NOMBRE: ALEX BENAVIDEZ

CARRERA: INGENIERIA EN SISTEMAS

MATERIA: SIMULACION

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 R0. 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 R0 es fundamental, porque si es grande, el contagio se alarga más rápidamente. Si R0 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 R0 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/)

Entonces, el número reproductivo (R0): 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). Ademas segun estimaciones de la OMS la probabilidad de fallecimiento es de 1.2% - 4.2% segun 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)

H In [1]:

1 pip install pygame

Requirement already satisfied: pygame in c:\users\alex\anaconda3\envs\alex\l

ib\site-packages (2.0.0)

Note: you may need to restart the kernel to use updated packages.

```
from random import randrange # Obtener un numero randomico
 1
   import pygame
 3
 4
   #Parametros de inicio
   PROBA_MUERTE = 8.4 # Probabilidad de que la gente muera COVID
 5
   CONTAGION RATE = 4.0 # Factor R0 para la simulación COVID probabilidad
 7
   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)
 9
   nb rows = 50 #Numero de filas
10
   nb_cols = 50 #Numero de columnas
11
12
   global display, myfont, states, states_temp #Declaracion de variables globales
13
14
   #Declaro colores en formato RGB
15
   WHITE = (255, 255, 255)
16
17
   BLUE = (0, 0, 255)
   GREEN = (0, 247, 0)
18
19
   BLACK = (0, 0, 0)
20
   #Obtiene los vecinos dado un punto x,y
21
22
   def get_vecinos(x, y):
23
       incx = randrange(3)
       incy = randrange(3)
24
25
       incx = (incx * 1) - 1
26
       incy = (incy * 1) - 1
27
       x2 = x + incx
28
       y2 = y + incy
29
       #Validar limites
30
       if x2 < 0:
            x2 = 0
31
       if x2 >= nb_cols:
32
33
            x2 = nb_cols - 1
34
       if y2 < 0:
35
           y2 = 0
36
       if y2 >= nb_rows:
37
            y2 = nb rows - 1
38
       return [x2, y2] # Nuevos contagiados
39
40
   #Genero las personas que cuentan con inmunidad o vacuna
   def vacunar():
41
42
       for x in range(nb cols):
43
            for y in range(nb_rows):
44
                if randrange(99) < PROBA_VACU:</pre>
45
                    states[x][y] = 1
46
47
   #Funcion que permite contar el numero de muertosde la matriz states == -1
   def contar muertes():
48
49
       contador = 0
50
       for x in range(nb_cols):
51
            for y in range(nb rows):
52
                if states[x][y] == -1:
53
                    contador += 1
54
       return contador
55
56
   #Definimos datos de inicio
57
   states = [[0] * nb_cols for i1 in range(nb_rows)]
58
   states temp = states.copy()
    states[randrange(50)][randrange(50)] = 10 \# Estado inicial de la simulacion Posicion d
```

```
60
    it = 0 # Variable para contar las Iteraciones
 61
    total muerte = 0 # Contabiliza el numero de muertos
    vacunar() #Llamar a la funcion vacunar
 62
 63
 64
    pygame.init() #Incializo el motor de juegos pygame
    pygame.font.init() #Inicializo el tipo de letra
 65
    display=pygame.display.set_mode((800,750),0,32) #Tamanio de la ventana
 66
    pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
 67
    font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
 68
    display.fill(WHITE) # Color de fondo
 69
 70
    while True:
 71
 72
         pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
 73
         it = it + 1
 74
         if it <= 10000 and it >= 2:
 75
             states_temp = states.copy() #Copia de la matriz
 76
             #Recorrera la matriz
 77
             for x in range(nb cols):
 78
                 for y in range(nb_rows):
 79
                     state = states[x][y]
                     if state == -1:
 80
 81
                         pass
                     if state >= 10: # Numero de dias de contagio
 82
                         states\_temp[x][y] = state + 1
 83
 84
                     if state >= 20:
                         if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi</pre>
 85
 86
                              states_temp[x][y] = -1 # Muere
 87
                         else:
 88
                              states_temp[x][y] = 1 # Cura o recupera
                     if state >= 10 and state <= 20: # Rango de infectado</pre>
 89
                         if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas</pre>
 90
 91
                              neighbour = get_vecinos(x, y) #Obtenemos Los vecinos a contagi
 92
                             x2 = neighbour[0]
 93
                             y2 = neighbour[1]
 94
                             neigh_state = states[x2][y2]
                              if neigh_state == 0: #Verifico que este sano
 95
 96
                                  states_temp[x2][y2] = 10 # Contagia
 97
             states = states_temp.copy()
             total_muerte = contar_muertes() # contar el numero de muertos
 98
 99
         pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
100
         textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,12
101
         display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
102
103
         #Graficar el estado del paciente matriz
         for x in range(nb cols):
104
             for y in range(nb_rows):
105
106
                 if states[x][y] == 0:
                     color = BLUE # No infectado
107
                 if states[x][y] == 1:
108
                     color = GREEN # Recupero
109
110
                 if states[x][y] \Rightarrow= 10:
                     color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
111
112
                 if states[x][y] == -1:
                     color = BLACK # Muerto
113
                 pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
114
                 pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1)
115
116
         #Escuachar los eventos del teclado
117
         for event in pygame.event.get():
118
             if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: #Presiona y
119
                 pygame.quit() #Termino simulacion
120
             if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y e
```

```
121
                 #Reiniciamos valores
122
                 states = [[0] * nb_cols for i1 in range(nb_rows)]
123
                 states temp = states.copy()
                 states[5][5] = 10
124
                 it = 0
125
                 total_muerte = 0
126
127
                 vacunar() #Llamar a la funcion vacunar
128
129
        pygame.display.update()# Mandar actualizar La ventana
```

```
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html (htt
ps://www.pygame.org/contribute.html)
```

