Simulacion-Contactos-RO

May 30, 2021

1 Practica

En concecuencia, generar 5 simulaciones: 1. R0 investigar el valor de varianza del RO dentro del Ecuador. 2. El valor 4, el cual representaría el peor de los casos. 3. El valor 1.4 en el mejor de los casos 4. Revisar e inveStigar algun tipo de software que permita simular la taza de contagio en una epidemia, aplicar a los datos del Ecuador y obtener un R0 con los datos del país. 5. Investigar datos de vacunación, muerte y el RO calculado de los datos historico RO = beta/gamma

Puntos extras: Plantee y realize mejoras al modelo de simulacion.

[1]:
```
#El valor 4, el cual representaría el peor de los casos.
from random import randrange
import pygame

running = True
#Parametros de inicio
PROBA_MUERTE = 8.4  # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 4  # Factor RO para la simulacion COVID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 75 # Tiempo de un dia en milisegundos (Cada 75 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
```

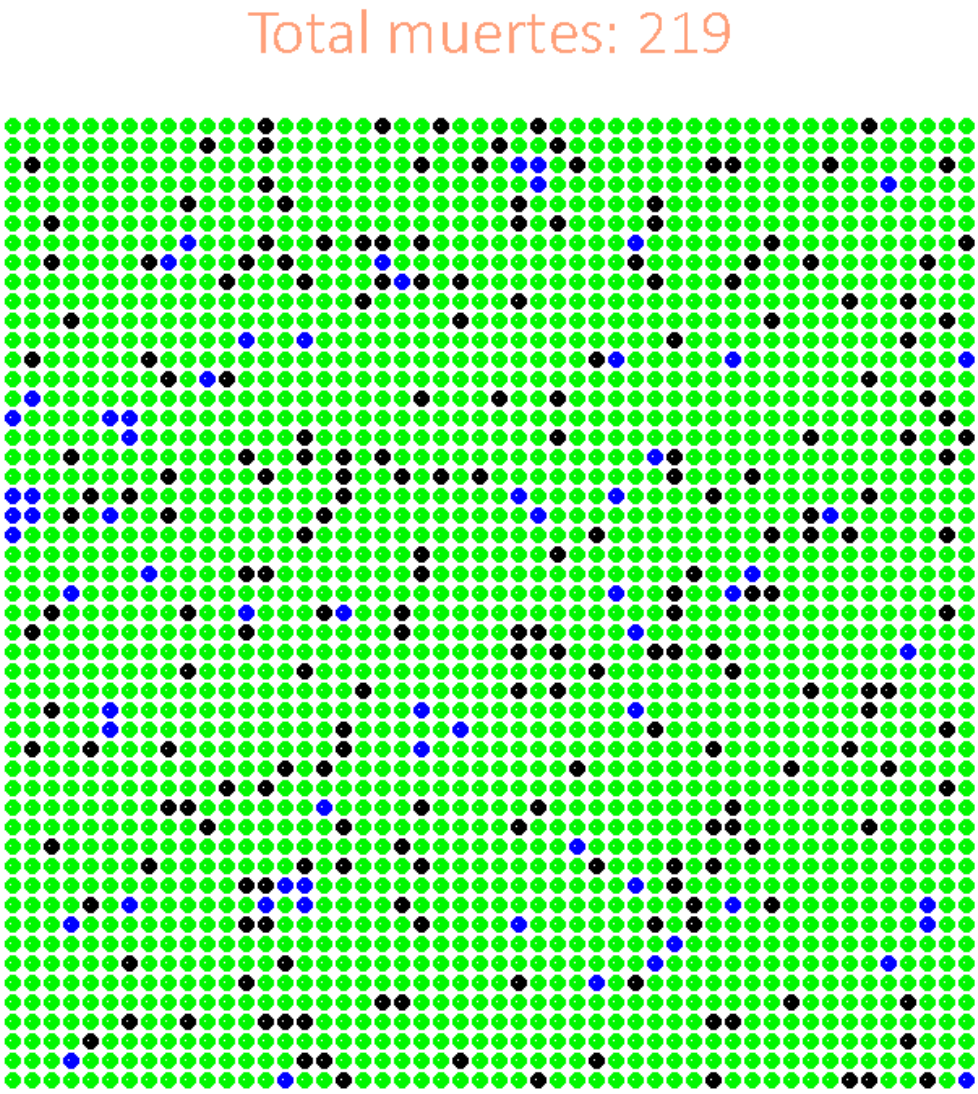
```
    x2 = x + incx
    y2 = y + incy
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2]

#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
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
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()
display=pygame.display.set_mode((800,750),0,32) #Tamanio 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 running:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
```

```
        states_temp = states.copy()
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10:
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE:
