

Número básico de reproducción, o ritmo reproductivo básico

Universidad "Politecnica Salesiana"

Alumno: Juan Cañar.
Docente: Ing. Diego Quisi.

Simulacion del grado de propagacion de la Covid-19 Ecuador

Para realizar esta simulacion se utilizar la libreria pygame para ello se debe instalar pygame

conda install -c cogsci pygame.

Introduccion

En una epidemia, el parámetro fundamental, del que todo depende, es R_0 . Este símbolo se refiere al número de personas que, mede cada infectado contagia antes de convertirse en inofensivo (bien porque está en aislamiento, hospitalizado o ha muerto).

El valor R_0 es fundamental, porque si es grande, el contagio se alarga más rápidamente. Si R_0 es 2, y si el tiempo medio en el que se permanece contagiados es una semana, y hay 1.000 infectados, entonces después de una semana los infectados será 3.000 (los 1.000 del inicio más 2.000 nuevos contagiados).

Si R_0 es 5, después de una semana los infectados será 6.000 (los 1.000 de partida más 5.000 nuevos contagiados). A este punto, el ciclo vuelve a partir, con más o menos retraso, dependiendo de cuánto tiempo un nuevo infectado emplea en convertirse a sí mismo en contagioso.

Se reconstruye la dinámica de transmisión de una enfermedad inventada con cuatro escenarios diversos:

- 1. Sin ninguna medida de contención;
- 2. Con la cuarentena absoluta, aunque se «escapa» algún infectado;
- 3. Con formas de aislamiento y la distancia de seguridad entre personas que permiten salir solo a un ciudadano de cada cuatro;
- 4. Si sale solamente un ciudadano de cada ocho. En definitiva, solo con el aislamiento se puede contener la epidemia y lograr que la respuesta sanitaria sea eficaz.

<https://www.washingtonpost.com/graphics/2020/world/corona-simulator/>

[\(https://www.washingtonpost.com/graphics/2020/world/corona-simulator/\)](https://www.washingtonpost.com/graphics/2020/world/corona-simulator/)

Entonces, el número reproductivo (R_0): Este valor representa el número promedio de personas que un individuo infectado puede contagiar. Para el COVID-19, se estima que se encuentra entre 1.4 y 4 (Qun Li, 2020). Además según estimaciones de la OMS la probabilidad de fallecimiento es de 1.2% - 4.2% según

[https://www.thelancet.com/journals/laninf/article/PIIS1473-3099\(20\)30243-7/fulltext](https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30243-7/fulltext)

[\(https://www.thelancet.com/journals/laninf/article/PIIS1473-3099\(20\)30243-7/fulltext\)](https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30243-7/fulltext)

In []:



```
from random import randrange # Obtener un numero randomico
import pygame

#Parametros de inicio
PROBA_MUERTE = 8.4 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 4.0 # Factor R0 para la simulacion COVID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 50 # Tiempo de un dia en milisegundos (Cada 25 es un dia)
nb_rows = 50 #Numero de filas
nb_cols = 50 #Numero de columnas

global display, myfont, states, states_temp #Declaracion de variables globales

#Declaro colores en formato RGB
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 247, 0)
BLACK = (0, 0, 0)

#Obtiene los vecinos dado un punto x,y
def get_vecinos(x, y):
    incx = randrange(3)
    incy = randrange(3)
    incx = (incx * 1) - 1
    incy = (incy * 1) - 1
    x2 = x + incx
    y2 = y + incy
    #Validar limites
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2] # Nuevos contagiados

#Genero las personas que cuentan con inmunidad o vacuna
def vacunar():
    for x in range(nb_cols):
        for y in range(nb_rows):
            if randrange(99) < PROBA_VACU:
                states[x][y] = 1

#Funcion que permite contar el numero de muertos de la matriz states == -1
def contar_muertes():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == -1:
                contador += 1
    return contador

#Definimos datos de inicio
states = [[0] * nb_cols for i1 in range(nb_rows)]
states_temp = states.copy()
states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion del In
```

```

it = 0 # Variable para contar Las Iteraciones
total_muerte = 0 # Contabiliza el numero de muertos
vacunar() #Llamar a La funcion vacunar

pygame.init() #Incializo el motor de juegos pygame
pygame.font.init() #Incializo el tipo de Letra
display=pygame.display.set_mode((800,750),0,32) #Tamaño de La ventana
pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
display.fill(WHITE) # Color de fondo

while True:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy() #Copia de la matriz
        #Recorrera la matriz
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10: # Numero de dias de contagio
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE: # Genero un randomico para verificar s
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infectado
                    if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas entr
                        neighbour = get_vecinos(x, y) #Obtenemos los vecinos a contagiar
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0: #Verifico que este sano
                            states_temp[x2][y2] = 10 # Contagia
            states = states_temp.copy()
            total_muerte = contar_muertes() # contar el numero de muertos

    pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
    textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,122)) #
    display.blit(textsurface, (250,30)) # Graficar el texto de muertes
    #Graficar el estado del paciente matriz
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == 0:
                color = BLUE # No infectado
            if states[x][y] == 1:
                color = GREEN # Recupero
            if states[x][y] >= 10:
                color = (states[x][y] * 12, 50, 50) # Inyectado - Rojo
            if states[x][y] == -1:
                color = BLACK # Muerto
            pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5)
            pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1))
    #Escuchar los eventos del teclado
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona y Escap
            pygame.quit() #Termino simulacion
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y espaci

```

```
#Reiniciamos valores
states = [[0] * nb_cols for i1 in range(nb_rows)]
states_temp = states.copy()
states[5][5] = 10
it = 0
total_muerte = 0
vacunar() #Llamar a la funcion vacunar
```

```
pygame.display.update()# Mandar actualizar la ventana
```

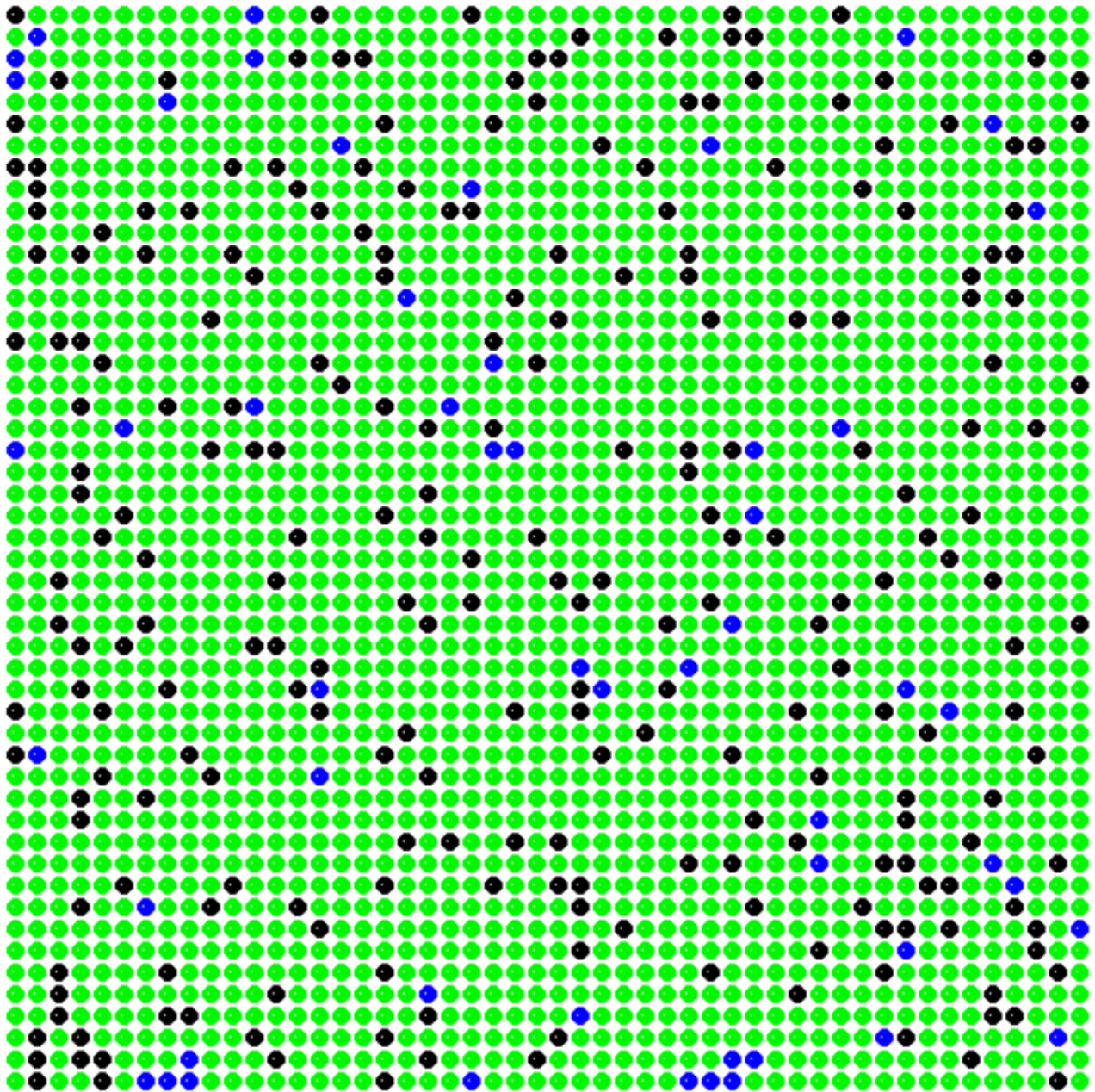
pygame 2.0.0 (SDL 2.0.12, python 3.7.6)

Hello from the pygame community. <https://www.pygame.org/contribute.html> (<https://www.pygame.org/contribute.html>)

COLORES

- Azul No infectado
 - Verde Recuperado
 - Rojo Infectado
 - Negro Muerto
-

Total muertes: 2👤👤



Practica

En consecuencia, generar 5 simulaciones:

1. R0 investigar el valor de varianza del RO dentro del Ecuador

- En epidemiología, se dice que el número básico de reproducción, o ritmo reproductivo básico (que en realidad NO es una tasa sino una ratio) es el número promedio de casos nuevos que genera un caso dado, a lo largo de un período infeccioso. Dicho de otra forma, es la cantidad de personas que se infectan en promedio a partir de una persona enferma.
- Hay enfermedades como el Corona virus que el R0 tiene un valor promedio de 2.5 o sea que una misma persona puede contagiar en promedio a 2.5 personas. El R0 de otras enfermedades infecciosas puede llegar a ser hasta de 18, por ejemplo en el caso del sarampión lo que la hace una enfermedad altamente contagiosa.

- Este concepto se aplica específicamente a una población de personas que previamente estaban libres de infección y no han sido vacunadas.
- Cuando el valor del R0 es menor que 1, cada infección existente causaría menos de una infección nueva., por ejemplo, 100 personas enfermas infectarán a menos de otras 100. Cada generación sucesiva de infecciones será más pequeña que la anterior. En este caso, la enfermedad irá disminuyendo y finalmente desaparecerá.
- Si el valor de R0 es igual a 1, cada infección existente causa una nueva infección. La enfermedad se mantendrá viva y estable, pero no habrá un brote o una epidemia y cuando el R0 es más de 1, cada infección existente causa más de una infección nueva. La enfermedad se transmitirá entre las personas y podría haber un brote o epidemia.
- Pero el comportamiento del R0 es más complejo de lo que parece, ya que está influenciado por una serie enorme de variables: es importante entender que varía de una ciudad a otra, de un día a otro y por supuesto depende de las condiciones del enfermo, pero lo más complejo es que también depende de las condiciones del virus.
- El término mal llamado "Tasa Básica reproductiva" viene de la demografía donde se utiliza para describir las tasas de natalidad. De ahí que la R se refiere a la reproducción y el 0 a la generación cero, como en el paciente "cero". Por eso se le conoce de manera conjunta como Número básico reproductivo o R0.

$$Var(X) = \frac{\sum_1^n (x_i - \bar{X})^2}{n}$$

Donde

- X: variable sobre la que se pretenden calcular la varianza
- x_i : observación número i de la variable X. i puede tomará valores entre 1 y n.
- n: número de observaciones.
- \bar{X} : Es la media de la variable X.

En resumen: el valor del R0 nos ayuda a entender la posible trayectoria de un brote.

5. Revisar e investigar algún tipo de software que permite simular la tasa de contagio en una epidemia, aplicar a los datos del Ecuador y obtener un R0 con los datos del país.

Puntos extras: Plantee y realice mejoras al modelo de simulación.

2. El valor 4, el cual representaría el peor de los casos.

In [8]:



```
from random import randrange # Obtener un numero randomico
import pygame

#Parametros de inicio
PROBA_MUERTE = 2.0 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 4.0 # Factor R0 para la simulacion COVID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 50 # Tiempo de un dia en milisegundos (Cada 25 es un dia)
nb_rows = 50 #Numero de filas
nb_cols = 50 #Numero de columnas

global display, myfont, states, states_temp #Declaracion de variables globales

#Declaro colores en formato RGB
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 247, 0)
BLACK = (0, 0, 0)

#Obtiene los vecinos dado un punto x,y
def get_vecinos(x, y):
    incx = randrange(3)
    incy = randrange(3)
    incx = (incx * 1) - 1
    incy = (incy * 1) - 1
    x2 = x + incx
    y2 = y + incy
    #Validar limites
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2] # Nuevos contagiados

#Genero las personas que cuentan con inmunidad o vacuna
def vacunar():
    for x in range(nb_cols):
        for y in range(nb_rows):
            if randrange(99) < PROBA_VACU:
                states[x][y] = 1

#Funcion que permite contar el numero de muertos de la matriz states == -1
def contar_muertes():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == -1:
                contador += 1
    return contador

#Definimos datos de inicio
states = [[0] * nb_cols for i1 in range(nb_rows)]
states_temp = states.copy()
states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion del In
```



```

it = 0 # Variable para contar Las Iteraciones
total_muerte = 0 # Contabiliza el numero de muertos
vacunar() #Llamar a La funcion vacunar

pygame.init() #Incializo el motor de juegos pygame
pygame.font.init() #Incializo el tipo de Letra
display=pygame.display.set_mode((800,750),0,32) #Tamaño de La ventana
pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
display.fill(WHITE) # Color de fondo

while True:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy() #Copia de la matriz
        #Recorrera la matriz
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10: # Numero de dias de contagio
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE: # Genero un randomico para verificar s
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infectado
                    if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas entr
                        neighbour = get_vecinos(x, y) #Obtenemos los vecinos a contagiar
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0: #Verifico que este sano
                            states_temp[x2][y2] = 10 # Contagia
            states = states_temp.copy()
            total_muerte = contar_muertes() # contar el numero de muertos

    pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
    textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,122)) #
    display.blit(textsurface, (250,30)) # Graficar el texto de muertes
    #Graficar el estado del paciente matriz
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == 0:
                color = BLUE # No infectado
            if states[x][y] == 1:
                color = GREEN # Recupero
            if states[x][y] >= 10:
                color = (states[x][y] * 12, 50, 50) # Inyectado - Rojo
            if states[x][y] == -1:
                color = BLACK # Muerto
            pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5)
            pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1))
    #Escuchar los eventos del teclado
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona y Escap
            pygame.quit() #Termino simulacion
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y espaci

```

```

#Reiniciamos valores
states = [[0] * nb_cols for i1 in range(nb_rows)]
states_temp = states.copy()
states[5][5] = 10
it = 0
total_muerte = 0
vacunar() #Llamar a la funcion vacunar

```

```

pygame.display.update()# Mandar actualizar la ventana

```

pygame 2.0.0 (SDL 2.0.12, python 3.7.6)

Hello from the pygame community. <https://www.pygame.org/contribute.html> (<https://www.pygame.org/contribute.html>)

```

-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-8-4792f591e97f> in <module>
    70
    71 while True:
--> 72     pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    73     it = it + 1
    74     if it <= 10000 and it >= 2:

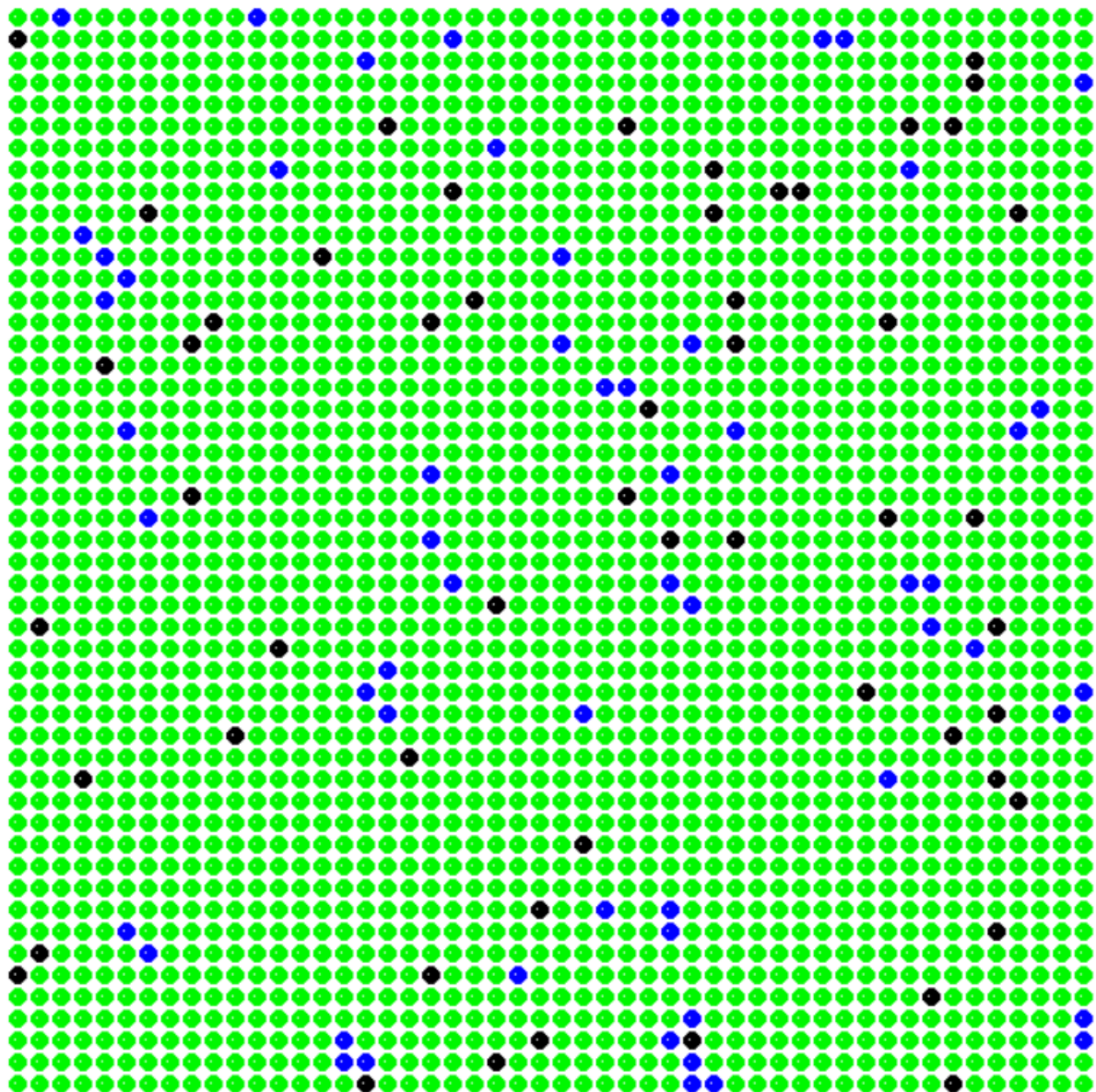
```

KeyboardInterrupt:

COLORES

- Azul No infectado
- Verde Recuperado
- Rojo Infectado
- Negro Muerto

Total muertes: 54



4. El valor 1.4 en el mejor de los casos

In [1]:



```
from random import randrange # Obtener un numero randomico
import pygame

#Parametros de inicio
PROBA_MUERTE = 4.0 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 1.4 # Factor R0 para la simulacion COVID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 50 # Tiempo de un dia en milisegundos (Cada 25 es un dia)
nb_rows = 50 #Numero de filas
nb_cols = 50 #Numero de columnas

global display, myfont, states, states_temp #Declaracion de variables globales

#Declaro colores en formato RGB
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 247, 0)
BLACK = (0, 0, 0)

#Obtiene los vecinos dado un punto x,y
def get_vecinos(x, y):
    incx = randrange(3)
    incy = randrange(3)
    incx = (incx * 1) - 1
    incy = (incy * 1) - 1
    x2 = x + incx
    y2 = y + incy
    #Validar limites
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2] # Nuevos contagiados

#Genero las personas que cuentan con inmunidad o vacuna
def vacunar():
    for x in range(nb_cols):
        for y in range(nb_rows):
            if randrange(99) < PROBA_VACU:
                states[x][y] = 1

#Funcion que permite contar el numero de muertos de la matriz states == -1
def contar_muertes():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == -1:
                contador += 1
    return contador

#Definimos datos de inicio
states = [[0] * nb_cols for i1 in range(nb_rows)]
states_temp = states.copy()
states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion del In
```

```

it = 0 # Variable para contar Las Iteraciones
total_muerte = 0 # Contabiliza el numero de muertos
vacunar() #Llamar a La funcion vacunar

pygame.init() #Incializo el motor de juegos pygame
pygame.font.init() #Incializo el tipo de Letra
display=pygame.display.set_mode((800,750),0,32) #Tamaño de La ventana
pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
display.fill(WHITE) # Color de fondo

while True:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy() #Copia de la matriz
        #Recorrera la matriz
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10: # Numero de dias de contagio
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE: # Genero un randomico para verificar s
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infectado
                    if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas entr
                        neighbour = get_vecinos(x, y) #Obtenemos los vecinos a contagiar
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0: #Verifico que este sano
                            states_temp[x2][y2] = 10 # Contagia
            states = states_temp.copy()
            total_muerte = contar_muertes() # contar el numero de muertos

    pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
    textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,122)) #
    display.blit(textsurface, (250,30)) # Graficar el texto de muertes
    #Graficar el estado del paciente matriz
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == 0:
                color = BLUE # No infectado
            if states[x][y] == 1:
                color = GREEN # Recupero
            if states[x][y] >= 10:
                color = (states[x][y] * 12, 50, 50) # Inyectado - Rojo
            if states[x][y] == -1:
                color = BLACK # Muerto
            pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5)
            pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1))
    #Escuchar los eventos del teclado
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona y Escap
            pygame.quit() #Termino simulacion
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y espaci

```

```
#Reiniciamos valores
states = [[0] * nb_cols for i1 in range(nb_rows)]
states_temp = states.copy()
states[5][5] = 10
it = 0
total_muerte = 0
vacunar() #Llamar a la funcion vacunar
```

```
pygame.display.update()# Mandar actualizar la ventana
```

pygame 2.0.0 (SDL 2.0.12, python 3.7.6)

Hello from the pygame community. <https://www.pygame.org/contribute.html> (<https://www.pygame.org/contribute.html>)

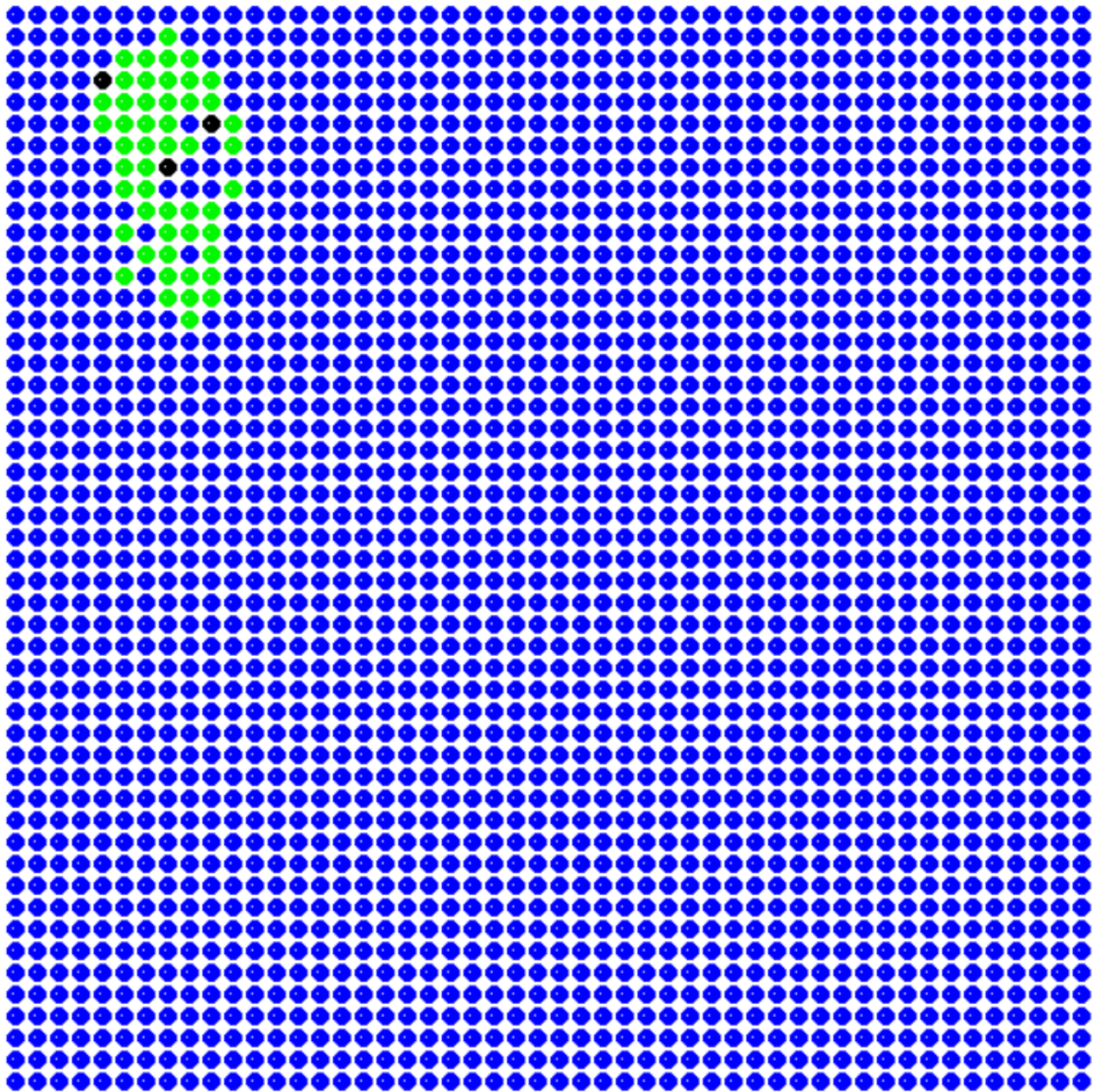
```
-----
error                                Traceback (most recent call last)
<ipython-input-1-3c6540bca17f> in <module>
    127                 vacunar() #Llamar a la funcion vacunar
    128
--> 129     pygame.display.update()# Mandar actualizar la ventana
```

error: video system not initialized

COLORES

- Azul No infectado
- Verde Recuperado
- Rojo Infectado
- Negro Muerto

Total muertes: 3



5. Revisar e investigar algun tipo de software que permite simular la taza de contagio en una epidemia, aplicar a los datos del Ecuador y obtener un RO con los datos del pais.

In [16]:

```
import scipy.integrate as spi
import numpy as np
import pylab as pl
'''tamaño poblacional'''
N=1
beta=12.4247
gamma=0.014286
'''time step'''
TS=1.0
ND=70.0
S0=1-1e-6
I0=1e-6
INPUT = (S0, I0, 0.0)

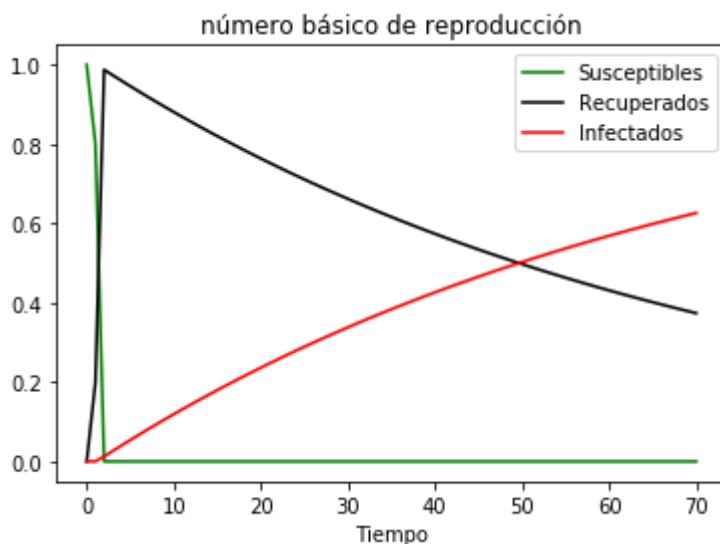
def diff_eqs(INP,t):
    Y=np.zeros((3))
    V = INP
    '''Las ecuaciones diferenciales'''
    Y[0] = - beta * V[0] * V[1]
    Y[1] = beta * V[0] * V[1] - gamma * V[1]
    Y[2] = gamma * V[1]
    return Y # For odeint

t_start = 0.0; t_end = ND; t_inc = TS
t_range = np.arange(t_start, t_end+t_inc, t_inc)
RES = spi.odeint(diff_eqs,INPUT,t_range)

#Gráfica
pl.plot(RES[:,0]*N, '-g', label='Susceptibles')
pl.plot(RES[:,1]*N, '-k', label='Recuperados')
pl.plot(RES[:,2]*N, '-r', label='Infectados')
pl.legend(loc=0)
pl.title('número básico de reproducción')
pl.xlabel('Tiempo')
#pl.savefig('sirpy')
```

Out[16]:

Text(0.5, 0, 'Tiempo')



Simulación en R: Es un software libre usado en el campo estadístico, minería de datos, investigación biomédica y matemática financiera; se distribuye bajo licencia GNU GLP y se encuentra disponible para Windows, Macintosh, UNIX y Linux (10). El software permite cargar librerías para la simulación del modelo epidemiológico; se hace uso de la librería desolver, que permite la solución de ecuaciones diferenciales.

Simulación en Berkeley Madonna: Es un software de modelado matemático desarrollado por la Universidad de Berkeley; es de gran uso en el modelado de sistemas epidemiológicos, es utilizado este software para la simulación de sus modelos.

Simulación en Matlab: Software matemático basado en matrices muy difundido y utilizado en el campo de la ingeniería y de la matemática ;para la simulación del modelo epidemiológico es necesario crear un script que realiza la simulación y genera una gráfica con el resultado del modelo

Analisis

Es notable que en cada simulación, y gráfica, cambia y varían los datos esto debido a las valor de R_0 , ya que si se da un valor de R_0 bastante bajo, los casos de muerte son considerablemente bajos, lo que sucedería si fuese lo contrario con un valor de R_0 mas elevado, el índice de mortalidad sería demasiado alto y también el de los contagios, para las simulaciones se establecieron valores de 1.4 es mínimo ya que representa el valor más pequeño de R_0 y con 4, que representa el valor máximo R_0 .

Conclusiones

Mediante el R_0 nos permite tener un número promedio de casos nuevos que genera un caso dado, a lo largo de un período infeccioso. Dicho de otra forma, es la cantidad de personas que se infectan en promedio a partir de una persona enferma.

Hay enfermedades como el Corona virus que el R_0 tiene un valor promedio de 2.5 o sea que una misma persona puede contagiar en promedio a 2.5 personas

Opinion

Es un buen método para trabajar con simulaciones, pero al no existir datos tan verídicos esto queda sujeto a cambios, debido a que las personas ya no reportan los casos, y se convierten en personas de alto riesgo que contagian a otros. La mayoría de personas ya no reportan o acuden a un hospital y las autoridades no tienen estadísticas que se acercan a la realidad como cuando tenían en cuarentena.

Referencias

- <https://dialnet.unirioja.es/descarga/articulo/6070913.pdf>
(<https://dialnet.unirioja.es/descarga/articulo/6070913.pdf>)
- <http://code.intef.es/simulamos-una-epidemia-virica/> (<http://code.intef.es/simulamos-una-epidemia-virica/>)
- https://www.juanbarrios.com/el-indice-r0-el-indicador-para-administrar-la-pandemia-del-covid19/#Que_significan_los_valores_de_R0_en_el_caso_de_la_infeccion_por_Corona_virus
(https://www.juanbarrios.com/el-indice-r0-el-indicador-para-administrar-la-pandemia-del-covid19/#Que_significan_los_valores_de_R0_en_el_caso_de_la_infeccion_por_Corona_virus)