1. Investigar el valor de varianza R0 en Ecuador

- El valor de varianza de R0 en Ecuador en este mes es: 0.59 1.10
- La probabilidad de muerte es: 7.1%

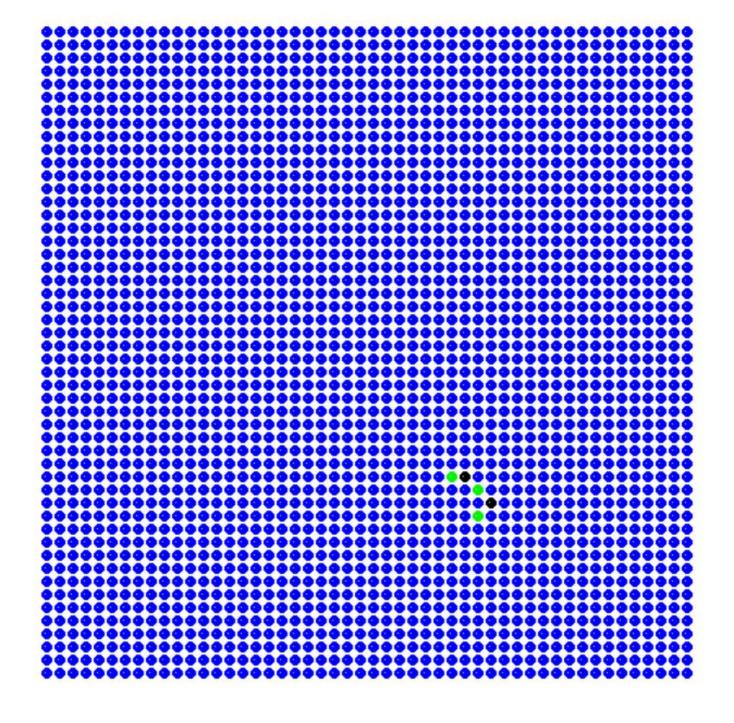
R0 = 0.59

```
1 | from random import randrange # Obtener un numero randomico
   import pygame
 3 #Parametros de inicio
 4 PROBA MUERTE = 7.1 # Probabilidad de que la gente muera COVID
  CONTAGION_RATE = 0.59 # Factor R0 para La simulacion COVID probabilidad
 5
 6 PROBA INFECT = CONTAGION RATE * 10
 7
   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
 9
   nb cols = 50 #Numero de columnas
10
11
12
   global display, myfont, states, states_temp #Declaracion de variables globales
13
14
   #Declaro colores en formato RGB
15
   WHITE = (255, 255, 255)
   BLUE = (0, 0, 255)
16
   GREEN = (0, 247, 0)
17
   BLACK = (0, 0, 0)
18
19
20
   #Obtiene los vecinos dado un punto x,y
   def get_vecinos(x, y):
21
22
       incx = randrange(3)
23
       incy = randrange(3)
       incx = (incx * 1) - 1
24
       incy = (incy * 1) - 1
25
26
       x2 = x + incx
27
       y2 = y + incy
28
       #Validar limites
29
       if x2 < 0:
30
            x2 = 0
       if x2 >= nb_cols:
31
32
           x2 = nb_cols - 1
33
       if y2 < 0:
           y2 = 0
34
35
        if y2 >= nb_rows:
36
            y2 = nb_rows - 1
37
        return [x2, y2] # Nuevos contagiados
38
39
   #Genero las personas que cuentan con inmunidad o vacuna
40
   def vacunar():
41
       for x in range(nb cols):
42
            for y in range(nb_rows):
43
                if randrange(99) < PROBA_VACU:</pre>
44
                    states[x][y] = 1
45
   #Funcion que permite contar el numero de muertosde la matriz states == -1
46
47
   def contar muertes():
48
        contador = 0
49
        for x in range(nb_cols):
50
            for y in range(nb_rows):
51
                if states[x][y] == -1:
52
                    contador +=
53
       return contador
54
55
   #Definimos datos de inicio
56
   states = [[0] * nb_cols for i1 in range(nb_rows)]
57
   states_temp = states.copy()
58
   states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion d
    it = 0 # Variable para contar las Iteraciones
```

```
total muerte = 0 # Contabiliza el numero de muertos
 60
 61
    vacunar() #Llamar a la funcion vacunar
 62
    pygame.init() #Incializo el motor de juegos pygame
 63
 64
    pygame.font.init() #Inicializo el tipo de letra
    display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
 65
    pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
 66
    font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
 67
    display.fill(WHITE) # Color de fondo
 68
 69
 70
    while True:
         pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
 71
 72
         it = it + 1
 73
         if it <= 10000 and it >= 2:
 74
             states_temp = states.copy() #Copia de La matriz
 75
             #Recorrera la matriz
 76
             for x in range(nb_cols):
                 for y in range(nb_rows):
 77
 78
                     state = states[x][y]
                     if state == -1:
 79
 80
                         pass
 81
                     if state >= 10: # Numero de dias de contagio
                         states_temp[x][y] = state + 1
 82
 83
                     if state >= 20:
                         if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi</pre>
 84
 85
                              states_temp[x][y] = -1 # Muere
 86
                         else:
                              states_temp[x][y] = 1 # Cura o recupera
 87
 88
                     if state >= 10 and state <= 20: # Rango de infectado</pre>
                         if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas
 89
                             neighbour = get_vecinos(x, y) #Obtenemos Los vecinos a contagi
 90
 91
                             x2 = neighbour[0]
 92
                             y2 = neighbour[1]
 93
                             neigh state = states[x2][y2]
 94
                             if neigh_state == 0: #Verifico que este sano
                                  states_temp[x2][y2] = 10 # Contagia
 95
 96
             states = states_temp.copy()
 97
             total_muerte = contar_muertes() # contar el numero de muertos
 98
 99
         pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
         textsurface = font.render("Total muertes: "+ str(total muerte), False, (255,160,12
100
         display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
101
         #Graficar el estado del paciente matriz
102
103
         for x in range(nb_cols):
104
             for y in range(nb rows):
                 if states[x][y] == 0:
105
                     color = BLUE # No infectado
106
                 if states[x][y] == 1:
107
                     color = GREEN # Recupero
108
                 if states[x][y] >= 10:
109
110
                     color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
111
                 if states[x][y] == -1:
                     color = BLACK # Muerto
112
                 pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
113
114
                 pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1)
         #Escuachar los eventos del teclado
115
116
         for event in pygame.event.get():
117
             if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y
118
                 pygame.quit() #Termino simulacion
119
             if event.type == pygame.KEYDOWN and event.key == pygame.K SPACE: #Presiona y e
120
                 #Reiniciamos valores
```

```
states = [[0] * nb_cols for i1 in range(nb_rows)]
121
122
                 states_temp = states.copy()
                 states[5][5] = 10
123
                 it = 0
124
                 total_muerte = 0
125
126
                 vacunar() #Llamar a la funcion vacunar
127
128
        pygame.display.update()# Mandar actualizar la ventana
```

```
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html (htt
ps://www.pygame.org/contribute.html)
```



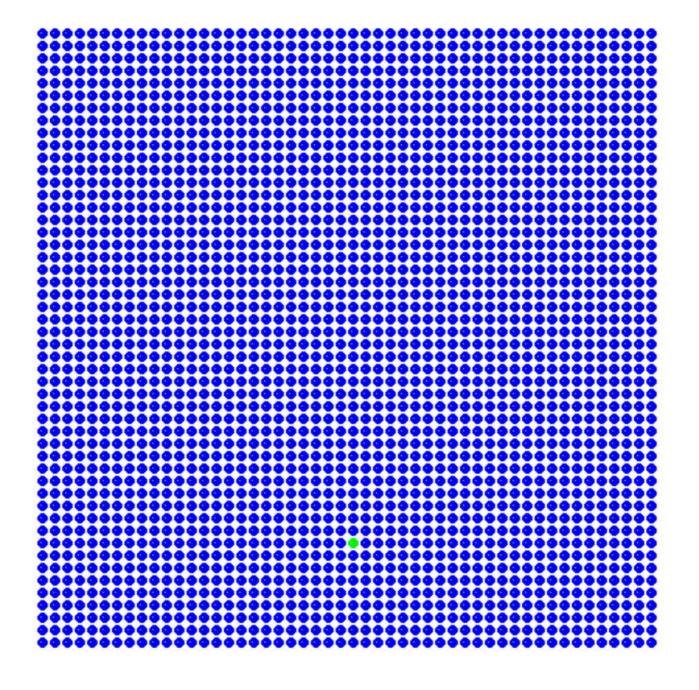
R0 = 1.10

```
1 | from random import randrange # Obtener un numero randomico
   import pygame
 3 #Parametros de inicio
 4 PROBA MUERTE = 7.1 # Probabilidad de que la gente muera COVID
 5 CONTAGION_RATE = 1.10 # Factor R0 para La simulacion COVID probabilidad
 6 PROBA INFECT = CONTAGION RATE * 10
 7
   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
 9
   nb cols = 50 #Numero de columnas
10
11
12
   global display, myfont, states, states_temp #Declaracion de variables globales
13
14
   #Declaro colores en formato RGB
15
   WHITE = (255, 255, 255)
   BLUE = (0, 0, 255)
16
   GREEN = (0, 247, 0)
17
   BLACK = (0, 0, 0)
18
19
20
   #Obtiene los vecinos dado un punto x,y
   def get_vecinos(x, y):
21
22
       incx = randrange(3)
23
       incy = randrange(3)
       incx = (incx * 1) - 1
24
25
       incy = (incy * 1) - 1
26
       x2 = x + incx
27
       y2 = y + incy
28
       #Validar limites
29
       if x2 < 0:
30
            x2 = 0
       if x2 >= nb_cols:
31
32
           x2 = nb_cols - 1
33
       if y2 < 0:
           y2 = 0
34
35
        if y2 >= nb_rows:
36
            y2 = nb_rows - 1
37
        return [x2, y2] # Nuevos contagiados
38
39
   #Genero las personas que cuentan con inmunidad o vacuna
40
   def vacunar():
41
       for x in range(nb cols):
42
            for y in range(nb_rows):
43
                if randrange(99) < PROBA_VACU:</pre>
44
                    states[x][y] = 1
45
   #Funcion que permite contar el numero de muertosde la matriz states == -1
46
47
   def contar muertes():
48
        contador = 0
49
        for x in range(nb_cols):
50
            for y in range(nb_rows):
51
                if states[x][y] == -1:
52
                    contador +=
53
       return contador
54
55
   #Definimos datos de inicio
56
   states = [[0] * nb_cols for i1 in range(nb_rows)]
57
   states_temp = states.copy()
58
   states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion d
    it = 0 # Variable para contar las Iteraciones
```

```
total muerte = 0 # Contabiliza el numero de muertos
 60
 61
    vacunar() #Llamar a la funcion vacunar
 62
    pygame.init() #Incializo el motor de juegos pygame
 63
 64
    pygame.font.init() #Inicializo el tipo de letra
    display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
 65
    pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
 66
    font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
 67
    display.fill(WHITE) # Color de fondo
 68
 69
 70
    while True:
         pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
 71
 72
         it = it + 1
 73
         if it <= 10000 and it >= 2:
 74
             states_temp = states.copy() #Copia de La matriz
 75
             #Recorrera la matriz
 76
             for x in range(nb_cols):
                 for y in range(nb_rows):
 77
 78
                     state = states[x][y]
                     if state == -1:
 79
 80
                         pass
 81
                     if state >= 10: # Numero de dias de contagio
                         states_temp[x][y] = state + 1
 82
 83
                     if state >= 20:
                         if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi</pre>
 84
 85
                              states_temp[x][y] = -1 # Muere
 86
                         else:
                              states_temp[x][y] = 1 # Cura o recupera
 87
 88
                     if state >= 10 and state <= 20: # Rango de infectado</pre>
                         if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas
 89
                             neighbour = get_vecinos(x, y) #Obtenemos Los vecinos a contagi
 90
 91
                             x2 = neighbour[0]
 92
                             y2 = neighbour[1]
 93
                             neigh state = states[x2][y2]
 94
                             if neigh_state == 0: #Verifico que este sano
                                  states_temp[x2][y2] = 10 # Contagia
 95
 96
             states = states_temp.copy()
 97
             total_muerte = contar_muertes() # contar el numero de muertos
 98
 99
         pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
         textsurface = font.render("Total muertes: "+ str(total muerte), False, (255,160,12
100
         display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
101
         #Graficar el estado del paciente matriz
102
103
         for x in range(nb_cols):
104
             for y in range(nb rows):
                 if states[x][y] == 0:
105
                     color = BLUE # No infectado
106
                 if states[x][y] == 1:
107
                     color = GREEN # Recupero
108
                 if states[x][y] >= 10:
109
110
                     color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
111
                 if states[x][y] == -1:
                     color = BLACK # Muerto
112
                 pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
113
114
                 pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1)
         #Escuachar los eventos del teclado
115
116
         for event in pygame.event.get():
117
             if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y
118
                 pygame.quit() #Termino simulacion
119
             if event.type == pygame.KEYDOWN and event.key == pygame.K SPACE: #Presiona y e
120
                 #Reiniciamos valores
```

```
states = [[0] * nb_cols for i1 in range(nb_rows)]
121
122
                 states_temp = states.copy()
                 states[5][5] = 10
123
                 it = 0
124
125
                 total_muerte = 0
126
                 vacunar() #Llamar a la funcion vacunar
127
128
        pygame.display.update()# Mandar actualizar la ventana
```

```
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html (htt
ps://www.pygame.org/contribute.html)
```



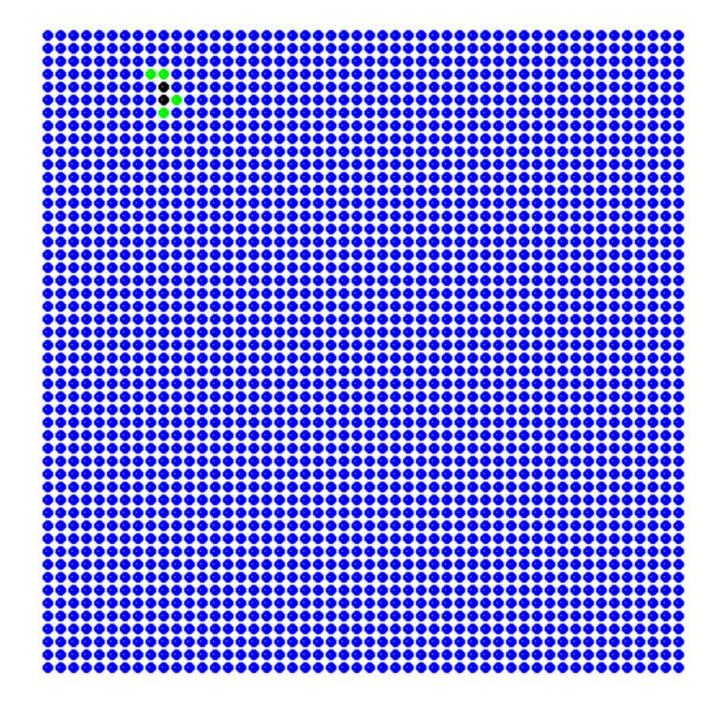
7/12/2020

```
1 | from random import randrange # Obtener un numero randomico
   import pygame
 3 #Parametros de inicio
 4 PROBA MUERTE = 7.1 # Probabilidad de que la gente muera COVID
  CONTAGION_RATE = 0.80 # Factor R0 para La simulacion COVID probabilidad
 5
 6 PROBA INFECT = CONTAGION RATE * 10
 7
   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
 9
   nb cols = 50 #Numero de columnas
10
11
12
   global display, myfont, states, states_temp #Declaracion de variables globales
13
14
   #Declaro colores en formato RGB
15
   WHITE = (255, 255, 255)
   BLUE = (0, 0, 255)
16
   GREEN = (0, 247, 0)
17
   BLACK = (0, 0, 0)
18
19
20
   #Obtiene los vecinos dado un punto x,y
   def get_vecinos(x, y):
21
22
       incx = randrange(3)
23
       incy = randrange(3)
       incx = (incx * 1) - 1
24
       incy = (incy * 1) - 1
25
26
       x2 = x + incx
27
       y2 = y + incy
28
       #Validar limites
29
       if x2 < 0:
30
            x2 = 0
       if x2 >= nb_cols:
31
32
           x2 = nb_cols - 1
33
       if y2 < 0:
           y2 = 0
34
35
        if y2 >= nb_rows:
36
            y2 = nb_rows - 1
37
        return [x2, y2] # Nuevos contagiados
38
39
   #Genero las personas que cuentan con inmunidad o vacuna
40
   def vacunar():
41
       for x in range(nb cols):
42
            for y in range(nb_rows):
43
                if randrange(99) < PROBA_VACU:</pre>
44
                    states[x][y] = 1
45
   #Funcion que permite contar el numero de muertosde la matriz states == -1
46
47
   def contar muertes():
48
        contador = 0
49
        for x in range(nb_cols):
50
            for y in range(nb_rows):
51
                if states[x][y] == -1:
52
                    contador +=
53
       return contador
54
55
   #Definimos datos de inicio
56
   states = [[0] * nb_cols for i1 in range(nb_rows)]
57
   states_temp = states.copy()
58
   states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion d
    it = 0 # Variable para contar las Iteraciones
```

```
total muerte = 0 # Contabiliza el numero de muertos
 60
 61
    vacunar() #Llamar a la funcion vacunar
 62
    pygame.init() #Incializo el motor de juegos pygame
 63
 64
    pygame.font.init() #Inicializo el tipo de letra
    display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
 65
    pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
 66
    font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
 67
    display.fill(WHITE) # Color de fondo
 68
 69
 70
    while True:
         pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
 71
 72
         it = it + 1
 73
         if it <= 10000 and it >= 2:
 74
             states_temp = states.copy() #Copia de La matriz
 75
             #Recorrera la matriz
 76
             for x in range(nb_cols):
                 for y in range(nb_rows):
 77
 78
                     state = states[x][y]
                     if state == -1:
 79
 80
                         pass
 81
                     if state >= 10: # Numero de dias de contagio
                         states_temp[x][y] = state + 1
 82
 83
                     if state >= 20:
                         if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi</pre>
 84
 85
                              states_temp[x][y] = -1 # Muere
 86
                         else:
                              states_temp[x][y] = 1 # Cura o recupera
 87
 88
                     if state >= 10 and state <= 20: # Rango de infectado</pre>
                         if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas
 89
                             neighbour = get_vecinos(x, y) #Obtenemos Los vecinos a contagi
 90
 91
                             x2 = neighbour[0]
 92
                             y2 = neighbour[1]
 93
                             neigh state = states[x2][y2]
 94
                             if neigh_state == 0: #Verifico que este sano
                                  states_temp[x2][y2] = 10 # Contagia
 95
 96
             states = states_temp.copy()
 97
             total_muerte = contar_muertes() # contar el numero de muertos
 98
 99
         pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
         textsurface = font.render("Total muertes: "+ str(total muerte), False, (255,160,12
100
         display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
101
         #Graficar el estado del paciente matriz
102
103
         for x in range(nb_cols):
104
             for y in range(nb rows):
                 if states[x][y] == 0:
105
                     color = BLUE # No infectado
106
                 if states[x][y] == 1:
107
                     color = GREEN # Recupero
108
                 if states[x][y] >= 10:
109
110
                     color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
111
                 if states[x][y] == -1:
                     color = BLACK # Muerto
112
                 pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
113
114
                 pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1)
         #Escuachar los eventos del teclado
115
116
         for event in pygame.event.get():
117
             if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y
118
                 pygame.quit() #Termino simulacion
119
             if event.type == pygame.KEYDOWN and event.key == pygame.K SPACE: #Presiona y e
120
                 #Reiniciamos valores
```

```
states = [[0] * nb_cols for i1 in range(nb_rows)]
121
122
                 states_temp = states.copy()
                 states[5][5] = 10
123
124
                 it = 0
                 total_muerte = 0
125
126
                 vacunar() #Llamar a la funcion vacunar
127
128
        pygame.display.update()# Mandar actualizar la ventana
```

```
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html (htt
ps://www.pygame.org/contribute.html)
```