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura
                if state >= 10 and state <= 20:
                    if randrange(99) < PROBA_INFECT: # Infecto a las personas
↪cercanas entre  10 y 20
                        neighbour = get_vecinos(x, y)
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0:
                            states_temp[x2][y2] = 10
        states = states_temp.copy()
        total_muerte = contar_muertes() # contar el numero de muertos

    pygame.draw.rect(display, WHITE, (250, 30, 350, 50)) # Grafico la fondo
    textsurface = font.render("Total muertes: "+ str(total_muerte), False,
↪(255,160,122)) #El numero de muertos
    display.blit(textsurface, (250, 30))
    #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) # Injectado - 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))
    #Escuachar los eventos del teclado
    for event in pygame.event.get():
```

```
        if (event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE) or␣
↪event.type == pygame.QUIT: #Presiona y Escape
            running = False #Termino simulacion
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:␣
↪#Presiona y espacio
            #Reiniciamos valores
            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
            it = 0
            total_muerte = 0
            vacunar()
    pygame.display.update()# Mandar actualizar la ventana
pygame.quit()
```

```
pygame 2.0.1 (SDL 2.0.14, Python 3.8.3)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

2 Modelo 1



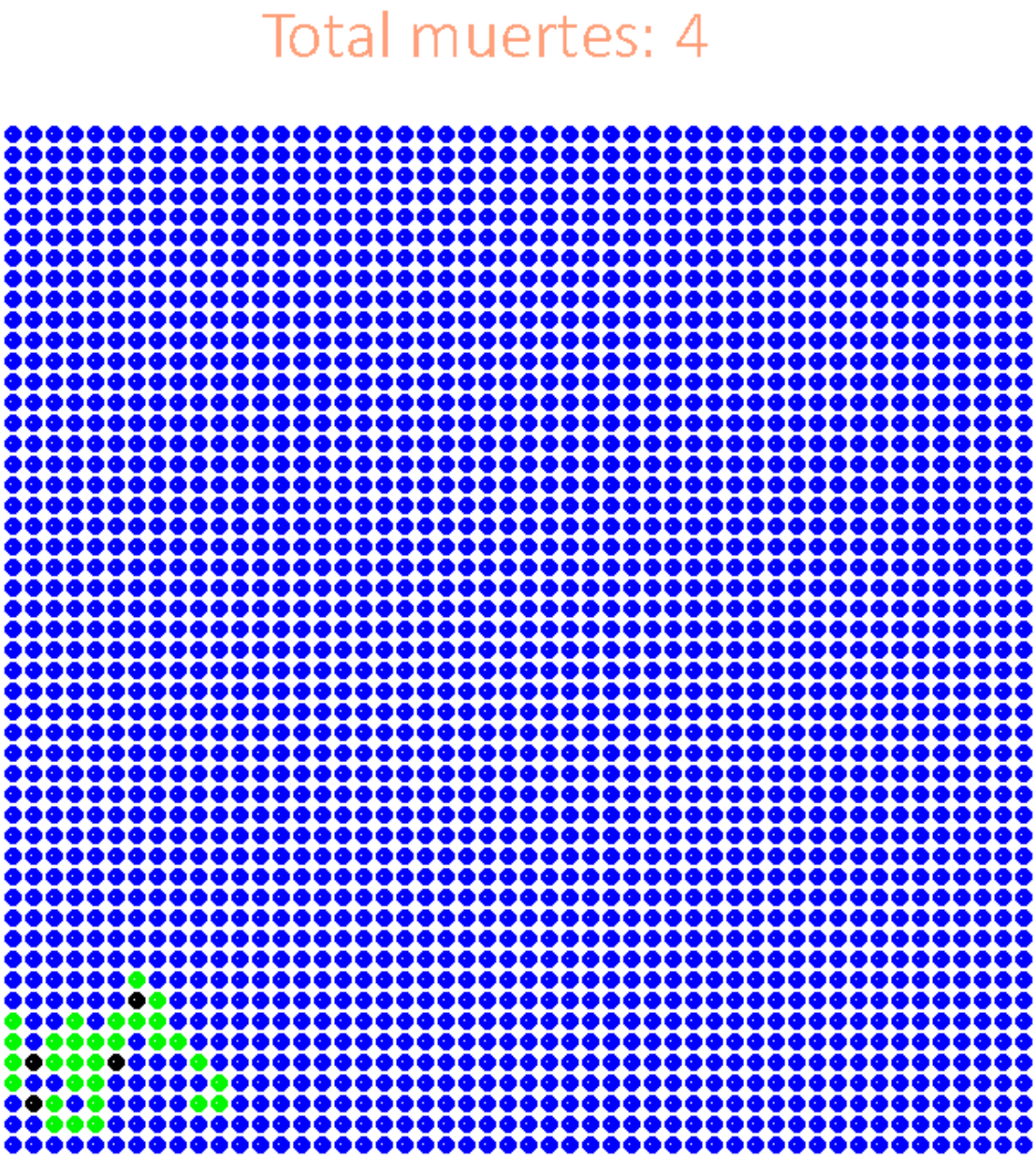
[ ]:
```
#El valor 1.4 en el mejor de los casos
from random import randrange
import pygame

