

Automatos celulares

Nome: Arthur Pontes Nader

Matrícula: 2019022294

Bibliotecas

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
```

Estados possíveis

0 - Suscetível - Amarelo

1 - Infectada - Vermelho

2 - Recuperada - Verde

Funções

```
In [2]: def criar_rede(L):

    N = L*L

    rede = np.zeros(N, dtype = np.int32)

    x = np.random.randint(N)

    rede[x] = 1

    return rede
```

```
In [3]: def gerar_vizinhos(L):

    N = L*L
    vizinhos = np.zeros((N, 4), dtype = np.int32)

    for k in range(N):

        vizinhos[k][0] = k + 1
        if((k+1)%L) == 0:
            vizinhos[k][0] = k + 1 - L

        vizinhos[k][1] = k + L
        if(k > N - 1 - L):
            vizinhos[k][1] = k + L - N

        vizinhos[k][2] = k - 1
        if(k%L == 0):
            vizinhos[k][2] = k + L - 1

        vizinhos[k][3] = k - L
        if(k < L):
```

```
vizinhos[k][3] = k + N - L

return vizinhos
```

```
In [4]: def mostrar_rede(rede, iteracao = "0"):

    L = int(np.sqrt(len(rede)))
    rede = np.reshape(rede, (L,L))

    plt.figure(figsize=(8, 8))

    for i in range(L):
        for j in range(L):
            if rede[i, j] == 2:
                plt.scatter(i,j,s=200, c='lime', marker = 's')
            elif rede[i, j] == 1:
                plt.scatter(i,j,s=200, c='tomato', marker = 's')
            else:
                plt.scatter(i,j,s=200, c='gold', marker = 's')

    if iteracao == "0":
        plt.title("Configuração Inicial\n", fontsize = 18)
    else:
        plt.title("Iteração " + iteracao + "\n", fontsize = 18)

    plt.axis('off')
    plt.plot()
```

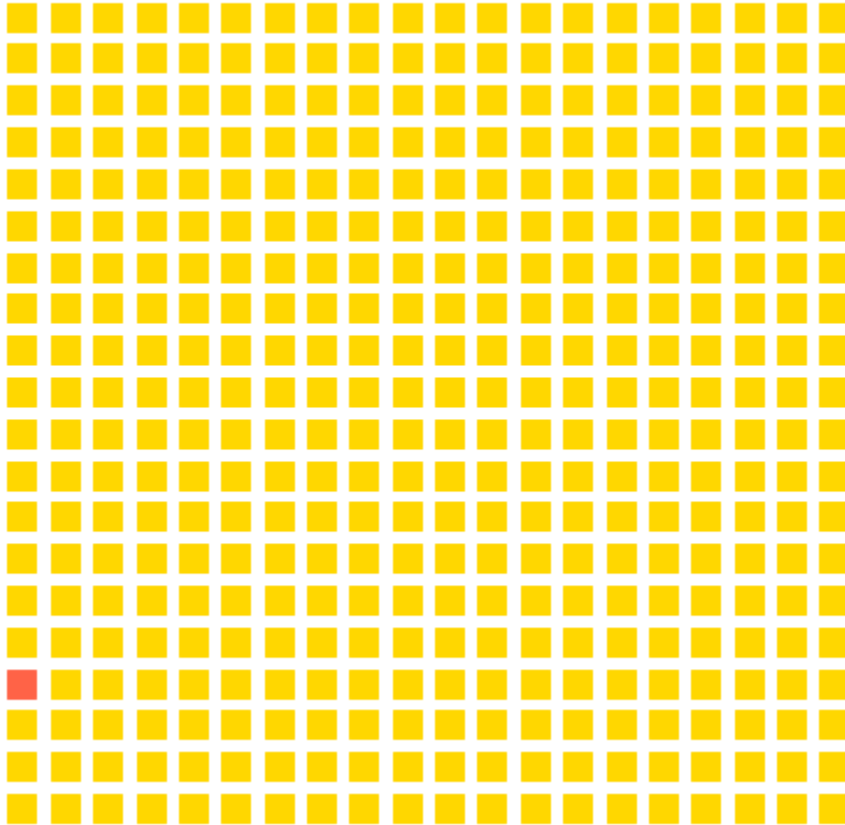
```
In [5]: def contar_casos_na_rede(rede):

    suscetiveis = list(rede).count(0)
    infectados = list(rede).count(1)
    recuperados = list(rede).count(2)

    return [suscetiveis, infectados, recuperados]
```

```
In [6]: r = criar_rede(20)
mostrar_rede(r)
```

Configuração Inicial



```
In [7]: contar_casos_na_rede(r)
```

```
Out[7]: [399, 1, 0]
```

Funções para evolução temporal

```
In [8]: def atualizar_rede(rede, vizinhos, prob_contaminacao, prob_cura):  
  
    N = len(rede)  
    nova_configuracao = np.copy(rede)  
  
    for i in range(N):  
  
        if rede[i] == 0:  
            for k in range(4):  
                if rede[vizinhos[i,k]] == 1 and np.random.random() < prob_contaminacao:  
                    nova_configuracao[i] = 1  
  
        elif rede[i] == 1:  
            if np.random.random() < prob_cura:  
                nova_configuracao[i] = 2  
  
    return nova_configuracao
```

```
In [9]: def propagar_pela_rede(L, iteracoes, prob_contaminacao, prob_cura):  
  
    configuracao_atual = criar_rede(L)  
    vizinhos = gerar_vizinhos(L)  
  
    evolucao_dos_casos = np.array([contar_casos_na_rede(configuracao_atual)])  
    evolucao_da_rede = [configuracao_atual]
```

```

for i in range(iteracoes):

    configuracao_atual = atualizar_rede(configuracao_atual, vizinhos, probab_aminacao,

    evolucao_da_rede.append(configuracao_atual)
    evolucao_dos_casos = np.vstack([evolucao_dos_casos, contar_casos_na_rede(configuraca

return evolucao_dos_casos, evolucao_da_rede

```

```

In [10]: def exibir_evolucao_dos_estados(dados, num_iteracoes):

    x = np.arange(num_iteracoes+1)

    plt.figure(figsize=(16, 10))
    plt.xlabel("Iteração", fontsize = 16)
    plt.ylabel("Indivíduos com estado da doença", fontsize = 16)

    labels = 'Indivíduos suscetíveis', 'Indivíduos infectados', 'Indivíduos recuperados'
    cores = "gold", "tomato", "lime"

    for i in range(3):
        plt.plot(x, medias[:, i], label=labels[i], c=cores[i])

    plt.title("Evolução temporal dos estados dos indivíduos\n", fontsize = 24)
    plt.legend(fontsize = 16)
    plt.show()

```

Resultados de diferentes configurações

Probabilidade de contaminação = 0.65 e Probabilidade de recuperação = 0.25

```

In [20]: casos, redes = propagar_pela_rede(20, 12, 0.65, 0.25)
        numero_de_casos = np.array([casos])

        for _ in range(9):

            casos, _ = propagar_pela_rede(20, 12, 0.65, 0.25)
            numero_de_casos = np.vstack([numero_de_casos, [casos]])

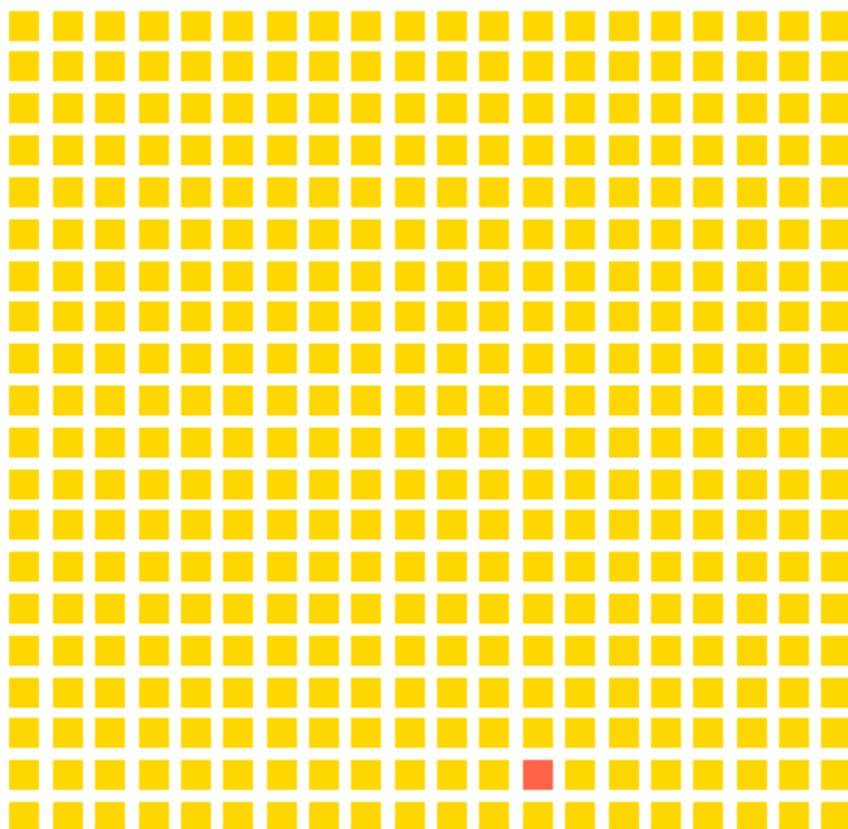
```

```

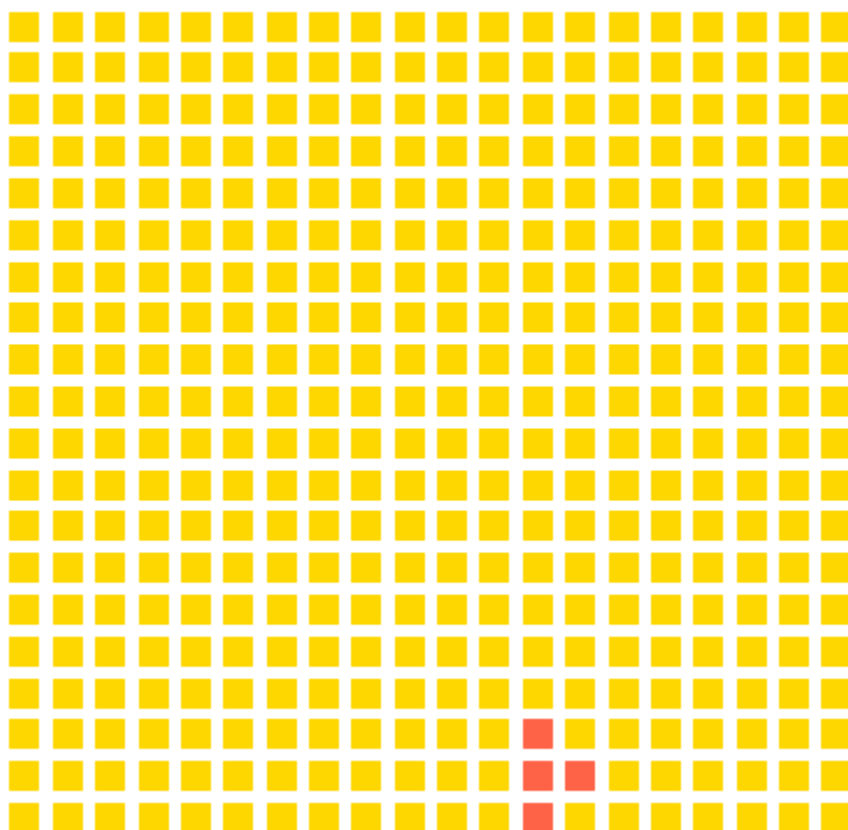
In [21]: mostrar_rede(redes)

```

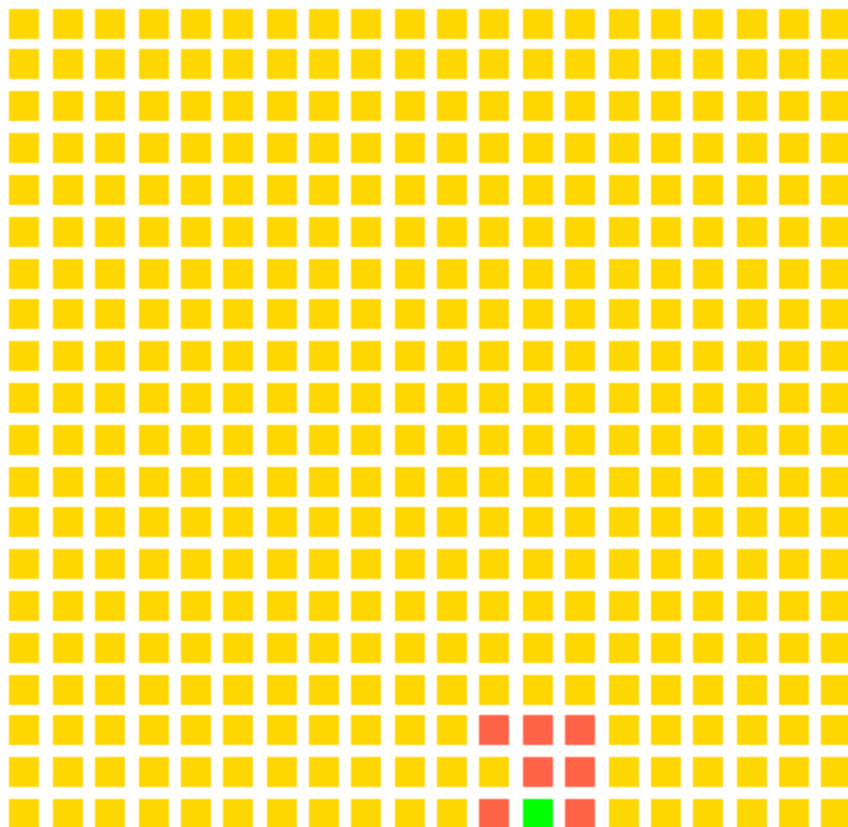
Configuração Inicial



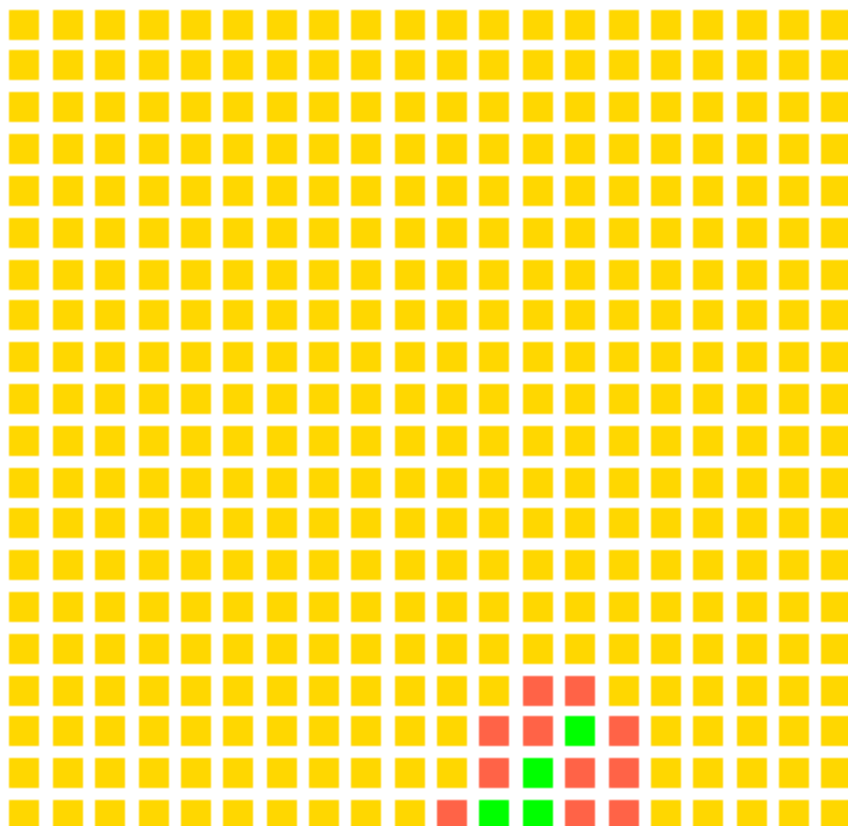
Iteração 1



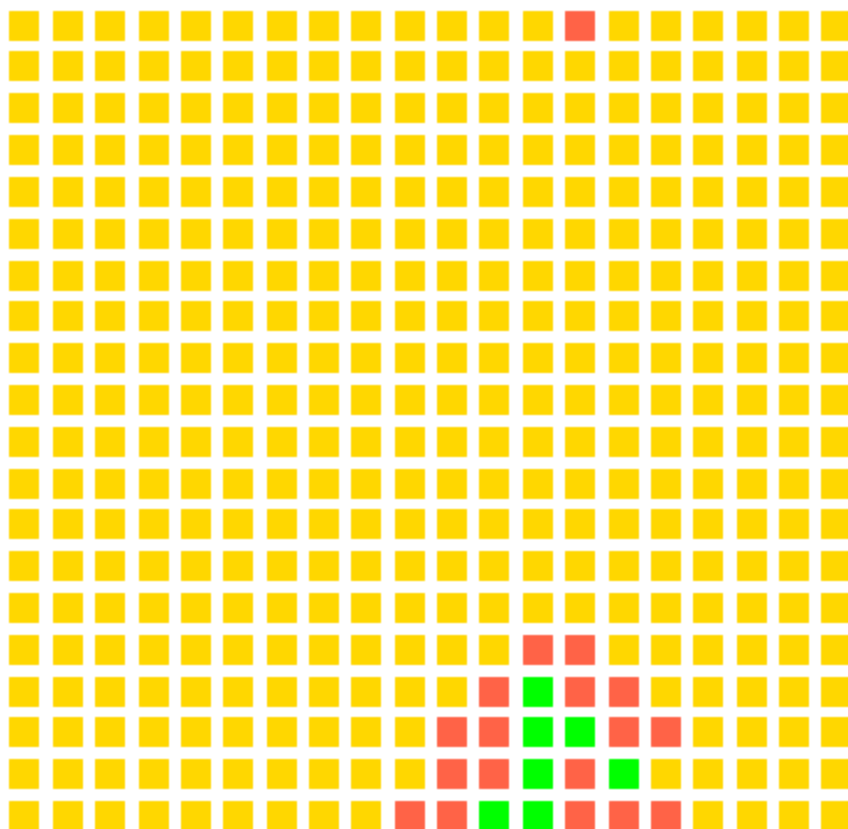
Iteração 2



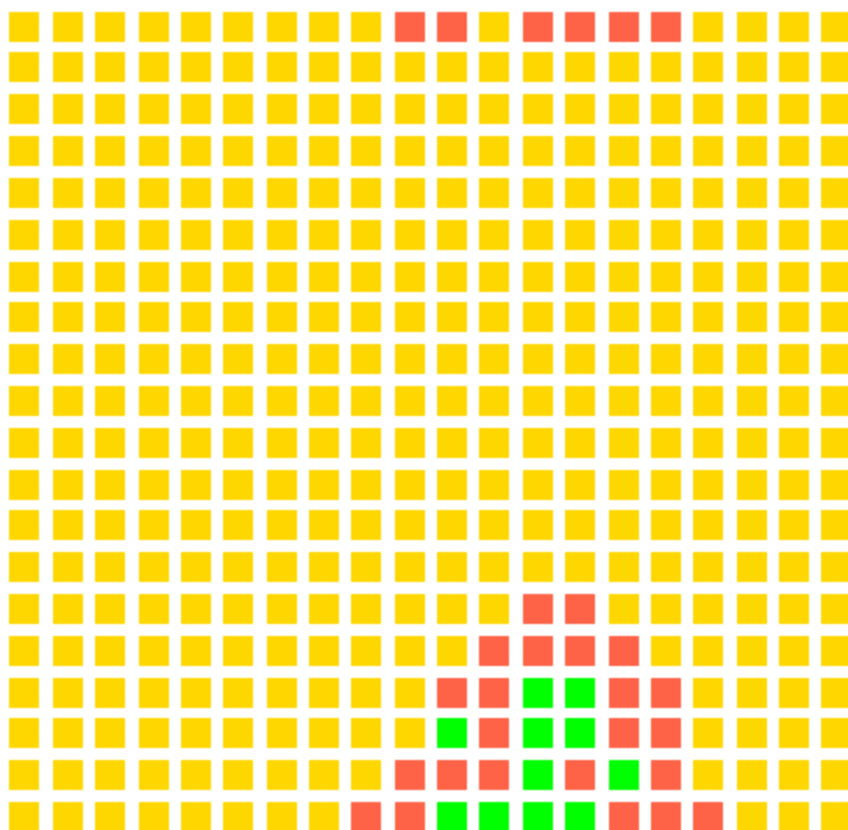
Iteração 3



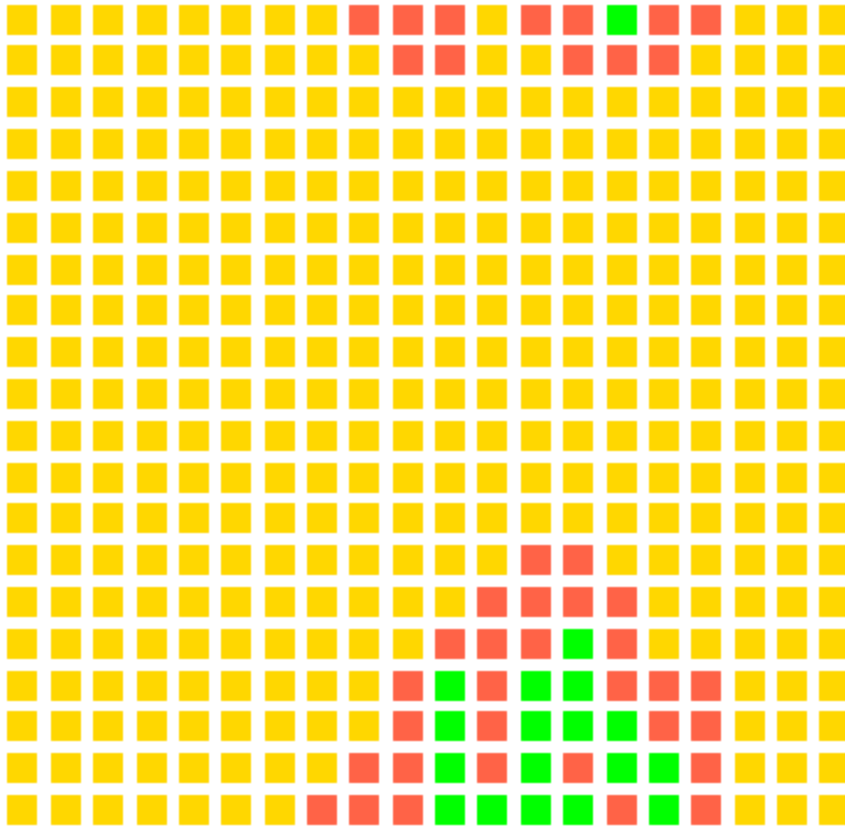
Iteração 4



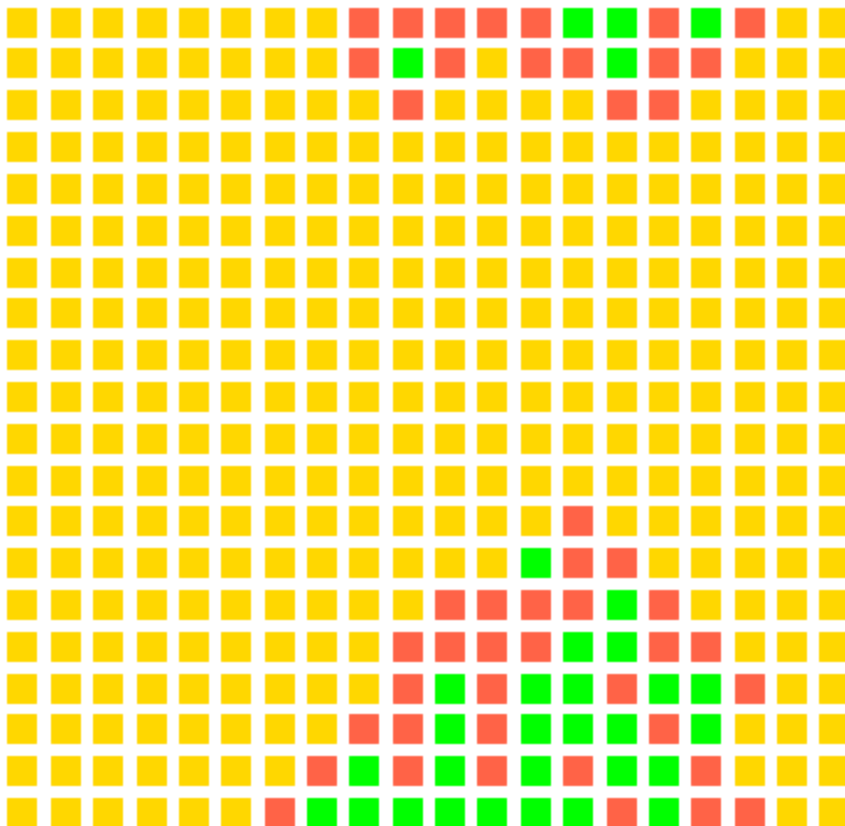
Iteração 5



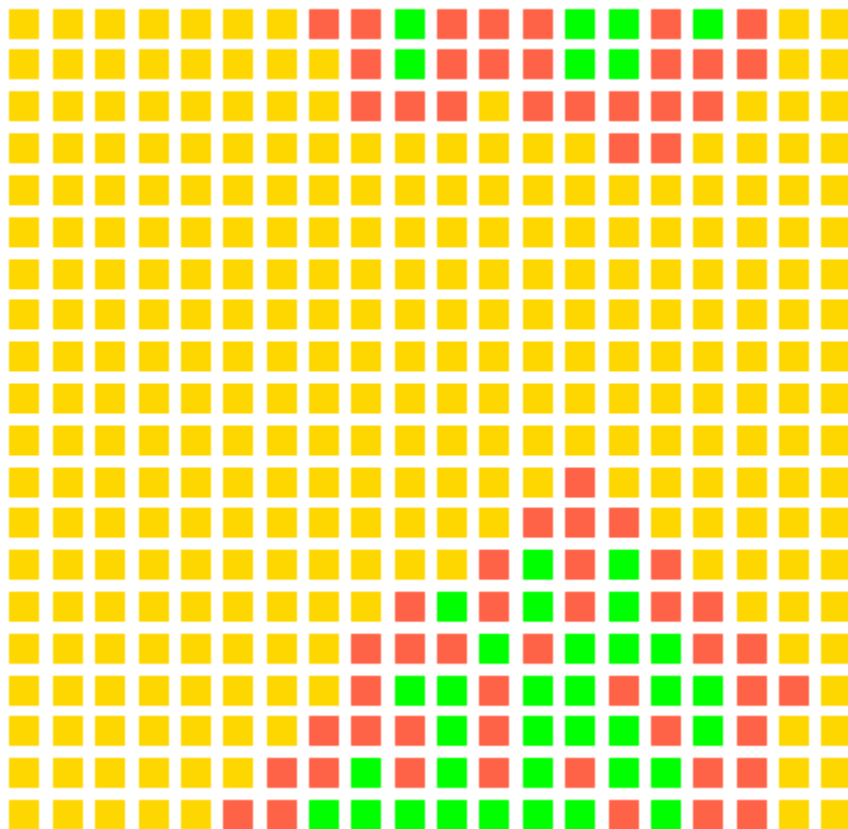
Iteração 6



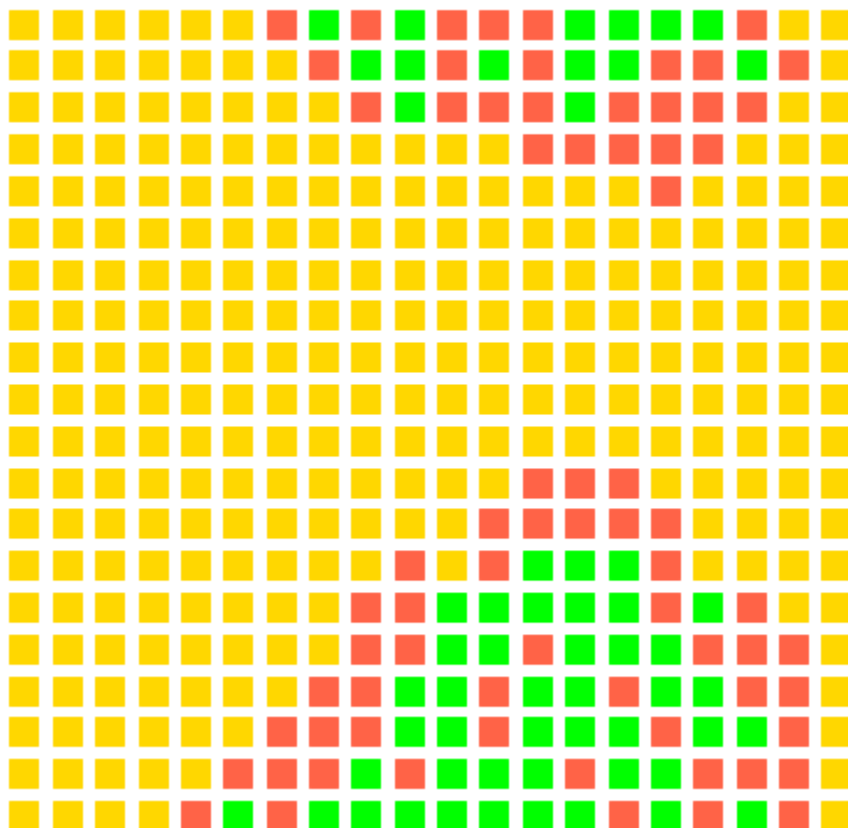
Iteração 7



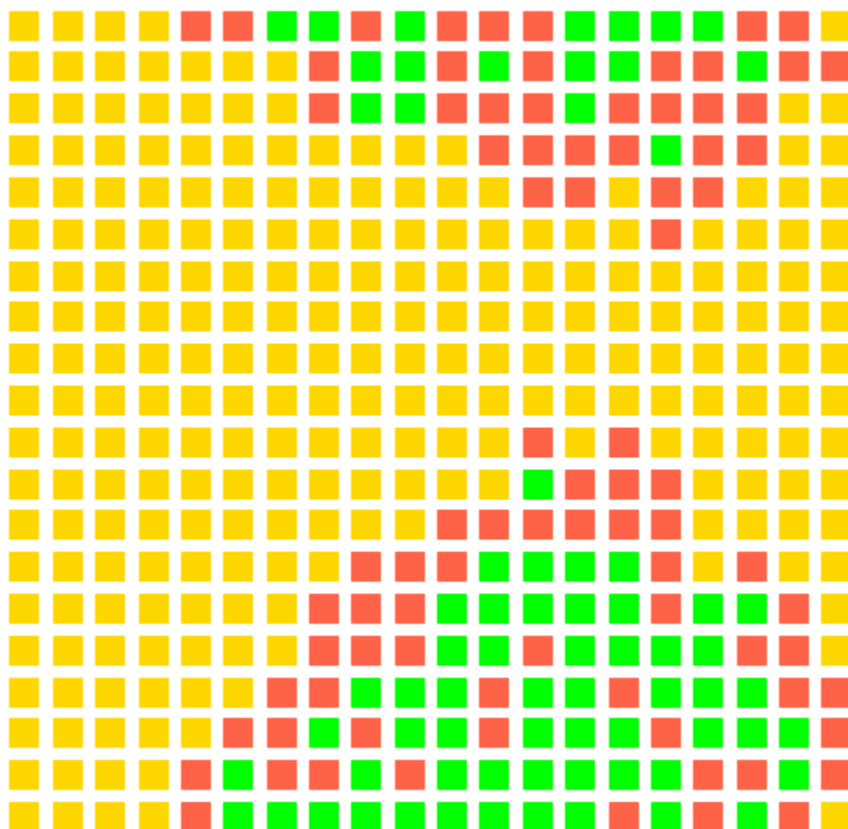
Iteração 8



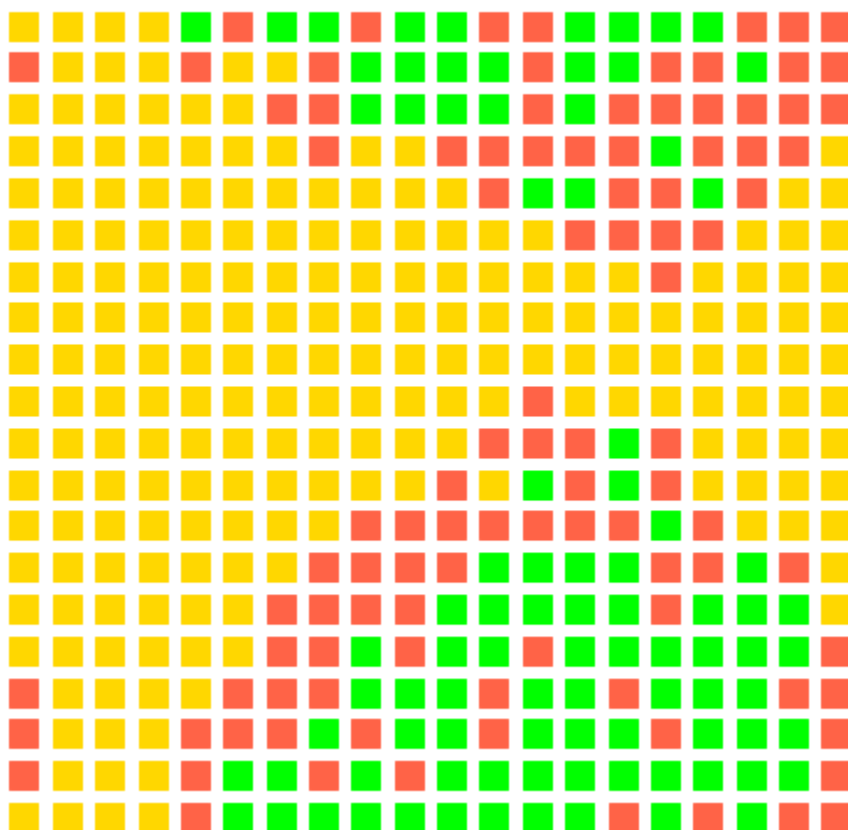
Iteração 9



Iteração 10



Iteração 11

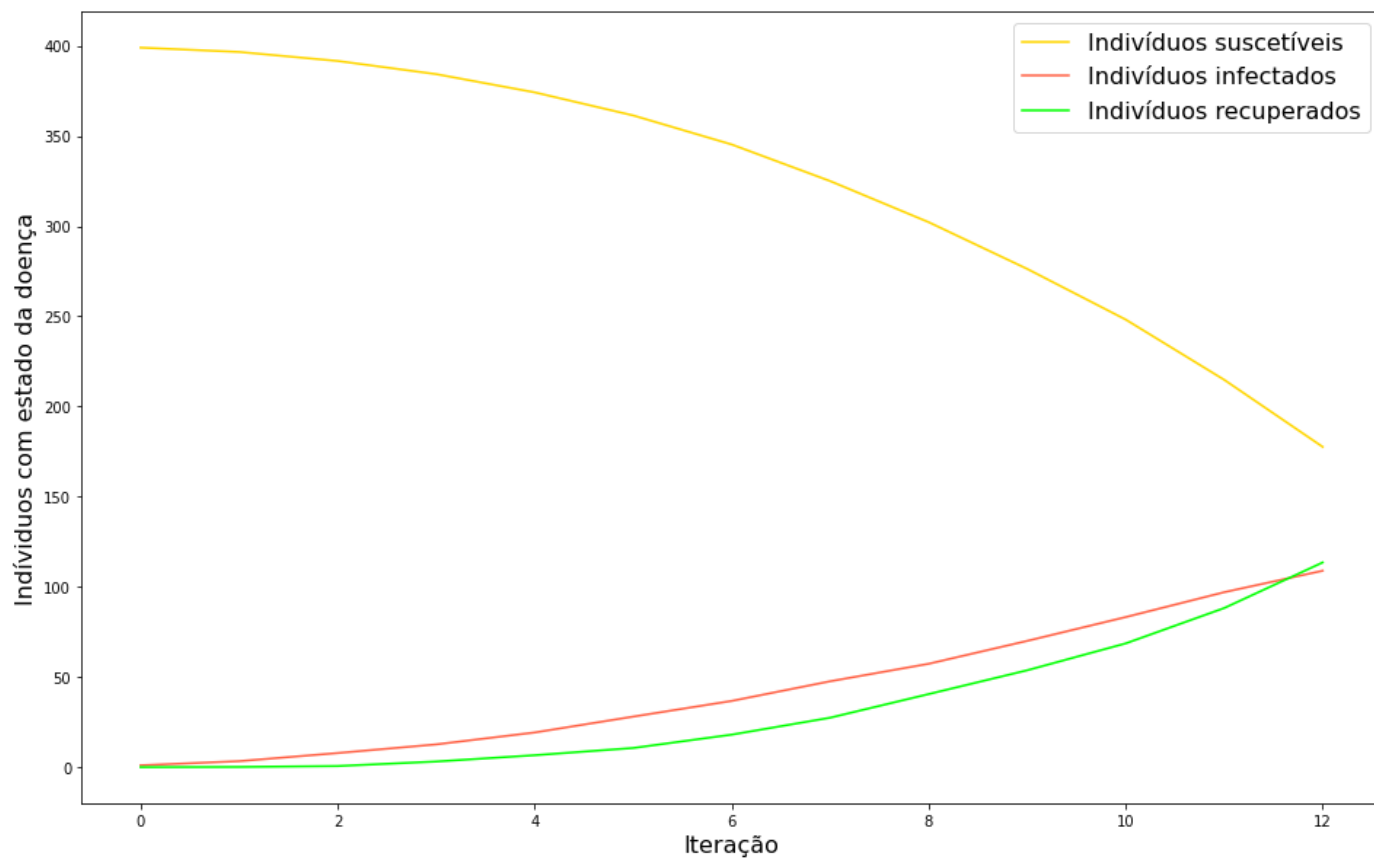


Iteração 12



```
In [22]: medias = np.mean(numero_de_casos, axis = 0)
         exibir_evolucao_dos_estados(medias, 12)
```

Evolução temporal dos estados dos indivíduos



Probabilidade de contaminação = 0.9 e Probabilidade de recuperação = 0.65

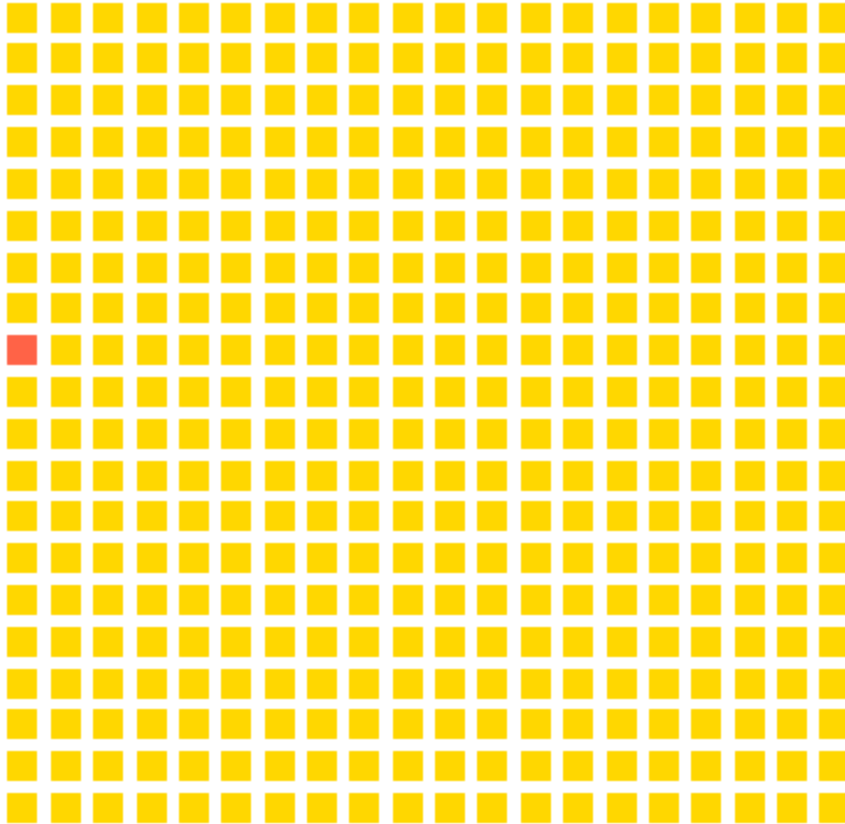
```
In [44]: casos, redes = propagar_pela_rede(20, 12, 0.9, 0.65)
numero_de_casos = np.array([casos])

for _ in range(9):

