muere con una probabilidad $p_m$ [2].
El código completo se encuentra en el repositorio, así como el gif para poder observar el crecimiento de las semillas [1].

```
for s in range(len(semillas)):
    for pos in range(n*n):
        fil = pos //  n
        col = pos % n
        if celda[fil, col] == fondo:
            ac.append(1)
        else:
            ac.append(0)
        if vida[s] == 1:
            cic[s] = 1
            (xs, ys) = semillas[s]
            dx, dy = fil - xs, col - ys
            dist = sqrt(dx*2 + dy*2)
            if dist <= radio[s]:
                if celda[fil, col] == fondo:
                    if (random() < pv):
                        vecinos[s] += 1
                        celda[fil, col] = ImageColor.getrgb(color[sc[s]])
    v = (vecinos[s])
    radio[s] += 1
    vc = cic[s]
    if v <= 1 and cic[s] == 1 and random() < pm:
        vida[s] = 0
        cic[s] = 2
```

Código 4: Determina el crecimiento probabilístico de la semilla.




(a) Matriz 50x50 15 semillas 25 % de que si una semilla no crece muere.




(b) Matriz 50x50 15 semillas 50 % de que si una semilla no crece muere.

Figura 15: Gráficas de probabilidad de crecimiento de semillas.

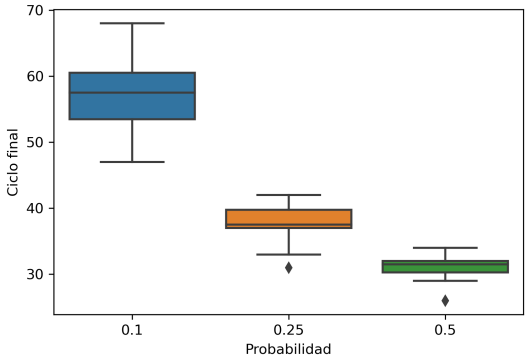
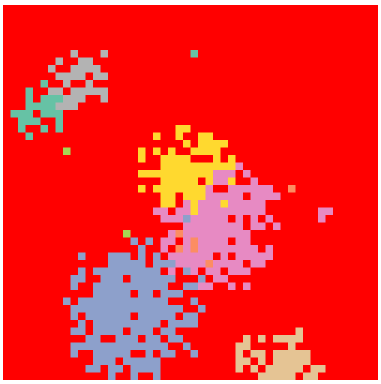
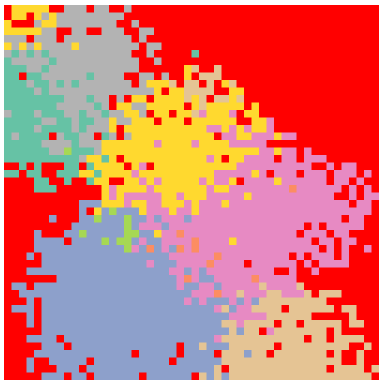



Figura 16: Gráfica de probabilidad de Matriz 50x50 15 semillas con 100 % de probabilidad de que si una semilla no crece muere .

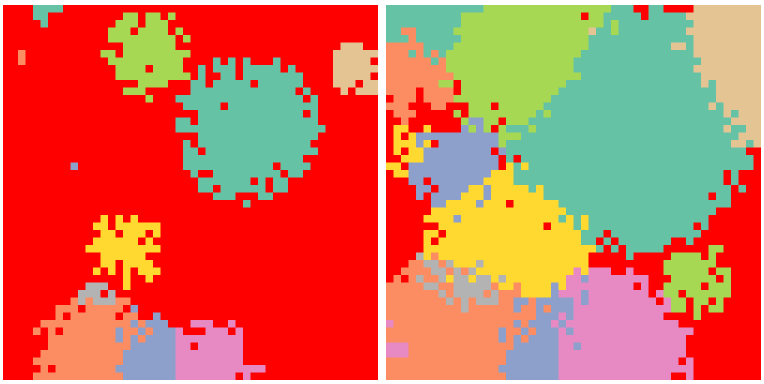


(a) Matriz 50x50 15 semillas con 25 % probabilidad de crecimiento.



(b) Matriz 50x50 15 semillas con 25 % probabilidad de crecimiento.

Figura 17: Imágenes del comportamiento con 15 semillas en diferentes ciclos.



(a) Matriz 50x50 15 semillas con 50 % probabilidad de crecimiento

(b) Matriz 50x50 15 semillas con 50 % probabilidad de crecimiento

Figura 18: Imágenes del comportamiento con 15 semillas en diferentes ciclos.

# Referencias


[1] *Denisse Leyva Repositorio*. URL: `https://github.com/Denisse251/Simulation/tree/main/Tarea.4`.

[2] *Elisa Schaeffer Práctica 4*. URL: `https://elisa.dyndns-web.com/teaching/comp/par/p4.html`.

[3] *Elisa Schaeffer Repositorio*. URL: `https://github.com/satuelisa/Simulation`.