running = True
#Parametros de inicio
PROBA_MUERTE = 8.4  # 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 = 75 # Tiempo de un dia en milisegundos (Cada 75 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
    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]

#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
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
```

```
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()
display=pygame.display.set_mode((800,750),0,32) #Tamanio 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 running:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy()
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10:
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE:
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura
                if state >= 10 and state <= 20:
                    if randrange(99) < PROBA_INFECT: # Infecto a las personas␣
↪cercanas entre  10 y 20
                        neighbour = get_vecinos(x, y)
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0:
                            states_temp[x2][y2] = 10
        states = states_temp.copy()
        total_muerte = contar_muertes() # contar el numero de muertos

    pygame.draw.rect(display, WHITE, (250, 30, 350, 50)) # Grafico la fondo
    textsurface = font.render("Total muertes: "+ str(total_muerte), False,␣
↪(255,160,122)) #El numero de muertos
    display.blit(textsurface, (250, 30))
    #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) # Injectado - 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))
    #Escuachar los eventos del teclado
    for event in pygame.event.get():
        if (event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE) or␣
↪event.type == pygame.QUIT: #Presiona y Escape
            running = False #Termino simulacion
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:␣
↪#Presiona y espacio
            #Reiniciamos valores
            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
            it = 0
            total_muerte = 0
            vacunar()
    pygame.display.update()# Mandar actualizar la ventana
pygame.quit()
```

3 Modelo 2



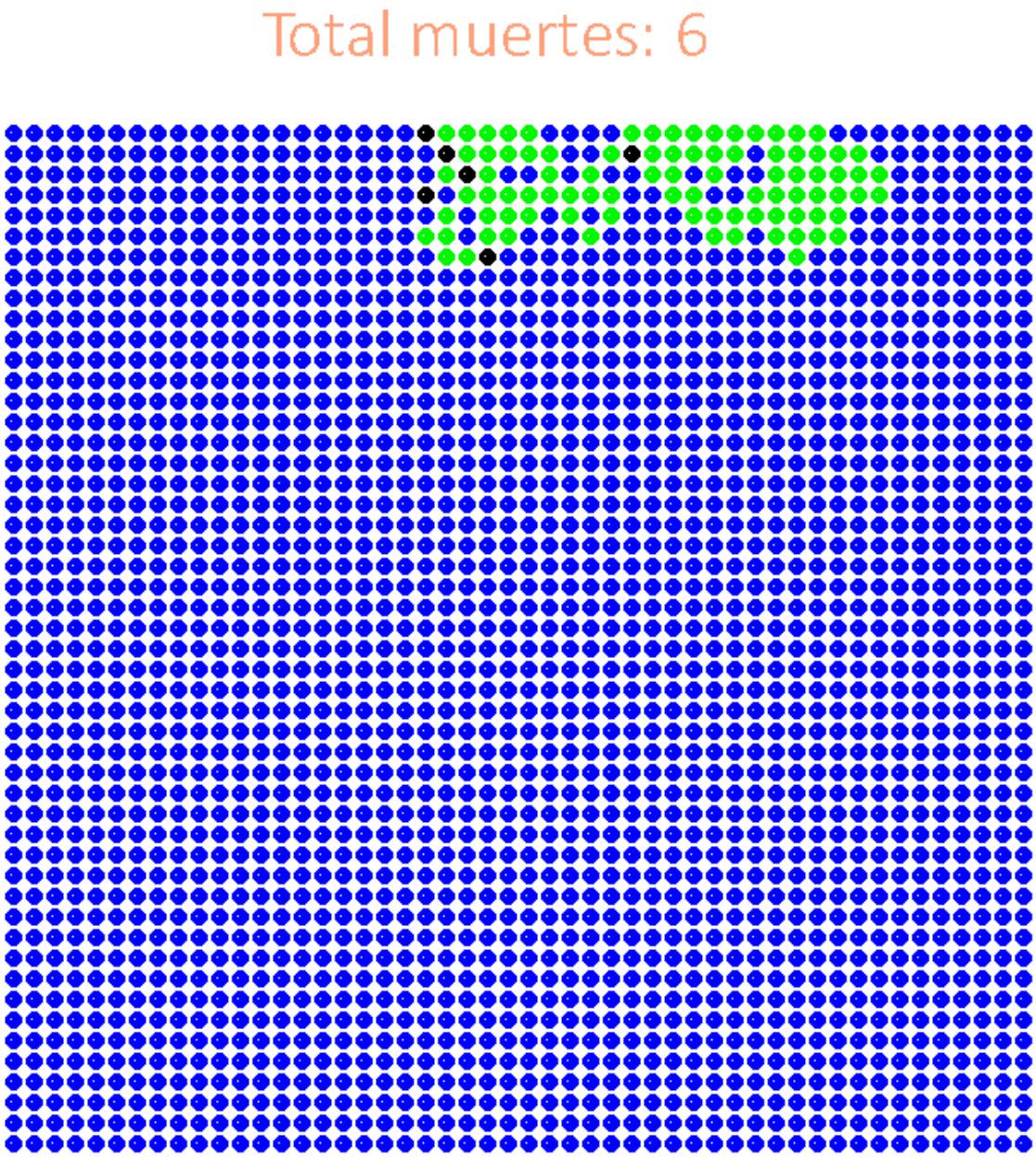
[ ]:
```
#
from random import randrange
import pygame

running = True
#Parametros de inicio
PROBA_MUERTE = 8.4  # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 2.2  # Factor RO para la simulacion COVID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
```

```
SIMULACION_SPEED = 75 # Tiempo de un dia en milisegundos (Cada 75 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
    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]

#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
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
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()
display=pygame.display.set_mode((800,750),0,32) #Tamanio 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 running:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy()
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10:
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE:
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura
                if state >= 10 and state <= 20:
                    if randrange(99) < PROBA_INFECT: # Infecto a las personas␣
↪cercanas entre  10 y 20
                        neighbour = get_vecinos(x, y)
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0:
                            states_temp[x2][y2] = 10
        states = states_temp.copy()
        total_muerte = contar_muertes() # contar el numero de muertos

    pygame.draw.rect(display, WHITE, (250, 30, 350, 50)) # Grafico la fondo
    textsurface = font.render("Total muertes: "+ str(total_muerte), False,␣
↪(255,160,122)) #El numero de muertos
    display.blit(textsurface, (250, 30))
```

```
    #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) # Injectado - 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))
    #Escuachar los eventos del teclado
    for event in pygame.event.get():
        if (event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE) or␣
↪event.type == pygame.QUIT: #Presiona y Escape
            running = False #Termino simulacion
        if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:␣
↪#Presiona y espacio
            #Reiniciamos valores
            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
            it = 0
            total_muerte = 0
            vacunar()
    pygame.display.update()# Mandar actualizar la ventana
pygame.quit()
```

4 Modelo 3



5 Conclusiones

Dado que los valores de R0 se encuentra entre 1.4 y 4. Se usó la simulación con el valor de R0 2.2 que hace referencia a una perspectiva realizada por “www.cureus.com”. Como se puede observar el valor de R0 influye mucho en la cantidad de personas que llegan a contagiarse, pues al tener el valor R0 igual a 4 los contagiados se extienden a los largo de toda la población.

6 Opinion

Personalmente creo que las medidas tomadas como la cuarentena nacional, son medidas muy necesarias para poder reducir el indice de rango de contagio, dado que si el numero de contagiados aumenta excesivamente no será posible tratar medicamente de la manera adecuada a todos los contagiados y aumentaría la probabilidad de muerte.

7 Referencia

- http://code.intef.es/simulamos-una-epidemia-virica/
- https://www.cureus.com/articles/29796-perspective-from-ecuador-the-second-country-with-more-confirmed-cases-of-coronavirus-disease-2019-in-south-america-a-review

[ ]: