

Universidad Politecnica Salesiana

Nombre: Jessica Ñauta

Asignatura: Simulación

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;
- 1. Con la cuarentena absoluta, aunque se «escapa» algún infectado;
- 1. Con formas de aislamiento y la distancia de seguridad entre personas que permiten salir solo a un ciudadano de cada cuatro;
- 1. 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/>

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)

```
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
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) #Tamano 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 veri
                        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 cercan
                        neighbour = get_vecinos(x, y) #Obtenemos los vecinos a conta
                        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,
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),
        pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1,
#Escuchar los eventos del teclado
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona
        pygame.quit() #Termino simulacion
    if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y
        #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 3.8.5)

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

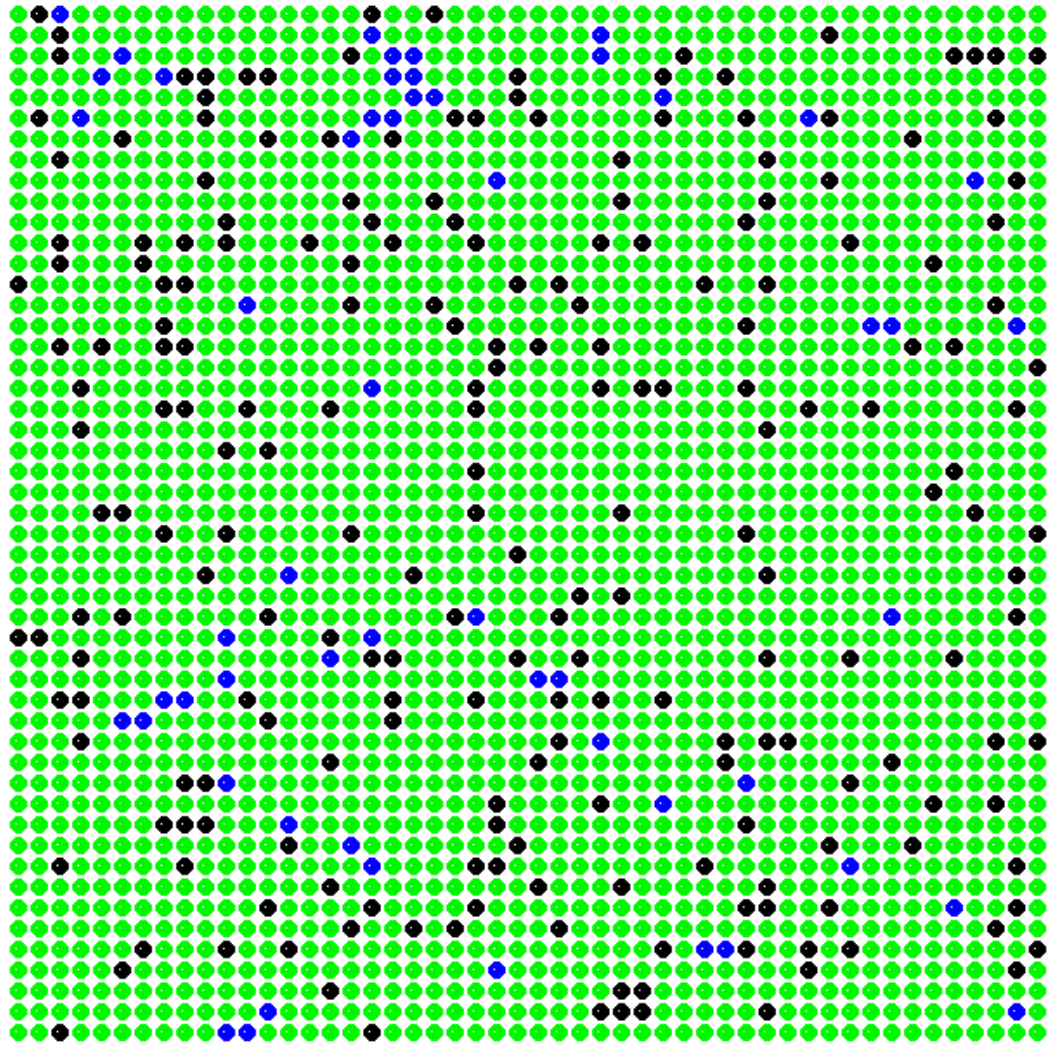
Verde Recuperado

Azul No infectado

Rojo Infectado

Negro Muerto

Total muertes: 244



Practica

En consecuencia, 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 permite simular la taza de contagio en una epidemia, aplicar a los datos del Ecuador y obtener un RO con los datos del pais.

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

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

Hasta el momento, el Instituto Nacional de Investigación en Salud Pública (INSPI) ha tomado 662.183 muestras para RT-PCR COVID-19 de las cuales 195.884 son casos confirmados con pruebas PCR.

Este indicador, de actualización diaria, reporta el número acumulado de las muestras tomadas para la realización de la prueba RT-PCR en los laboratorios autorizados en Ecuador. Cabe indicar que puede existir más de una muestra por persona durante el proceso diagnóstico.

- **169.804** pacientes recuperados.
- **23.748** casos con alta hospitalaria.
- **9.129** personas fallecidas (confirmados COVID-19)
- **541** hospitalizados estables.
- **350** hospitalizados con pronóstico reservado.
- **426.070** casos fueron descartados.
- **877.537** Llamadas al 171 relacionadas a COVID-19
- **350.921** Seguimiento telefónico, llamadas a pacientes con diagnóstico confirmado.
- **117.654** Teleconsultas, ciudadanos atendidos a través de APP SALUDEDEC y por un médico del 171.

El 29 de febrero de 2020 se confirmó el primer caso de coronavirus. El 13/03/2020 se activó el COE Nacional para la coordinación de la emergencia.

<https://www.salud.gob.ec/actualizacion-de-casos-de-coronavirus-en-ecuador/>

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

```
In [ ]: #Implementar
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
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() #Inicializo el tipo de letra
display=pygame.display.set_mode((800,750),0,32) #Tamano 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 veri
                        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 cercan

```

```

        neighbour = get_vecinos(x, y) #Obtenemos los vecinos a conta
        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,
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),
        pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1,
#Escuchar los eventos del teclado
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona
        pygame.quit() #Termino simulacion
    if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y
        #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

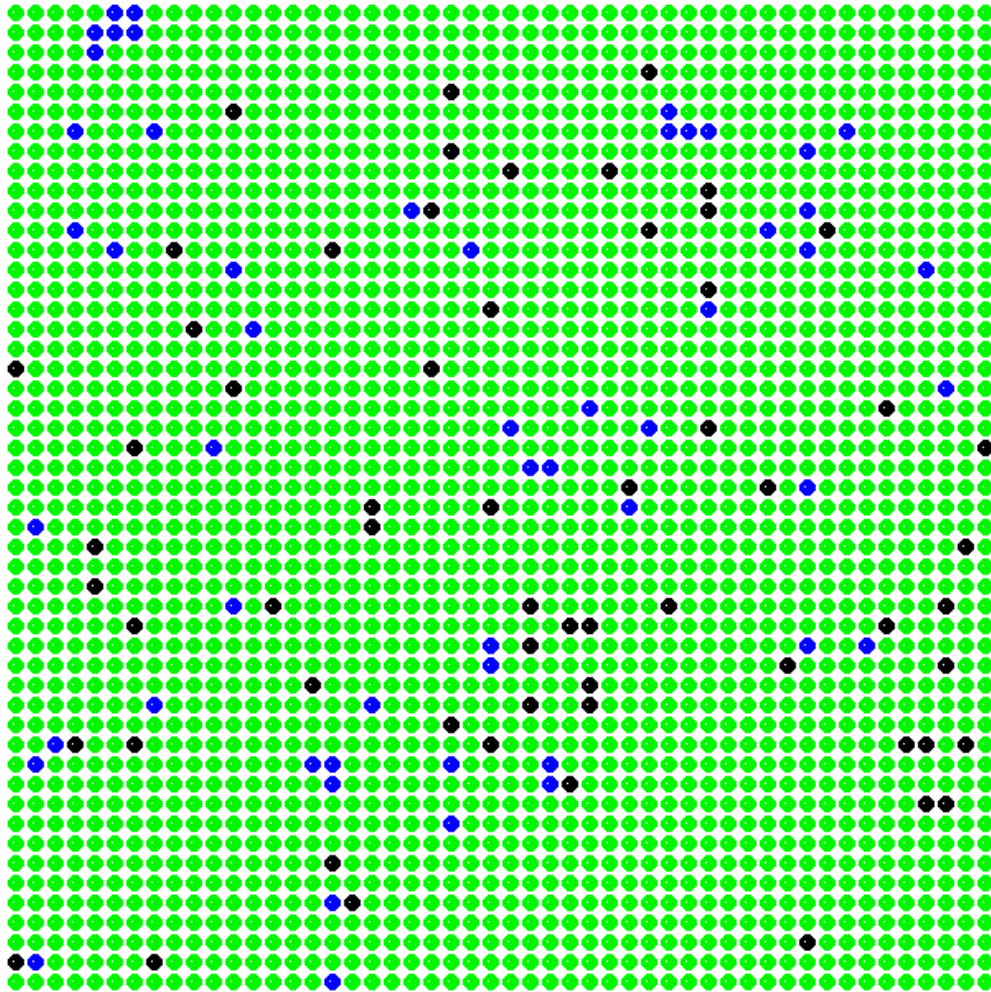
pygame.display.update()# Mandar actualizar la ventana

```

pygame 2.0.0 (SDL 2.0.12, python 3.8.5)

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

Total muertes: 64



3. El valor 1.4 en el mejor de los casos

```
In [ ]: #Implementar
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 = 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
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() #Inicializo el tipo de letra
display=pygame.display.set_mode((800,750),0,32) #Tamano 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 veri
                        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 cercan
                neighbour = get_vecinos(x, y) #Obtenemos los vecinos a conta
                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,
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),
        pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1,
#Escuchar los eventos del teclado
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona
        pygame.quit() #Termino simulacion
    if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y
        #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

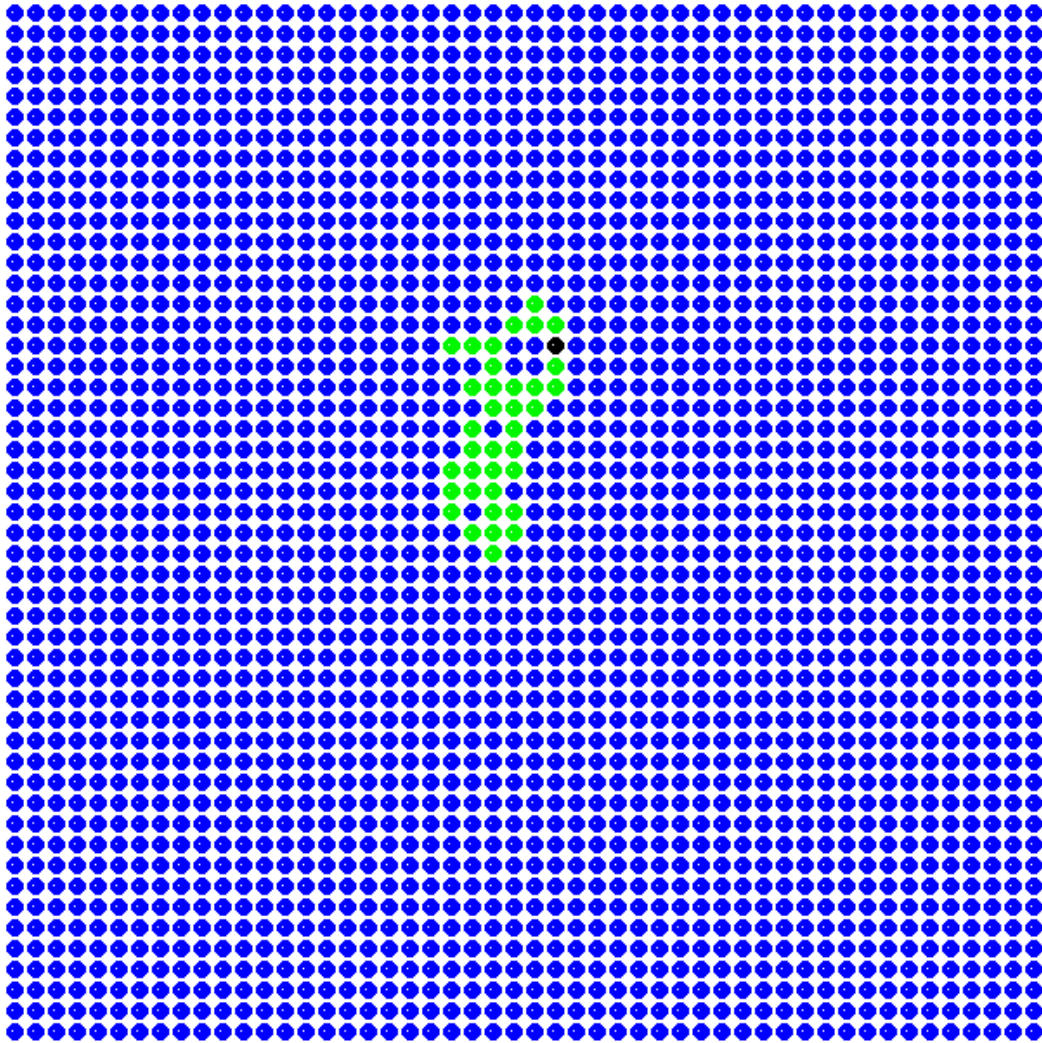
pygame.display.update()# Mandar actualizar la ventana

```

pygame 2.0.0 (SDL 2.0.12, python 3.8.5)

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

Total muertes: 1



4. Revisar e investigar algun tipo de software que permite simular la tasa de contagio en una epidemia, aplicar a los datos del Ecuador y obtener un R_0 con los datos del pais.

No encuentre ningun software que me permita simular la tasa de contagio y obtener el valor de valor R_0 .

Analisis

Como se puede observar cada grafica cambia encada simulación realizada, esto varía debido al valor de R_0 , ya que si se da un avalor bajo para R_0 los casos de muerte son bajos pero si se diera un valor alto, los casos de muerte serpien altos, en este caso se establecieron los valores de 4 que representa elpeor de los casos y el 1.4 que representa el mejor de los casos.

Conclusiones

Como conclusión tenemos que R_0 permite tener promedio de los casos nuevos que genera un caso dado, es decir la cantidad de personas que pueden infectarse a partir de una persona ya

infectada en este caso con covid-19.

Opinion

La simulación es una herramienta muy importante ya que nos permite tener un promedio de personas que se pueden infectar en un tiempo a partir de otras personas que ya tengan una infección y que hayan tenido contacto con personas sanas, en este caso no se puede tener datos certeros ya que hay personas que no informan sobre su contagio y pueden infectar a otras personas por lo que no hay estadísticas que nos den una información correcta.

Referencias

- <http://code.intef.es/simulamos-una-epidemia-virica/>
- <https://www.salud.gob.ec/actualizacion-de-casos-de-coronavirus-en-ecuador/>