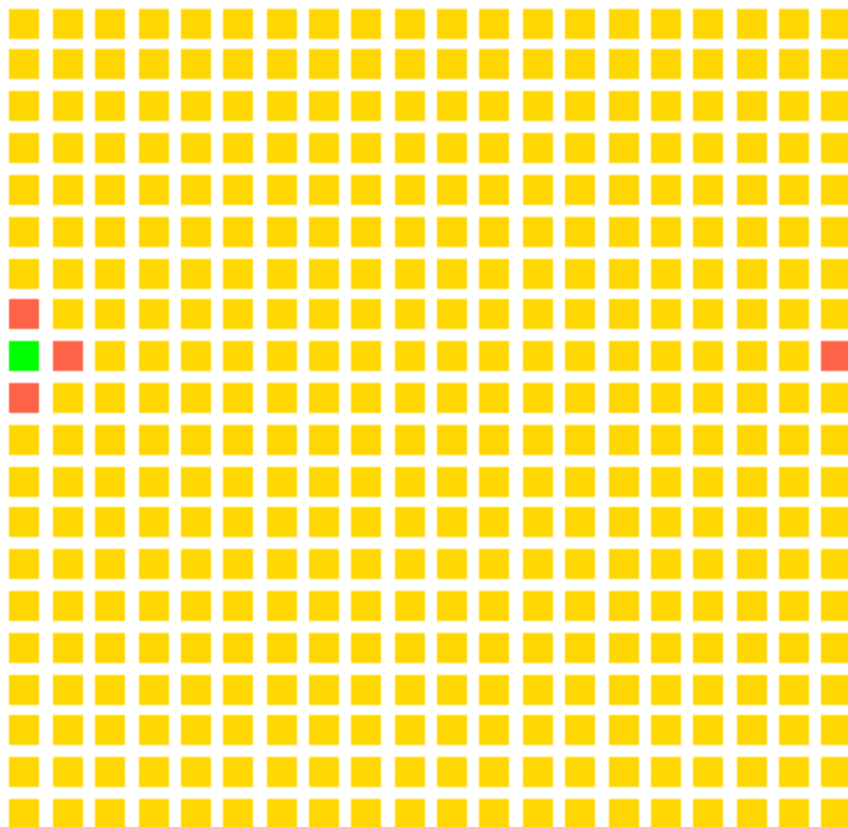
    casos, _ = propagar_pela_rede(20, 12, 0.9, 0.65)
    numero_de_casos = np.vstack([numero_de_casos, [casos]])

In [45]: for i in range(len(redes)):
    mostrar_rede(redes[i], str(i))
```

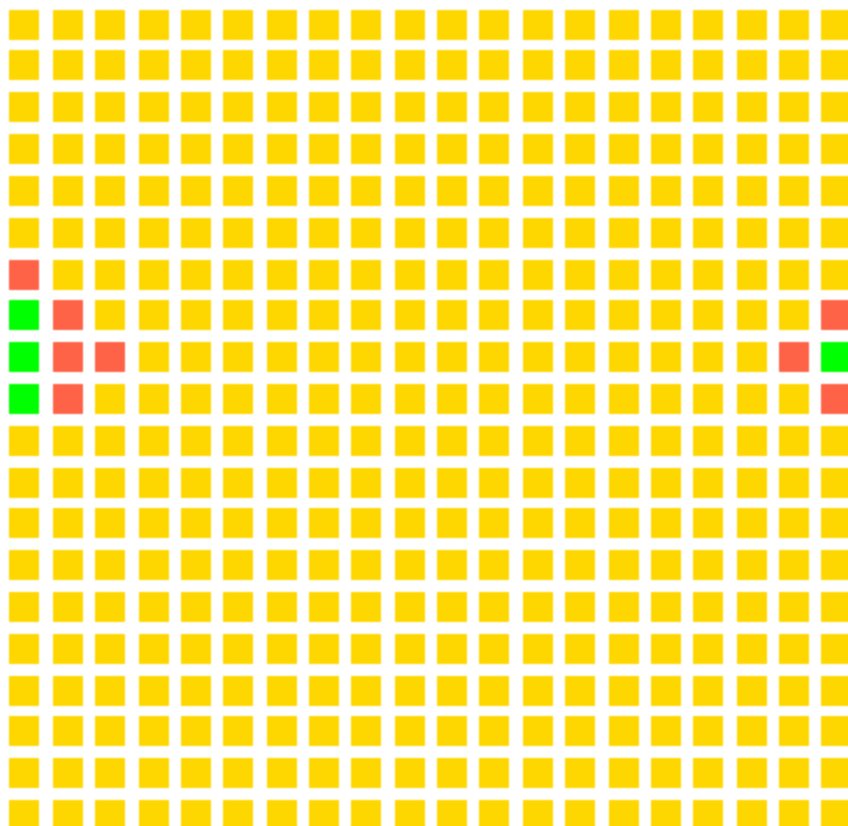
Configuração Inicial



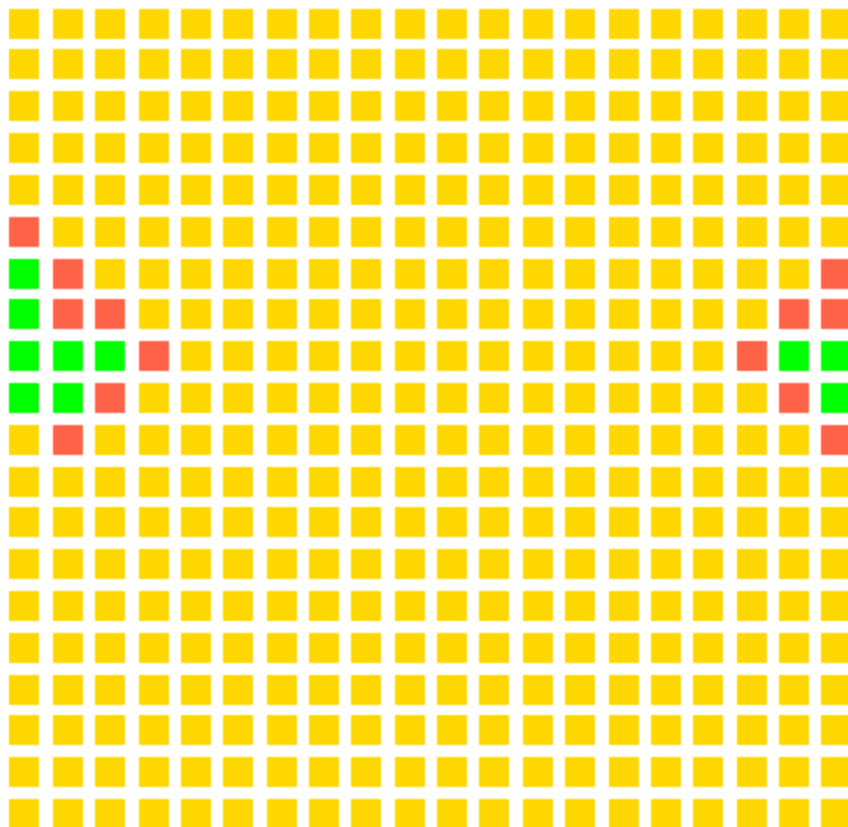
Iteração 1



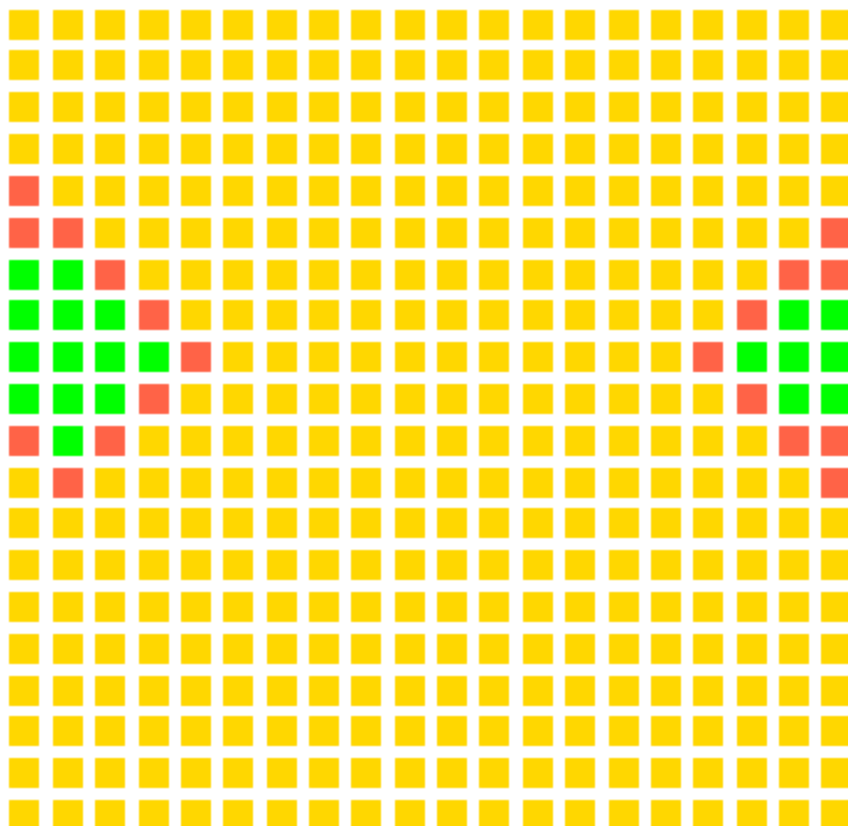
Iteração 2



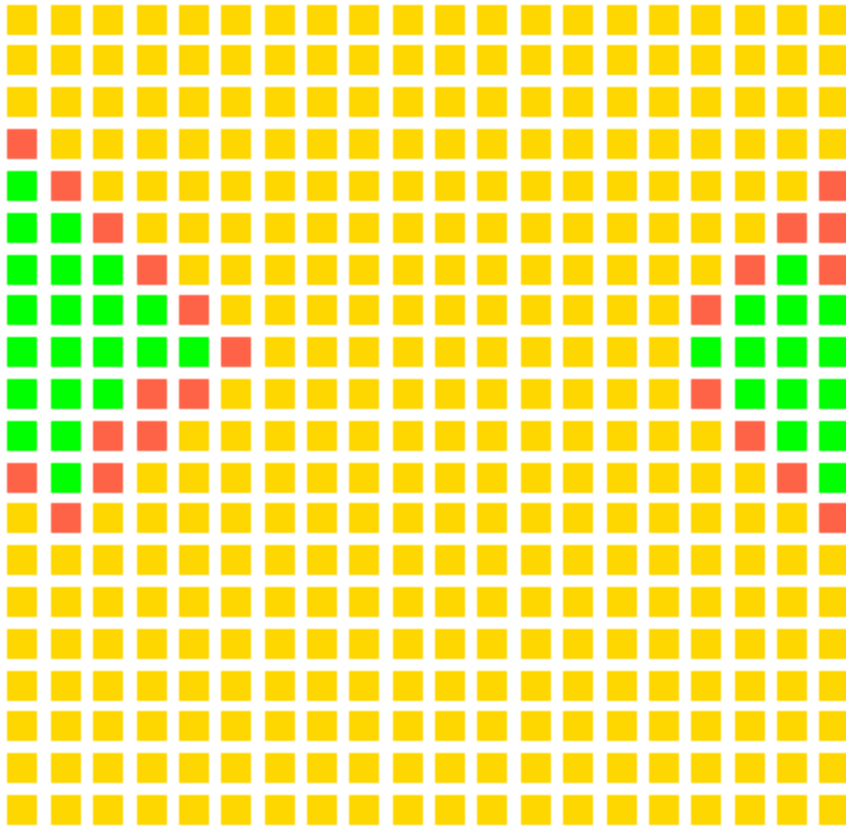
Iteração 3



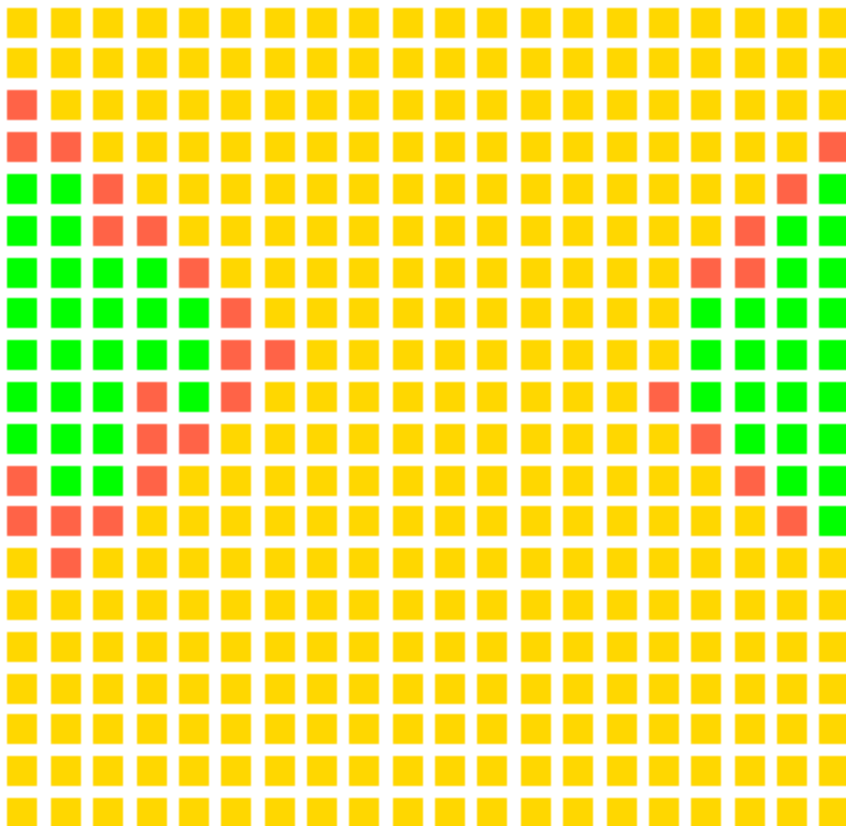
Iteração 4



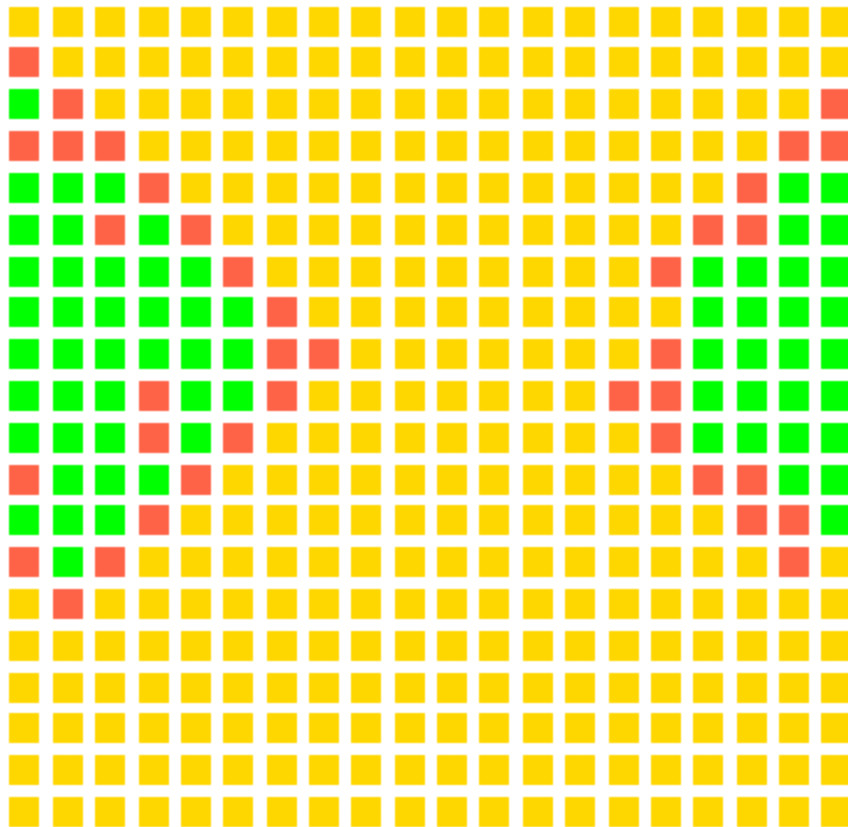
Iteração 5



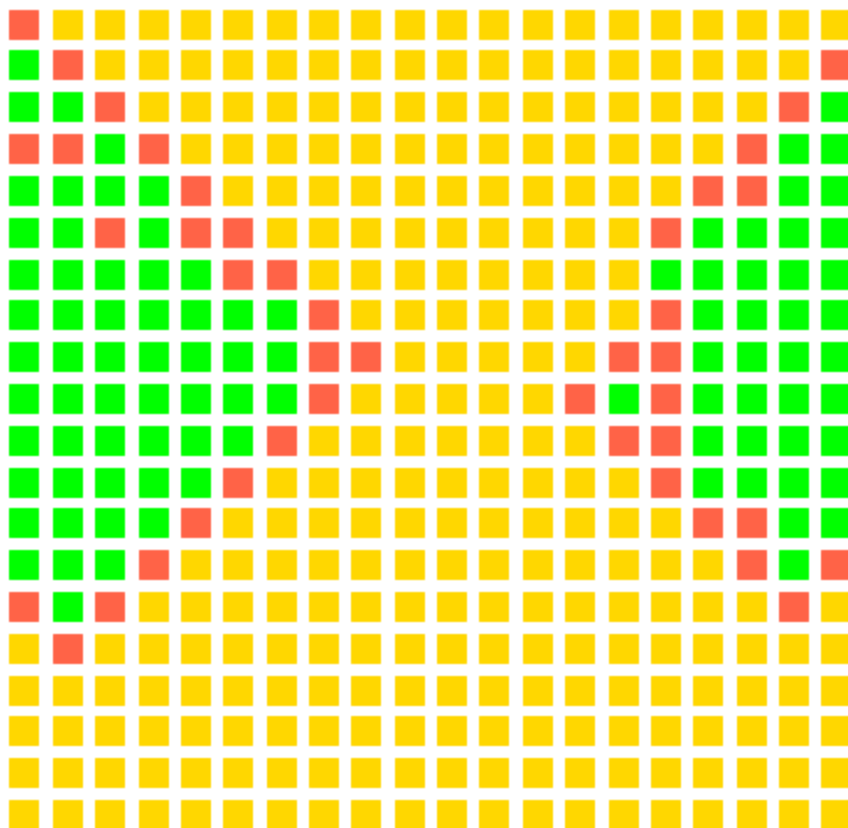
Iteração 6



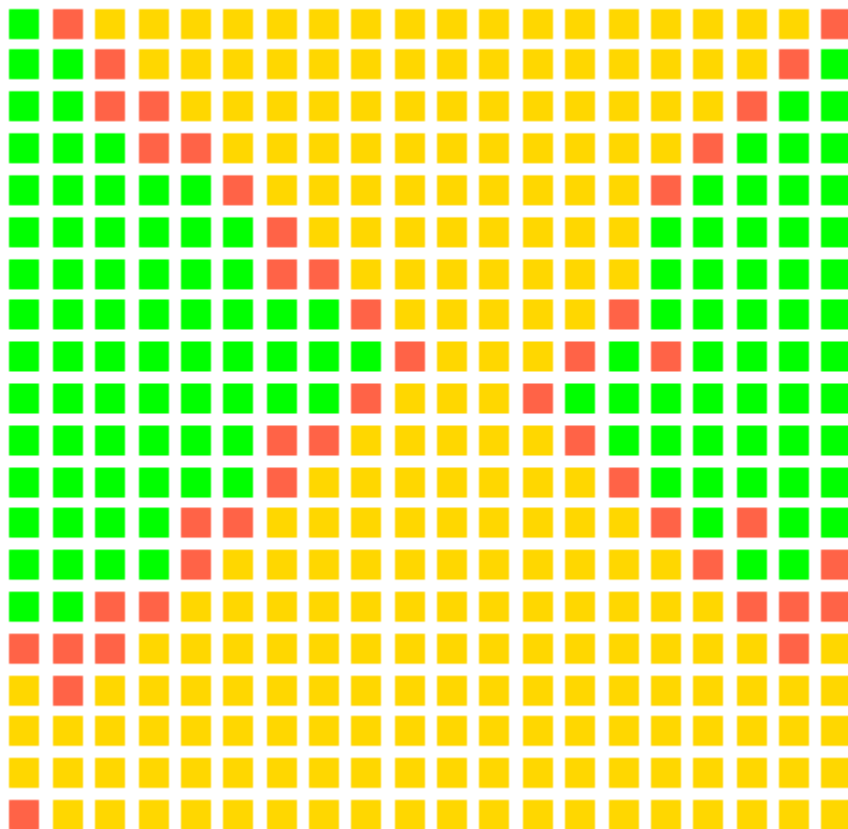
Iteração 7



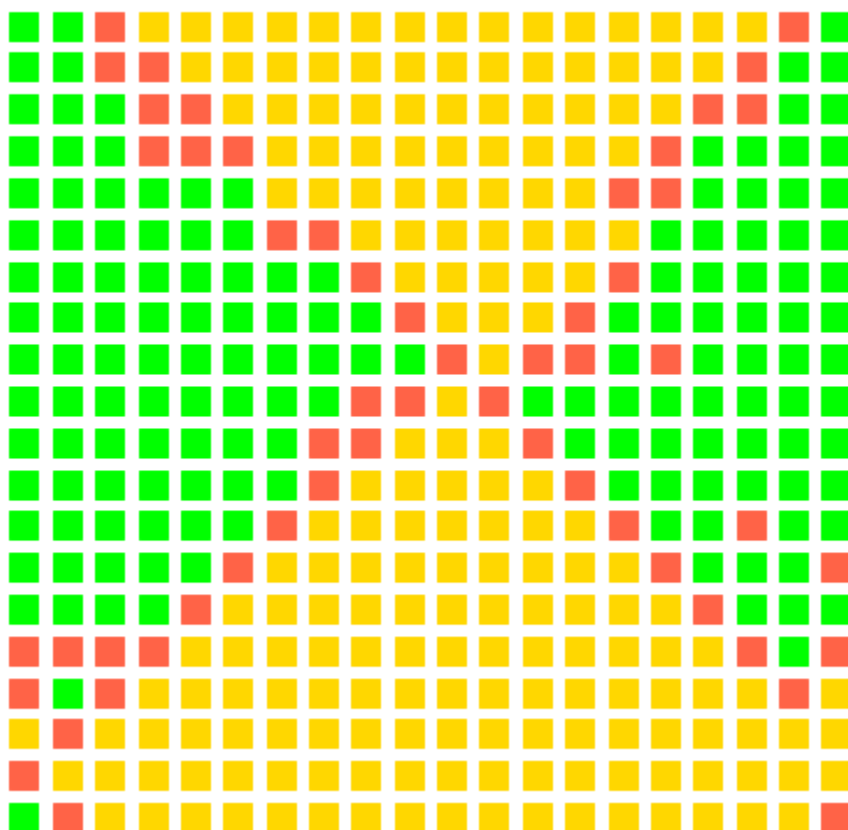
Iteração 8



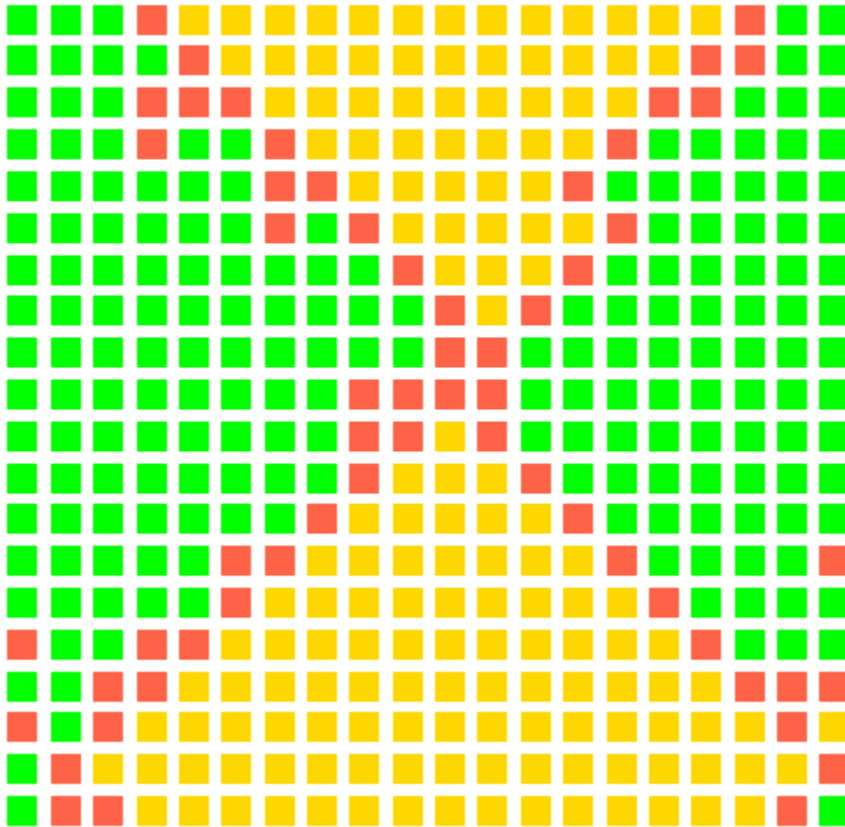
Iteração 9



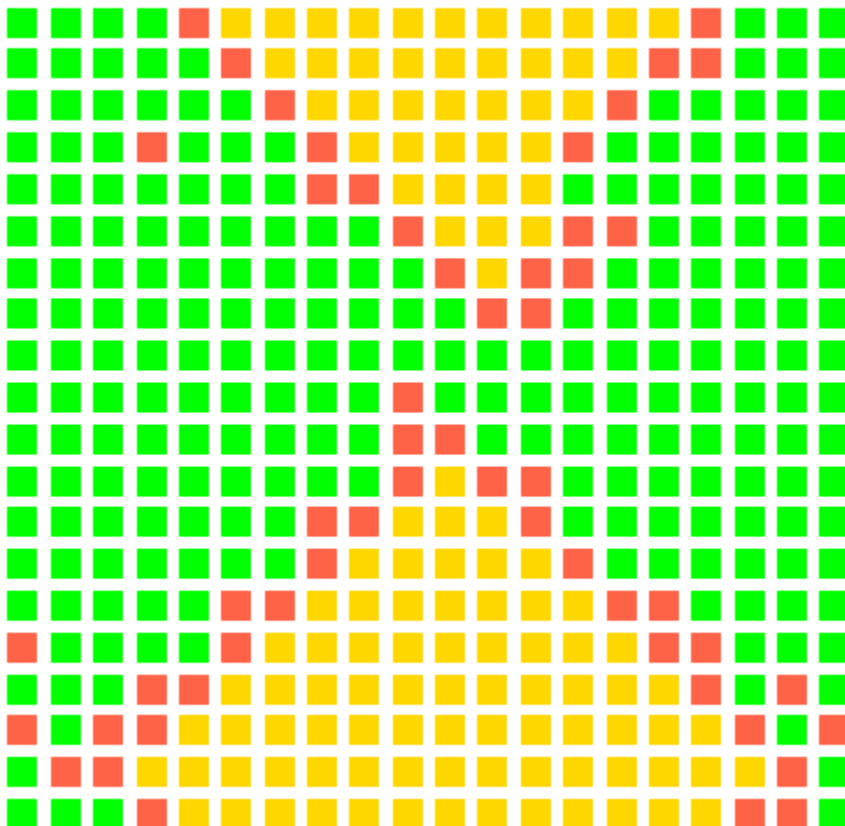
Iteração 10



Iteração 11

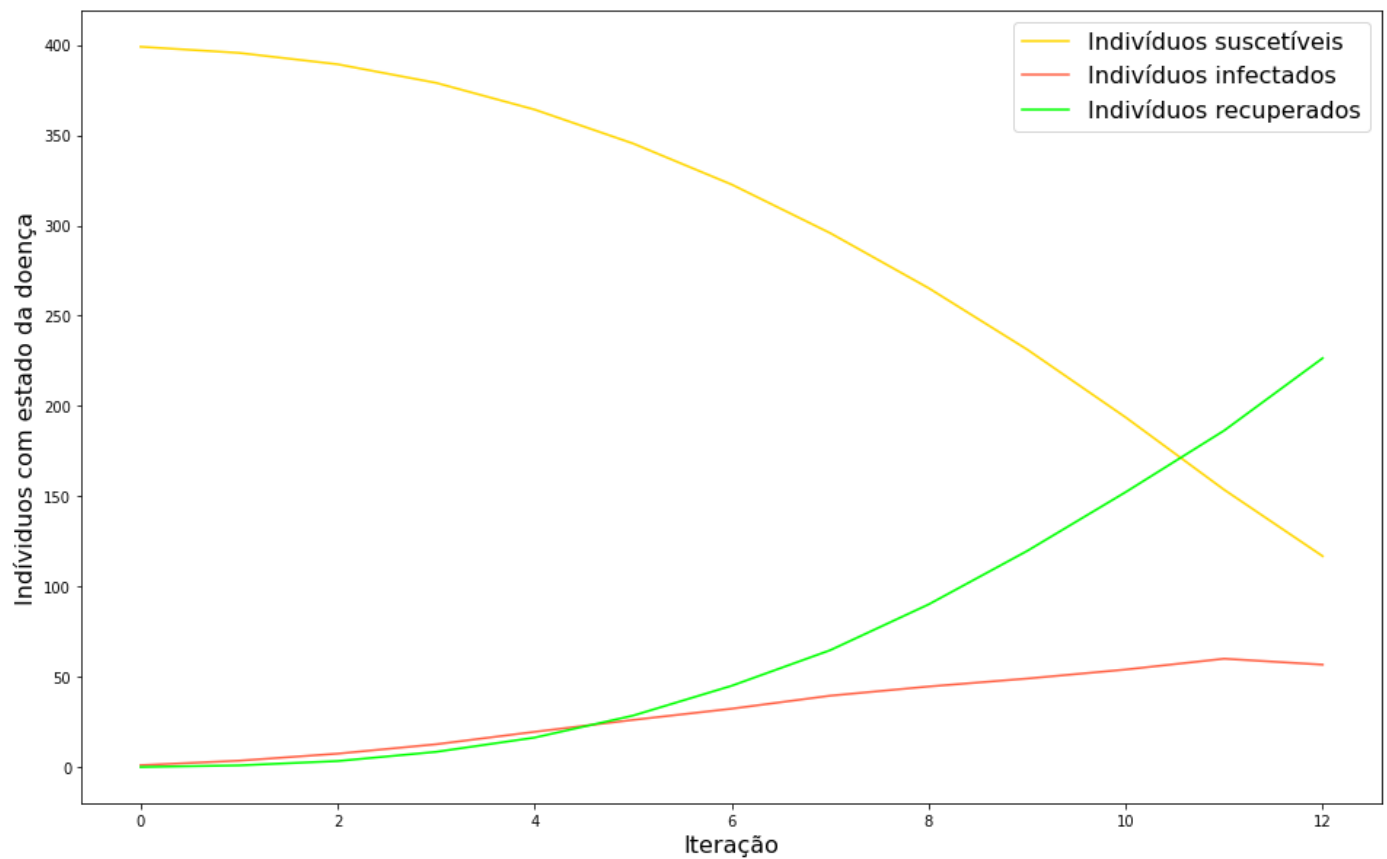


Iteração 12