2. Valor 4, para el peor de los casos.

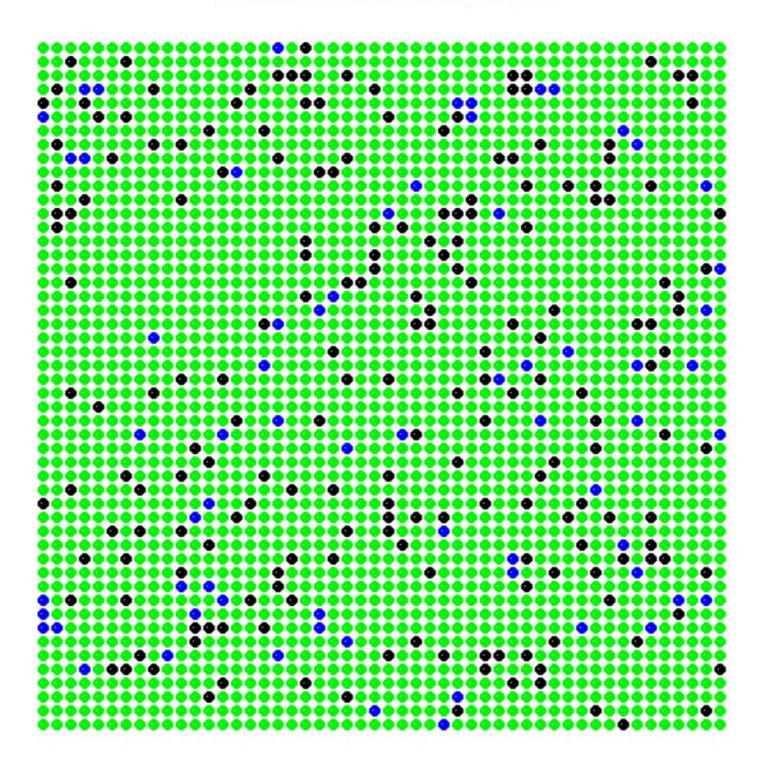
P0 = 4

```
1 | from random import randrange # Obtener un numero randomico
   import pygame
 3 #Parametros de inicio
 4 PROBA MUERTE = 7.1 # Probabilidad de que la gente muera COVID
  CONTAGION_RATE = 4 # Factor R0 para La simulacion COVID probabilidad
 5
 6 PROBA INFECT = CONTAGION RATE * 10
 7
   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
 9
   nb cols = 50 #Numero de columnas
10
11
12
   global display, myfont, states, states_temp #Declaracion de variables globales
13
14
   #Declaro colores en formato RGB
15
   WHITE = (255, 255, 255)
   BLUE = (0, 0, 255)
16
   GREEN = (0, 247, 0)
17
   BLACK = (0, 0, 0)
18
19
20
   #Obtiene los vecinos dado un punto x,y
   def get_vecinos(x, y):
21
22
       incx = randrange(3)
23
       incy = randrange(3)
       incx = (incx * 1) - 1
24
       incy = (incy * 1) - 1
25
26
       x2 = x + incx
27
       y2 = y + incy
28
       #Validar limites
29
       if x2 < 0:
30
            x2 = 0
       if x2 >= nb_cols:
31
32
           x2 = nb_cols - 1
33
       if y2 < 0:
            y2 = 0
34
35
        if y2 >= nb_rows:
36
            y2 = nb_rows - 1
37
        return [x2, y2] # Nuevos contagiados
38
39
   #Genero las personas que cuentan con inmunidad o vacuna
40
   def vacunar():
41
       for x in range(nb cols):
42
            for y in range(nb_rows):
43
                if randrange(99) < PROBA_VACU:</pre>
44
                    states[x][y] = 1
45
   #Funcion que permite contar el numero de muertosde la matriz states == -1
46
47
   def contar muertes():
48
        contador = 0
49
        for x in range(nb_cols):
50
            for y in range(nb_rows):
51
                if states[x][y] == -1:
52
                    contador +=
53
       return contador
54
55
   #Definimos datos de inicio
56
   states = [[0] * nb_cols for i1 in range(nb_rows)]
57
   states_temp = states.copy()
58
   states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion d
    it = 0 # Variable para contar las Iteraciones
```

```
total muerte = 0 # Contabiliza el numero de muertos
 60
 61
    vacunar() #Llamar a la funcion vacunar
 62
    pygame.init() #Incializo el motor de juegos pygame
 63
 64
    pygame.font.init() #Inicializo el tipo de letra
    display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
 65
    pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
 66
    font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
 67
    display.fill(WHITE) # Color de fondo
 68
 69
 70
    while True:
         pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
 71
 72
         it = it + 1
 73
         if it <= 10000 and it >= 2:
 74
             states_temp = states.copy() #Copia de La matriz
 75
             #Recorrera la matriz
 76
             for x in range(nb_cols):
                 for y in range(nb_rows):
 77
 78
                     state = states[x][y]
                     if state == -1:
 79
 80
                         pass
 81
                     if state >= 10: # Numero de dias de contagio
                         states_temp[x][y] = state + 1
 82
 83
                     if state >= 20:
                         if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi</pre>
 84
 85
                              states_temp[x][y] = -1 # Muere
 86
                         else:
                              states_temp[x][y] = 1 # Cura o recupera
 87
 88
                     if state >= 10 and state <= 20: # Rango de infectado</pre>
                         if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas
 89
                             neighbour = get_vecinos(x, y) #Obtenemos Los vecinos a contagi
 90
 91
                             x2 = neighbour[0]
 92
                             y2 = neighbour[1]
 93
                             neigh state = states[x2][y2]
 94
                             if neigh_state == 0: #Verifico que este sano
                                  states_temp[x2][y2] = 10 # Contagia
 95
 96
             states = states_temp.copy()
 97
             total_muerte = contar_muertes() # contar el numero de muertos
 98
 99
         pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
         textsurface = font.render("Total muertes: "+ str(total muerte), False, (255,160,12
100
         display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
101
         #Graficar el estado del paciente matriz
102
103
         for x in range(nb_cols):
104
             for y in range(nb rows):
                 if states[x][y] == 0:
105
                     color = BLUE # No infectado
106
                 if states[x][y] == 1:
107
                     color = GREEN # Recupero
108
                 if states[x][y] >= 10:
109
110
                     color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
111
                 if states[x][y] == -1:
                     color = BLACK # Muerto
112
                 pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
113
114
                 pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1)
         #Escuachar los eventos del teclado
115
116
         for event in pygame.event.get():
117
             if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y
118
                 pygame.quit() #Termino simulacion
119
             if event.type == pygame.KEYDOWN and event.key == pygame.K SPACE: #Presiona y e
120
                 #Reiniciamos valores
```

```
states = [[0] * nb_cols for i1 in range(nb_rows)]
121
122
                 states_temp = states.copy()
123
                 states[5][5] = 10
124
                it = 0
                total_muerte = 0
125
                vacunar() #Llamar a la funcion vacunar
126
127
128
        pygame.display.update()# Mandar actualizar la ventana
```

```
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html (htt
ps://www.pygame.org/contribute.html)
```



3. Valor 1.4, en el mejor de los casos

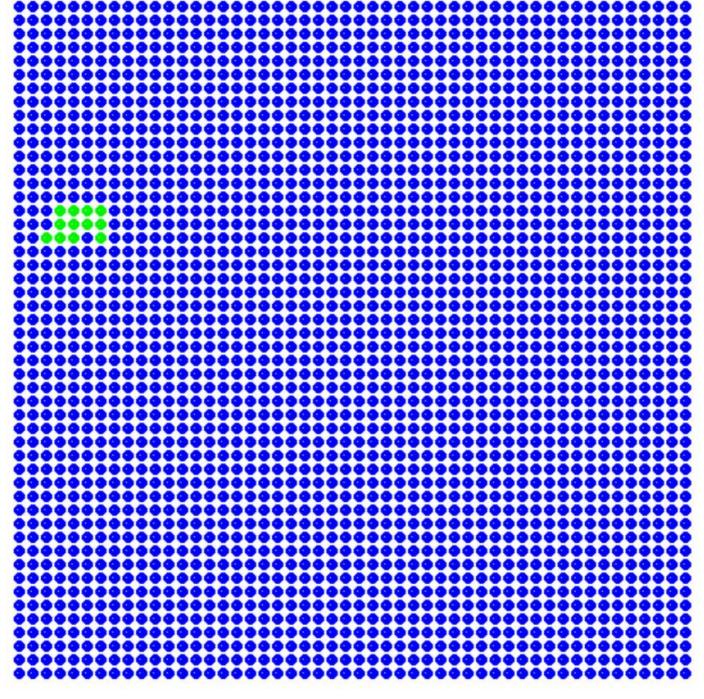
R0 = 1.4

```
1 | from random import randrange # Obtener un numero randomico
   import pygame
 3 #Parametros de inicio
 4 PROBA MUERTE = 7.1 # Probabilidad de que la gente muera COVID
 5 CONTAGION_RATE = 1.4 # Factor R0 para La simulacion COVID probabilidad
 6 PROBA INFECT = CONTAGION RATE * 10
 7
   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
 9
   nb cols = 50 #Numero de columnas
10
11
12
   global display, myfont, states, states_temp #Declaracion de variables globales
13
14
   #Declaro colores en formato RGB
15
   WHITE = (255, 255, 255)
   BLUE = (0, 0, 255)
16
   GREEN = (0, 247, 0)
17
   BLACK = (0, 0, 0)
18
19
20
   #Obtiene los vecinos dado un punto x,y
   def get_vecinos(x, y):
21
22
       incx = randrange(3)
23
       incy = randrange(3)
       incx = (incx * 1) - 1
24
25
       incy = (incy * 1) - 1
26
       x2 = x + incx
27
       y2 = y + incy
28
       #Validar limites
29
       if x2 < 0:
30
            x2 = 0
       if x2 >= nb_cols:
31
32
           x2 = nb_cols - 1
33
       if y2 < 0:
           y2 = 0
34
35
        if y2 >= nb_rows:
36
            y2 = nb_rows - 1
37
        return [x2, y2] # Nuevos contagiados
38
39
   #Genero las personas que cuentan con inmunidad o vacuna
40
   def vacunar():
41
       for x in range(nb cols):
42
            for y in range(nb_rows):
43
                if randrange(99) < PROBA_VACU:</pre>
44
                    states[x][y] = 1
45
   #Funcion que permite contar el numero de muertosde la matriz states == -1
46
47
   def contar muertes():
48
        contador = 0
49
        for x in range(nb_cols):
50
            for y in range(nb_rows):
51
                if states[x][y] == -1:
52
                    contador += 1
53
       return contador
54
55
   #Definimos datos de inicio
56
   states = [[0] * nb_cols for i1 in range(nb_rows)]
57
   states_temp = states.copy()
58
   states[randrange(50)][randrange(50)] = 10 # Estado inicial de la simulacion Posicion d
    it = 0 # Variable para contar las Iteraciones
```

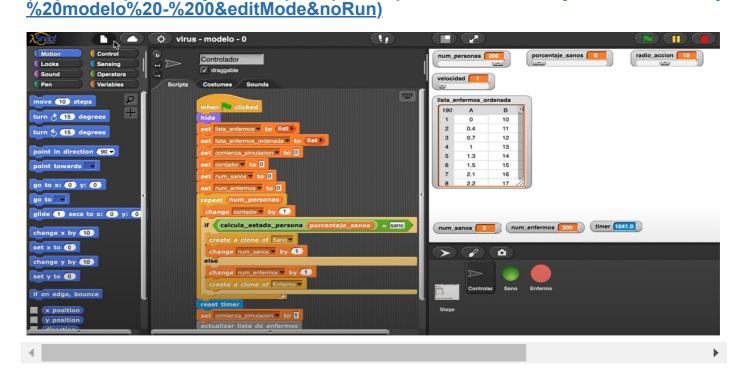
```
total muerte = 0 # Contabiliza el numero de muertos
 60
 61
    vacunar() #Llamar a la funcion vacunar
 62
    pygame.init() #Incializo el motor de juegos pygame
 63
 64
    pygame.font.init() #Inicializo el tipo de letra
    display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
 65
    pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
 66
    font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
 67
    display.fill(WHITE) # Color de fondo
 68
 69
 70
    while True:
         pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
 71
 72
         it = it + 1
 73
         if it <= 10000 and it >= 2:
 74
             states_temp = states.copy() #Copia de La matriz
 75
             #Recorrera la matriz
 76
             for x in range(nb_cols):
                 for y in range(nb_rows):
 77
 78
                     state = states[x][y]
                     if state == -1:
 79
 80
                         pass
 81
                     if state >= 10: # Numero de dias de contagio
                         states_temp[x][y] = state + 1
 82
 83
                     if state >= 20:
                         if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi</pre>
 84
 85
                              states_temp[x][y] = -1 # Muere
 86
                         else:
                              states_temp[x][y] = 1 # Cura o recupera
 87
 88
                     if state >= 10 and state <= 20: # Rango de infectado</pre>
                         if randrange(99) < PROBA_INFECT: # Infecto a las personas cercanas
 89
                             neighbour = get_vecinos(x, y) #Obtenemos Los vecinos a contagi
 90
 91
                             x2 = neighbour[0]
 92
                             y2 = neighbour[1]
 93
                             neigh state = states[x2][y2]
 94
                             if neigh_state == 0: #Verifico que este sano
                                  states_temp[x2][y2] = 10 # Contagia
 95
 96
             states = states_temp.copy()
 97
             total_muerte = contar_muertes() # contar el numero de muertos
 98
 99
         pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
         textsurface = font.render("Total muertes: "+ str(total muerte), False, (255,160,12
100
         display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
101
         #Graficar el estado del paciente matriz
102
103
         for x in range(nb_cols):
104
             for y in range(nb rows):
                 if states[x][y] == 0:
105
                     color = BLUE # No infectado
106
                 if states[x][y] == 1:
107
                     color = GREEN # Recupero
108
                 if states[x][y] >= 10:
109
110
                     color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
111
                 if states[x][y] == -1:
                     color = BLACK # Muerto
112
                 pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
113
114
                 pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1)
         #Escuachar los eventos del teclado
115
116
         for event in pygame.event.get():
117
             if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y
118
                 pygame.quit() #Termino simulacion
119
             if event.type == pygame.KEYDOWN and event.key == pygame.K SPACE: #Presiona y e
120
                 #Reiniciamos valores
```

```
states = [[0] * nb_cols for i1 in range(nb_rows)]
121
122
                 states_temp = states.copy()
                 states[5][5] = 10
123
                 it = 0
124
                 total_muerte = 0
125
126
                 vacunar() #Llamar a la funcion vacunar
127
         pygame.display.update()# Mandar actualizar la ventana
128
```

```
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html (htt
ps://www.pygame.org/contribute.html)
```



- 4. Revisar e investigar algún tipo de software que permita simular la tasa de contagio en una epidemia, aplicar a los datos de Ecuador y obtener un R0 con los datos del país.
- No pude encontrar un software que permita obtener el valor de R0 pero encontre otro software que permite simular una pandemia per utilizando otros valores como parametro como el porcentaje de sanos, el numero total de personas y por ultimo el radio de accion.
- Aqui ejemplo de uso: https://www.youtube.com/watch?v=- bl01ncbCJI&feature=emb_logo (https://www.youtube.com/watch?v=bl01ncbCJl&feature=emb_logo)
- Link de acceso al software de simulacion: https://snap.berkeley.edu/snap/snap.html#present:Username=juandali %20modelo%20-%200&editMode&noRun (https://snap.berkeley.edu/snap/snap.html#present:Username=juandalibaba&Proje



Analisis

 Con los diferentes valores de varianza se ha comprovado que el numero de muertos aumenta o disminuye considerando estos valores, cuanto mas bajo sea este valor menor sera el numero de muertes

Conclusiones

Mediante la libreria simpy se ha realizado las respetivas simulaciones, utilizando valores diferentes para poder ver como varia el resultado de la simulacion.

Opinion

 La simulacion ha demostrado que es una tecnologia muy importante hoy en dia, y mas cuando estamos pasando por una etapa de pandemia en donde la simulacion ha sido importante para poder determinar de cierta manera como perjudica esta pandemia utilizando datos reales.

Referencias

- http://code.intef.es/simulamos-una-epidemia-virica/ (http://code.intef.es/simulamos-una-epidemia-virica/)
- https://www.youtube.com/watch?v=-bI01ncbCJI&feature=emb_logo (https://www.youtube.com/watch?v=bl01ncbCJI&feature=emb_logo)