```
In [46]: medias = np.mean(numero_de_casos, axis = 0)
         exibir_evolucao_dos_estados(medias, 12)
```

Evolução temporal dos estados dos indivíduos



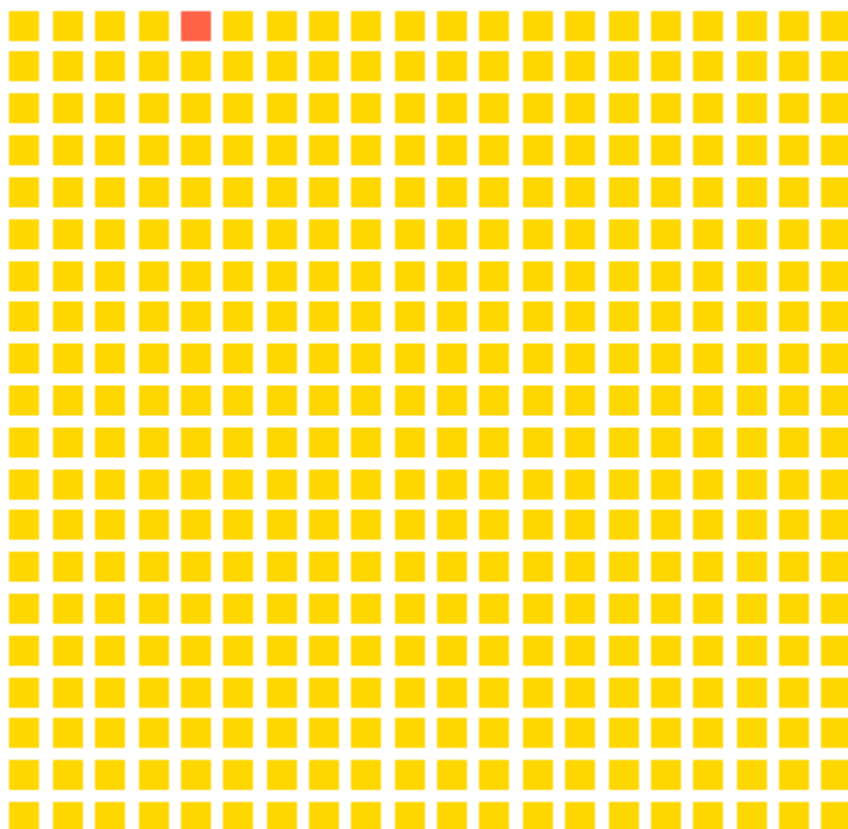
Probabilidade de contaminação = 0.55 e Probabilidade de recuperação = 0.85

```
In [41]: casos, redes = propagar_pela_rede(20, 12, 0.55, 0.85)
numero_de_casos = np.array([casos])

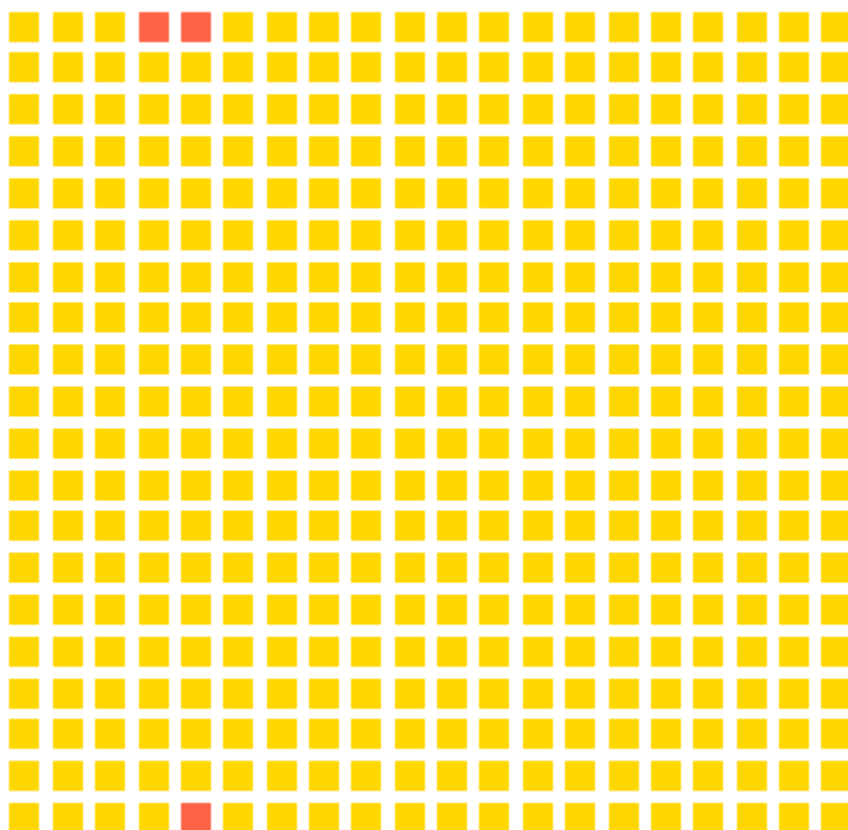
for _ in range(9):
    casos, _ = propagar_pela_rede(20, 12, 0.55, 0.85)
    numero_de_casos = np.vstack([numero_de_casos, [casos]])
```

```
In [42]: for i in range(len(redes)):
mostrar_rede(redes[i], str(i))
```

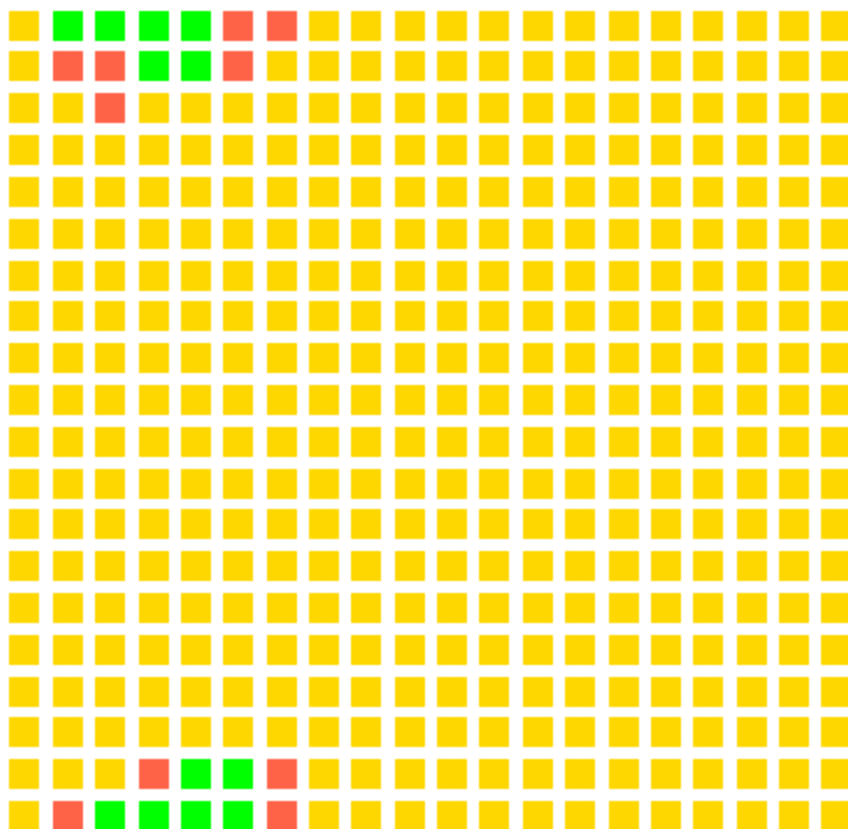
Configuração Inicial



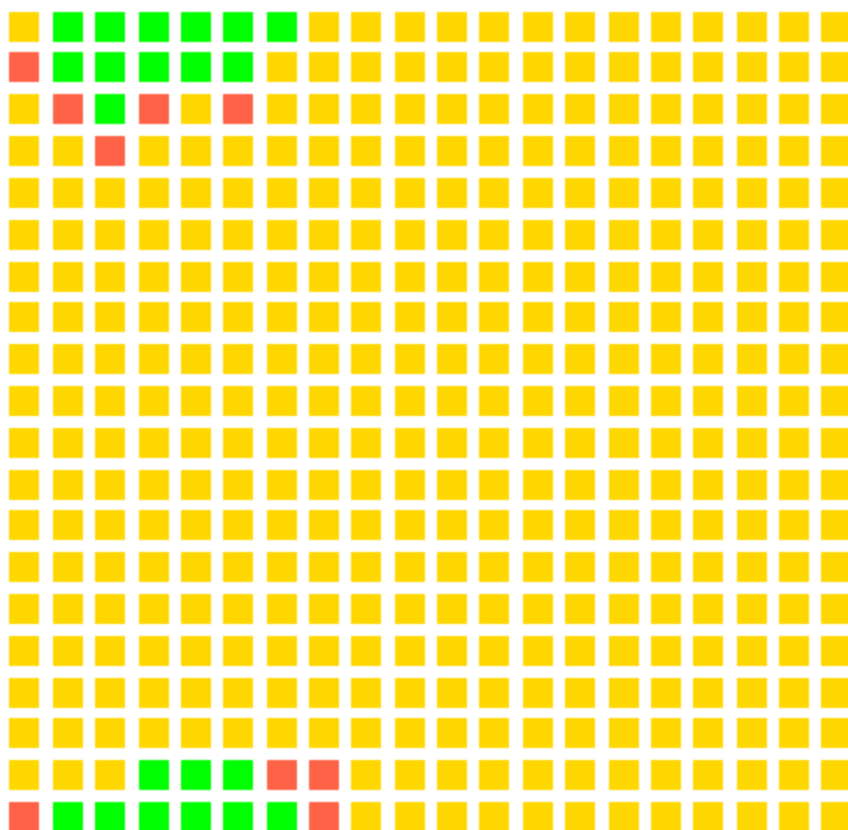
Iteração 1



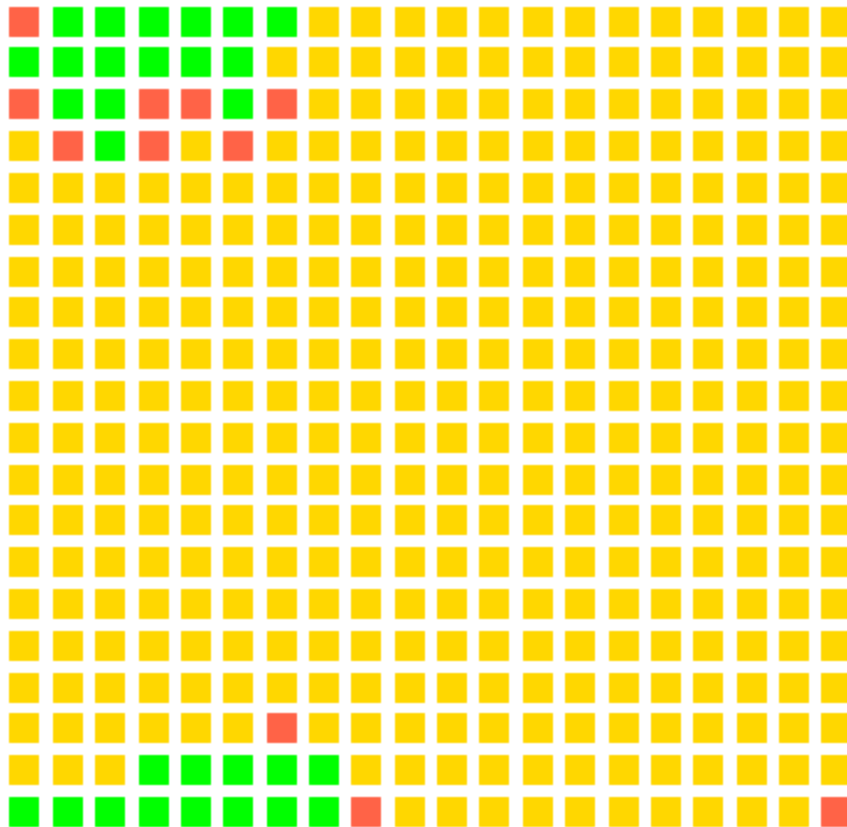
Iteração 4



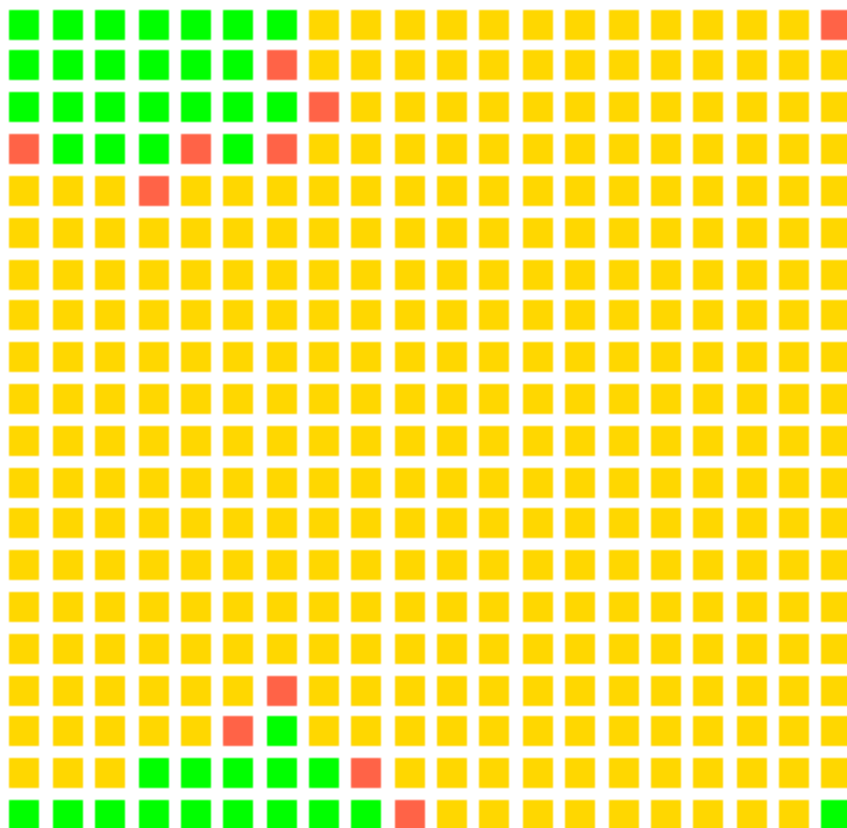
Iteração 5



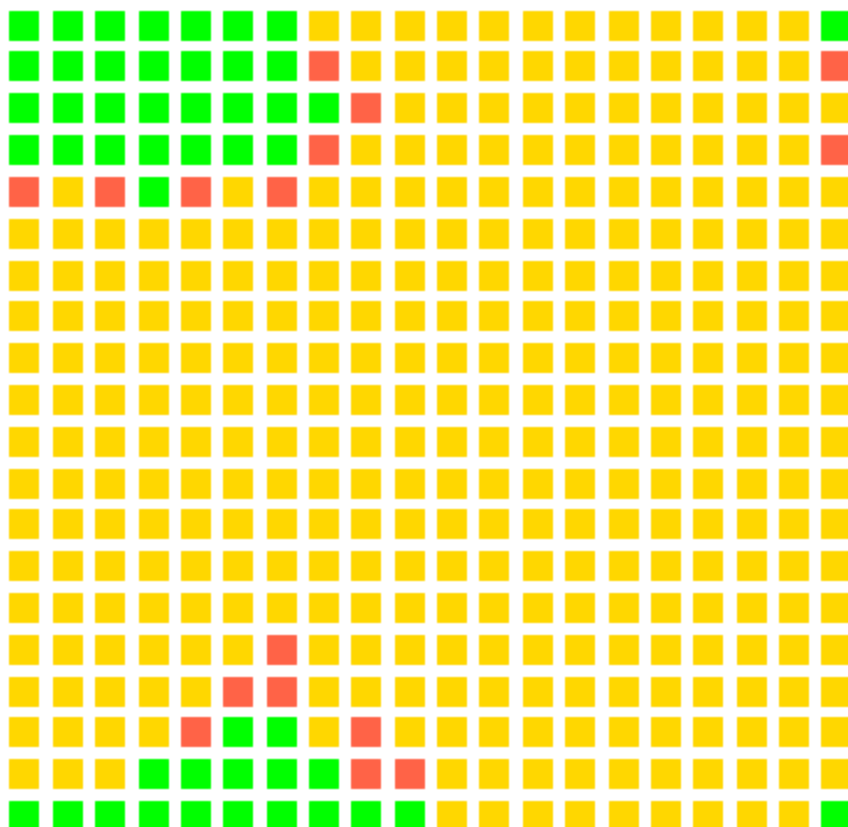
Iteração 6



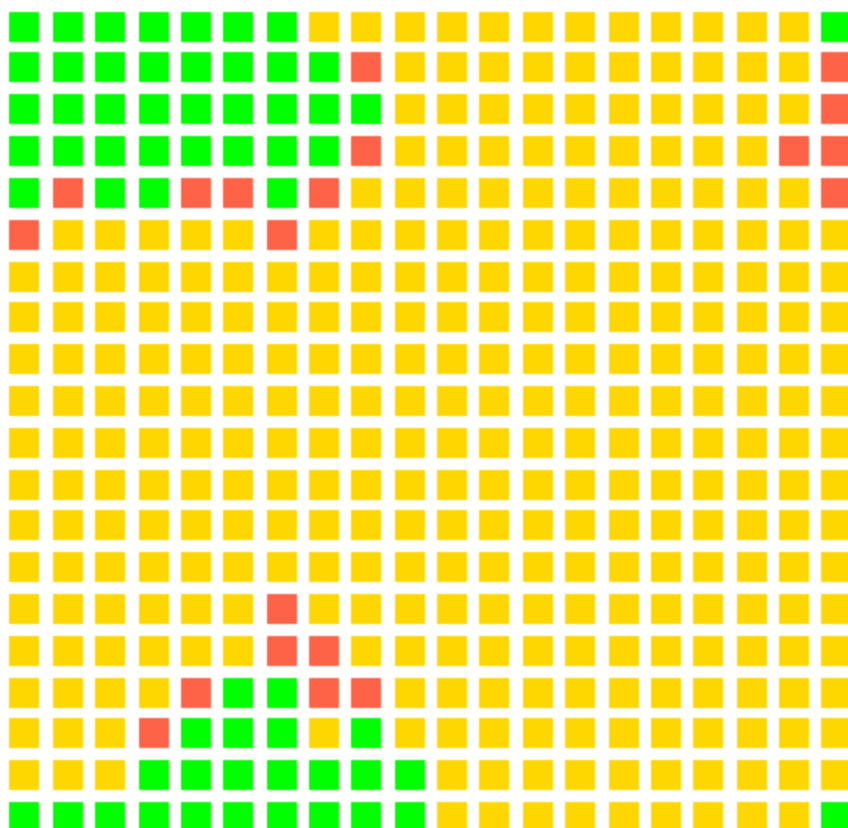
Iteração 7



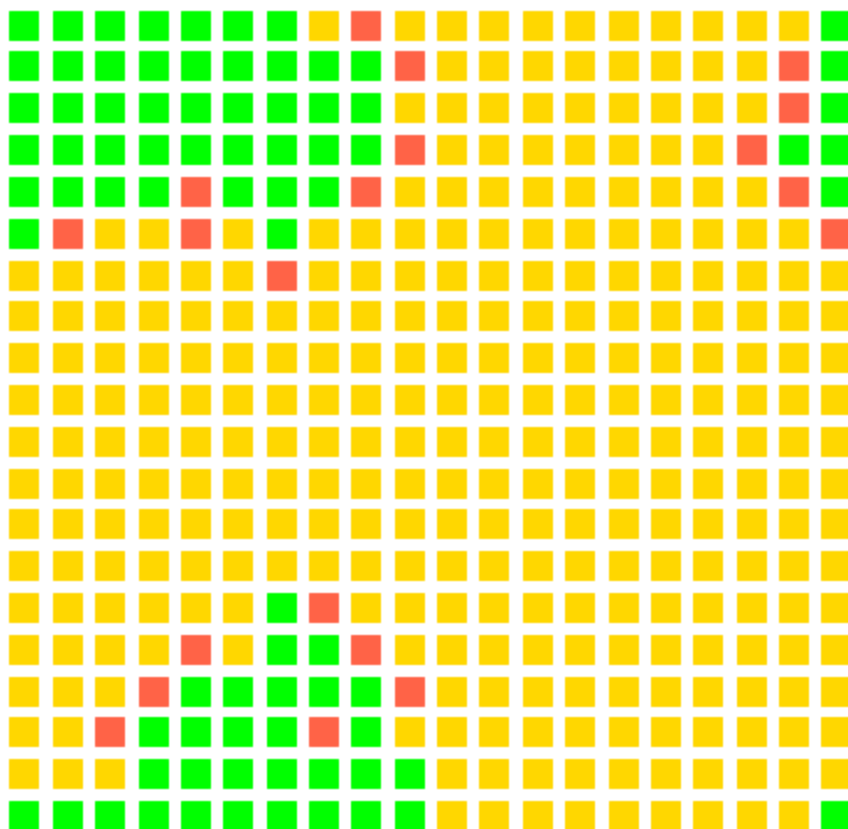
Iteração 8



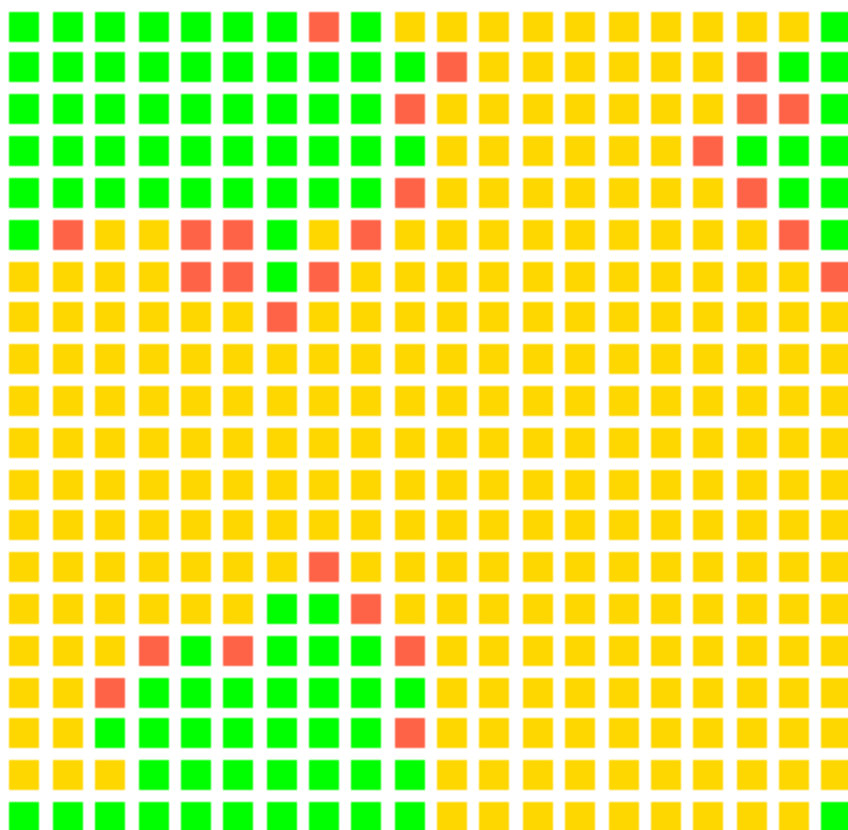
Iteração 9



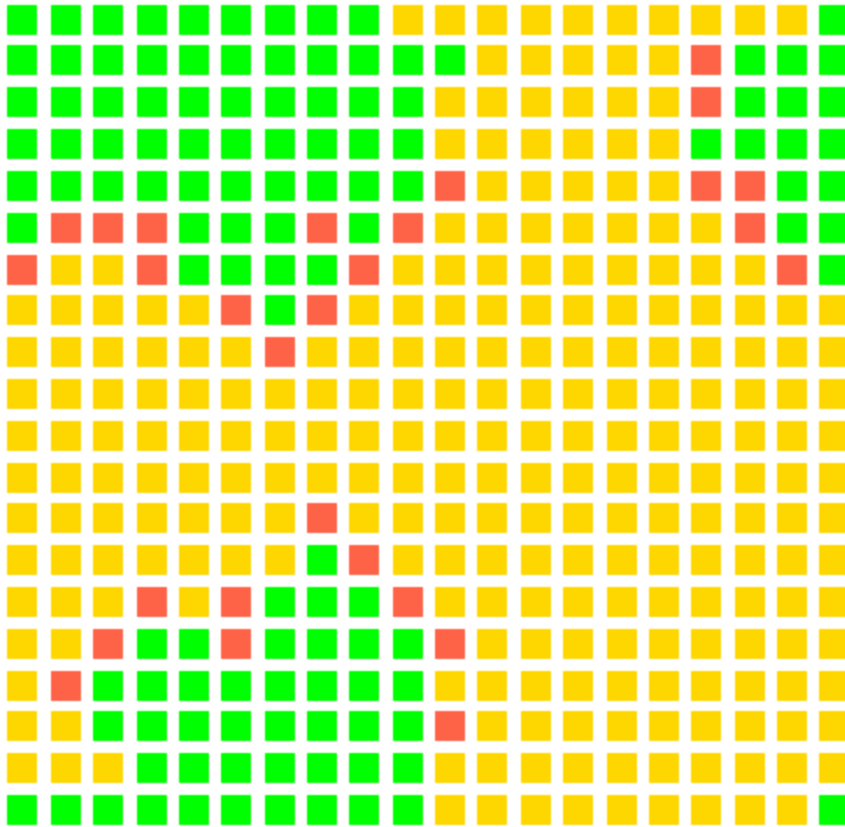
Iteração 10



Iteração 11

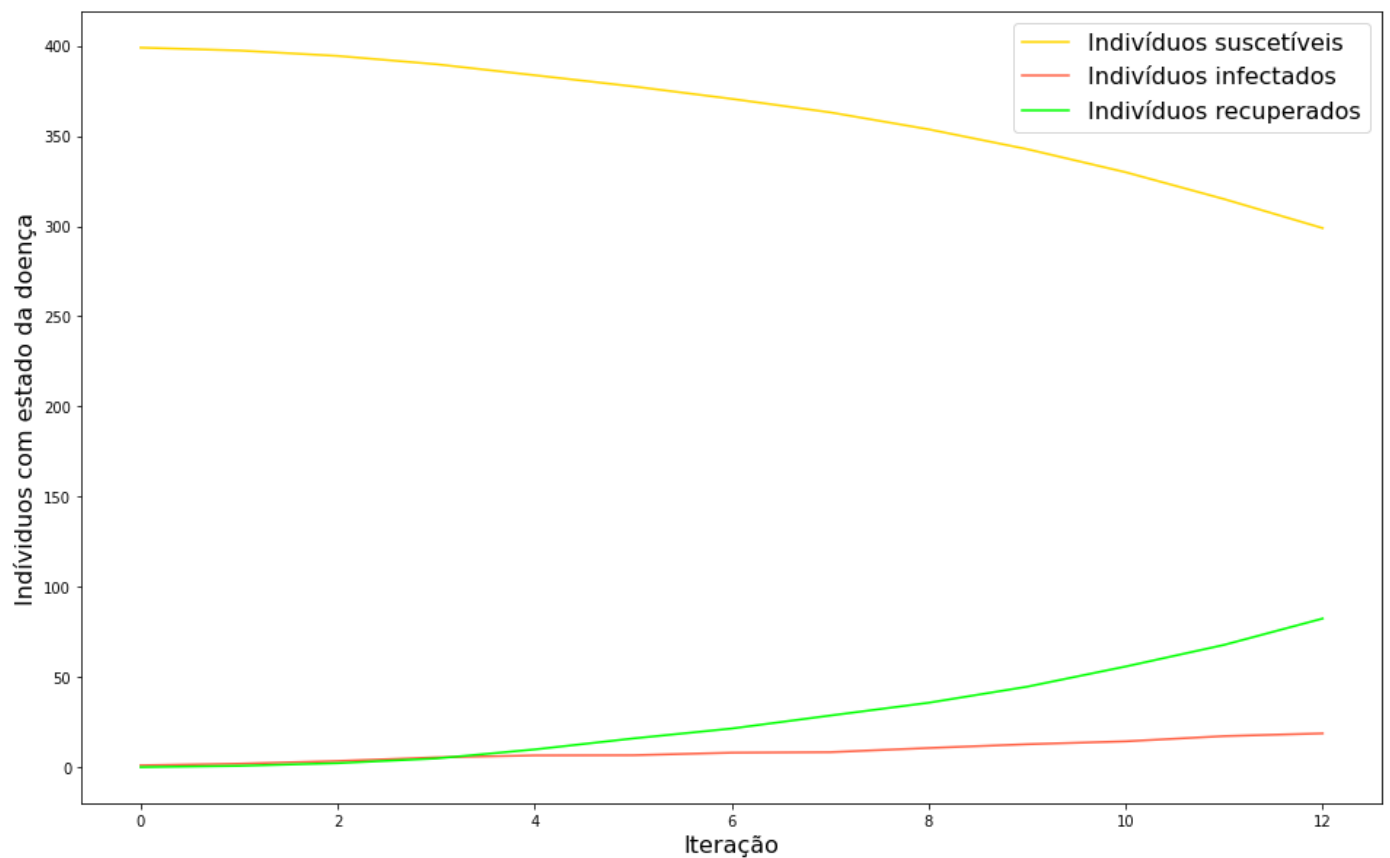


Iteração 12



```
In [43]: medias = np.mean(numero_de_casos, axis = 0)
         exibir_evolucao_dos_estados(medias, 12)
```

Evolução temporal dos estados dos indivíduos



Análise dos Resultados e Conclusão

Cada uma das redes foram criadas com 400 indivíduos, sendo que, como especificado na descrição da atividade, apenas um deles está infectado com a doença inicialmente. Além da configuração inicial, realizaram-se 12 iterações para se observar a propagação da doença na população.

Para a configuração com probabilidade de contaminação = 0.65 e probabilidade de recuperação = 0.25, notou-se que as curvas de indivíduos infectados e recuperados só foram se encontrar ao final da décima segunda iteração, o que ocorreu devido a probabilidade de recuperação não ser muito alta. Assim, indivíduos infectados tendem a passar mais tempo com a doença.

Já para a probabilidade de contaminação = 0.9 e probabilidade de recuperação = 0.65, percebe-se que mesmo com a probabilidade de contaminação sendo bem elevada, a probabilidade de recuperação ser relativamente maior que o caso anterior fez com que o número de infectados e recuperados já se igualassem na quarta iteração, sendo que após a última iteração, o número de recuperados já era mais de 4 vezes maior que o número de infectados. Percebe-se também que ao final do processo, o número de infectados apresentou tendência de queda.

Na última configuração testada, com probabilidade de contaminação = 0.55 e probabilidade de recuperação = 0.85, o número de infectados tem pouca tendência de crescimento ao longo das iterações devido a rápida recuperação.

Como indivíduos recuperados não estão mais suscetíveis a ter a doença, percebe-se que a propagação sempre irá terminar em algum momento, mesmo que demore muitas iterações para isso ocorrer. Pode-se notar que, com isso, podem acontecer casos em que nem todos os indivíduos tenham a doença antes da propagação terminar, o que provavelmente teria acontecido na continuidade da execução da última configuração testada.

Se a propagação da doença ocorresse em uma rede mundo pequeno ao invés da rede quadrada, a principal diferença que poderia ser observada seria na interação mais dinâmica entre os indivíduos, sendo que quanto mais conectada fosse a rede, maior seria a chance de propagação da doença. Esse modelo seria mais condizente com o mundo real, em que as pessoas interagem com uma quantidade variada de pessoas ao longo do dia, o que seria melhor representado pelas arestas de uma rede de mundo pequeno do que com os vizinhos imutáveis da rede quadrada.

In []: