

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, de media, 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/>

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 [ ]: 1 from random import randrange
2 import pygame
3
4 #Parametros de inicio
5 PROBA_MUERTE = 4 # Probabilidad de que la gente muera COVID
6 CONTAGION_RATE = 56.43 # Factor R0 para la simulacion COVID probabilidad
7 PROBA_INFECT = CONTAGION_RATE * 10
8 PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
9 SIMULACION_SPEED = 25 # Tiempo de un dia en milisegundos
10 nb_rows = 100 #Numero de filas
11 nb_cols = 100 #Numero de columnas
12
13 global display, myfont, states, states_temp
14
15 WHITE = (255, 255, 255)
16 BLUE = (0, 0, 255)
17 GREEN = (0, 247, 0)
18 BLACK = (0, 0, 0)
19
20 def get_vecinos(x, y):
21     incx = randrange(3)
22     incy = randrange(3)
23     incx = (incx * 1) - 1
24     incy = (incy * 1) - 1
25     x2 = x + incx
26     y2 = y + incy
27     if x2 < 0:
28         x2 = 0
29     if x2 >= nb_cols:
30         x2 = nb_cols - 1
31     if y2 < 0:
32         y2 = 0
33     if y2 >= nb_rows:
34         y2 = nb_rows - 1
35     return [x2, y2]
36
37 def vacunar():
38     for x in range(nb_cols):
39         for y in range(nb_rows):
40             if randrange(99) < PROBA_VACU:
41                 states[x][y] = 1
42
43 def contar_muertes():
44     contador = 0
45     for x in range(nb_cols):
46         for y in range(nb_rows):
47             if states[x][y] == -1:
48                 contador += 1
49     return contador
50
51 #Definimos datos de inicio
52 states = [[0] * nb_cols for i1 in range(nb_rows)]
53 states_temp = [[0] * nb_cols for i1 in range(nb_rows)]
54 states_temp = states.copy()
55 states[5][5] = 10
56 it = 0 # Iteraciones
57 total_muerte = 0
58 vacunar()
59
60 pygame.init()
61 pygame.font.init()
62 display=pygame.display.set_mode((1500,1000),0,32)
63 pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")
64 font=pygame.font.SysFont('Calibri', 40)
65 display.fill(WHITE)
66
```

```

66
67 while True:
68     pygame.time.delay(SIMULACION_SPEED)
69     it = it + 1
70     if it <= 10000 and it >= 2:
71         states_temp = states.copy()
72         for x in range(nb_cols):
73             for y in range(nb_rows):
74                 state = states[x][y]
75                 if state == -1:
76                     pass
77                 if state >= 10:
78                     states_temp[x][y] = state + 1
79                 if state >= 20:
80                     if randrange(99) < PROBA_MUERTE:
81                         states_temp[x][y] = -1
82                     else:
83                         states_temp[x][y] = 1
84                 if state >= 10 and state <= 20:
85                     if randrange(99) < PROBA_INFECT:
86                         neighbour = get_vecinos(x, y)
87                         x2 = neighbour[0]
88                         y2 = neighbour[1]
89                         neigh_state = states[x2][y2]
90                         if neigh_state == 0:
91                             states_temp[x2][y2] = 10
92         states = states_temp.copy()
93         total_muerte = contar_muertes()
94
95     pygame.draw.rect(display, WHITE, (250, 30, 260, 50))
96     textsurface = font.render("Total muertes: "+ str(total_muerte), True, (255,160,122))
97     display.blit(textsurface, (250, 30))
98
99     for x in range(nb_cols):
100         for y in range(nb_rows):
101             if states[x][y] == 0:
102                 color = BLUE
103             if states[x][y] == 1:
104                 color = GREEN
105             if states[x][y] >= 10:
106                 color = (states[x][y] * 12, 50, 50)
107             if states[x][y] == -1:
108                 color = BLACK
109             pygame.draw.circle(display, color, (100 + x * 12 + 1, 100 + y * 12 + 1), 3)
110             pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 0))
111
112     for event in pygame.event.get():
113         if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE:
114             pygame.quit()
115         if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
116             #Reiniciamos valores
117             states = [[0] * nb_cols for i1 in range(nb_rows)]
118             states_temp = [[0] * nb_cols for i1 in range(nb_rows)]
119             states_temp = states.copy()
120             states[50][50] = 10
121             it = 0
122             total_muerte = 0
123             vacunar()
124
125     pygame.display.update()

```

En consecuencia, generar 5 simulaciones:

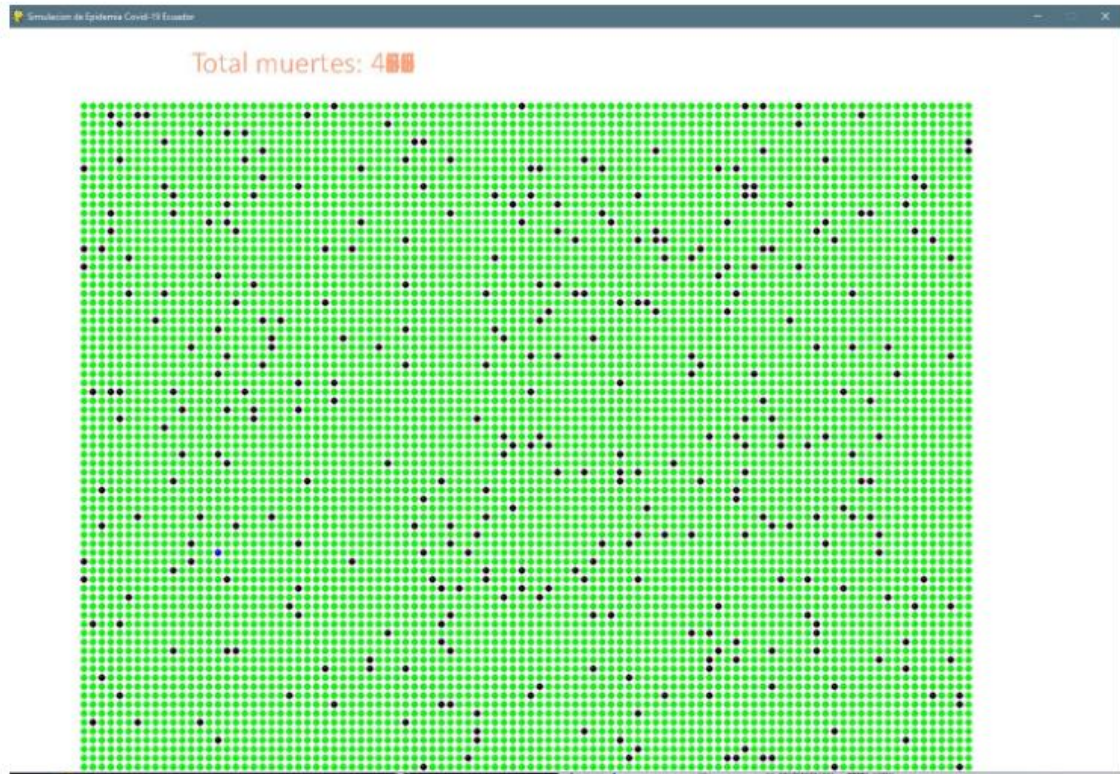
1. R0 obtenidos de la predicción del SIR (Trabajo anterior)
2. Predecir que va a ocurrir la próxima semana.
3. El valor 4, el cual representaría el peor de los casos.
4. El valor 1.4 en el mejor de los casos
5. R0 sin las medidas tomadas

Finalmente, agregar el número de días transcurridos, personas recuperadas y generar la curva SIR de las simulaciones.

Plantee y realice mejoras al modelo de simulación.

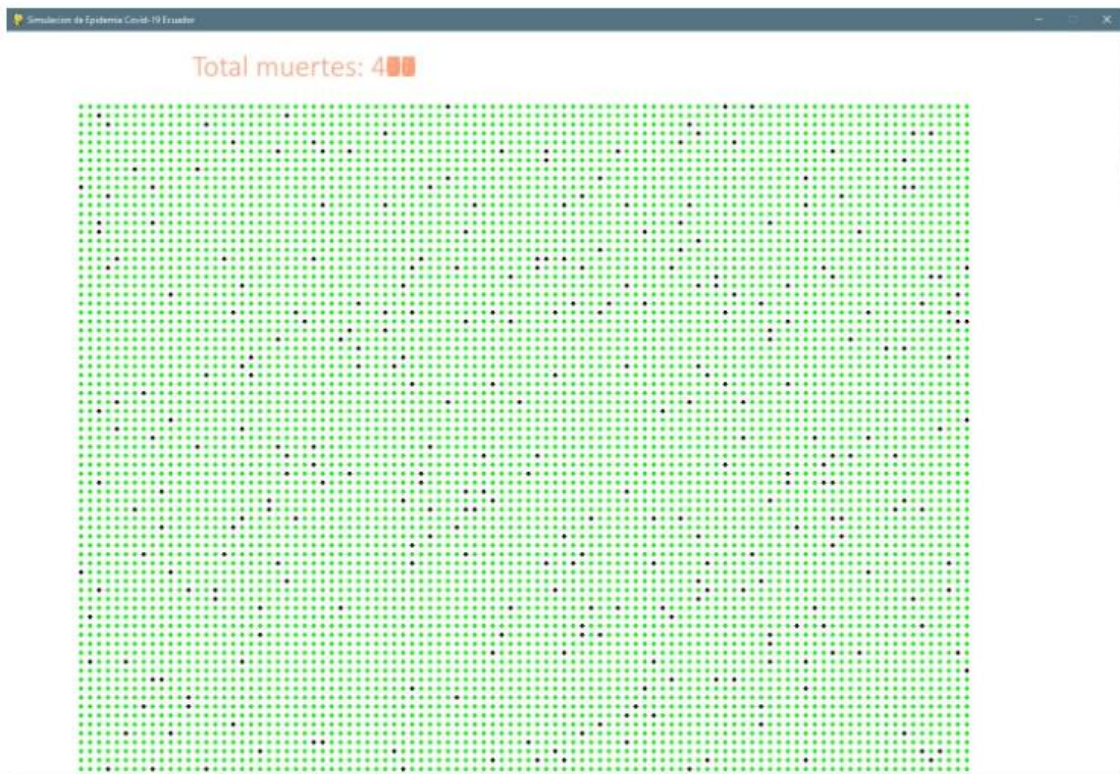
1 R0 obtenidos de la prediccion del SIR (Trabajo anterior)

91.52818043697134



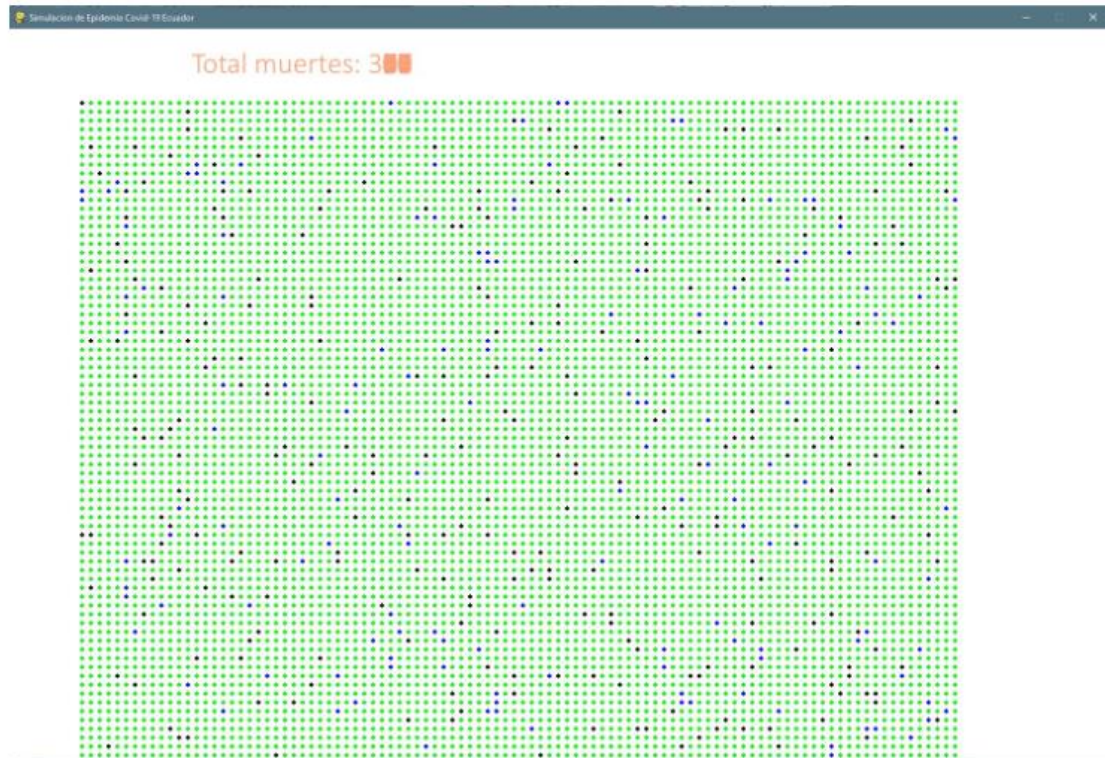
2 Predecir que va a ocurrir la proxima semana.

73.25426132532567



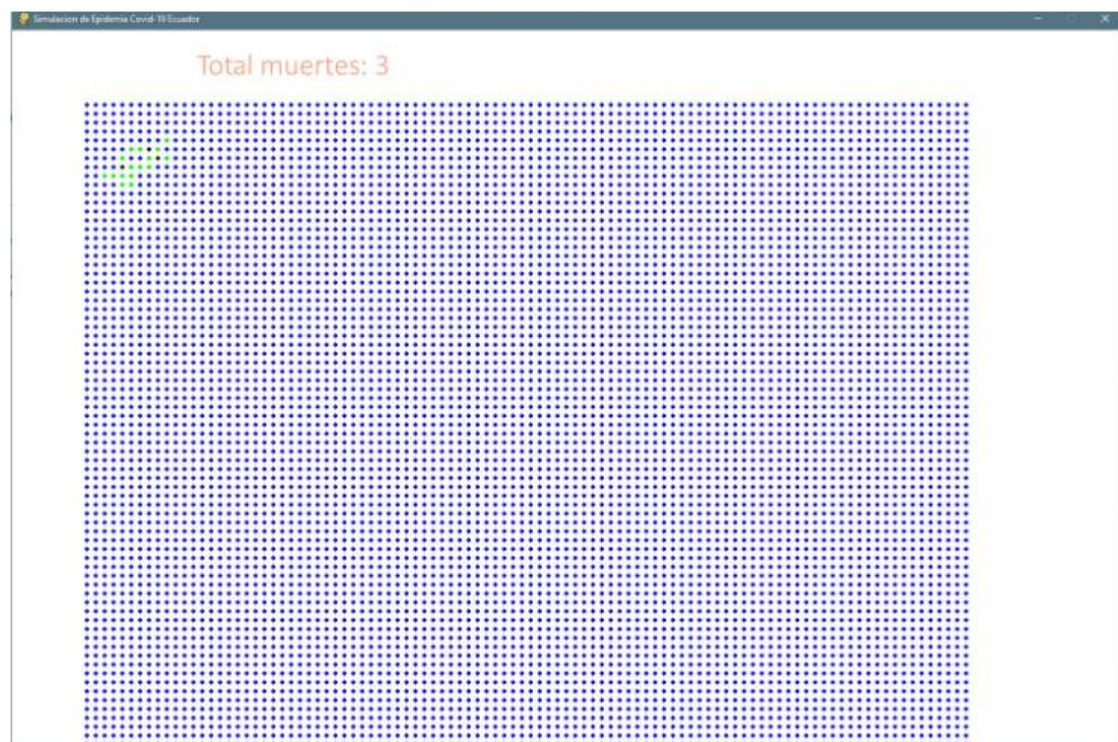
3 El valor 4, el cual representaría el peor de los casos.

```
PROBA_MUERTE = 4.2 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 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 = 25 # Tiempo de un dia en milisegundos
nb_rows = 100 #Numero de filas
nb_cols = 100 #Numero de columnas
```



4 El valor 1.4 en el mejor de los casos

```
#Parametros de inicio
PROBA_MUERTE = 4.2 # 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 = 25 # Tiempo de un dia en milisegundos
nb_rows = 100 #Numero de filas
nb_cols = 100 #Numero de columnas
```

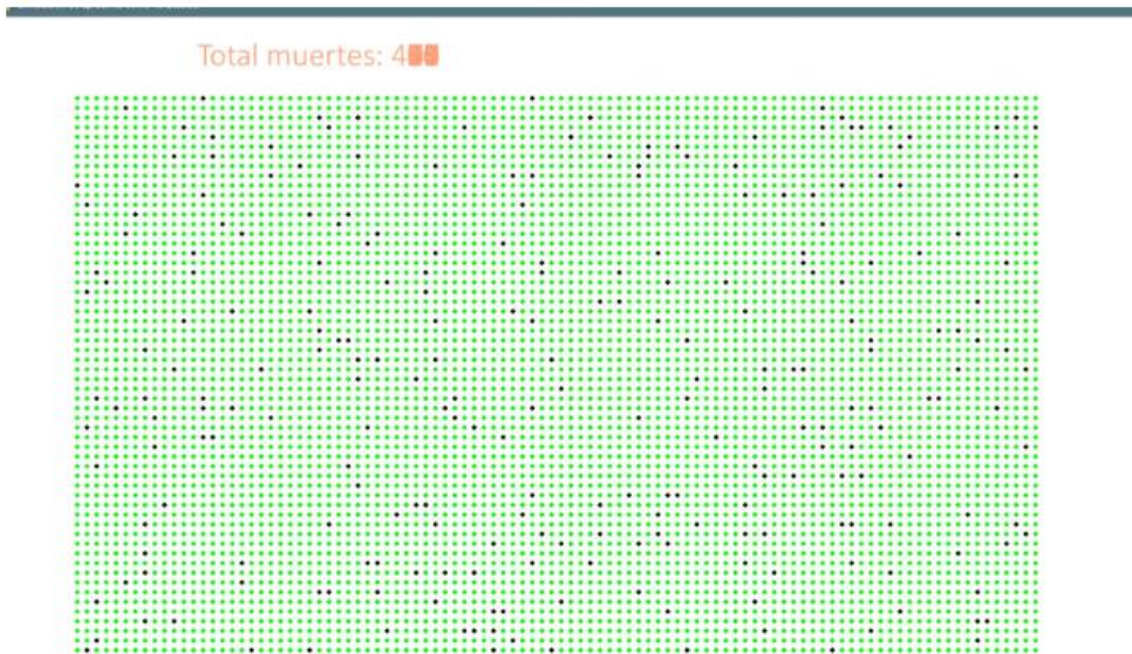


5 R0 sin las medidas tomada)

Cuarentena



sin Cuarentena



Analisis

con referente a los resultados el modelo para la simulación de contagiados nos puede dar una gran idea de cómo el virus se puede propagar en grandes cantidades de personas dando una media de muertes de un 450 por cada 10000 habitantes lo cual si es una cifra impresionante

Conclusiones

como parte del análisis de las muertes podemos decir que esta está en un nivel preocupante debido a que hay un crecimiento excesivo en la muerte y tener un promedio de 400 muertes es muy preocupante y si la cifra aumenta de contagiados hay muchas formas de que esto se des controle

Opinion

como opinión cabe acotar que estos son datos sintéticos un cuento los 5000 muertos que existen en el guayas ya que salieron fuera del contexto del coronavirus pero si por complicaciones respiratorias esto quiere decir que la curva de muertes es mucho mas grande lo que quiere decir que esta simulación podría ser peor lo que seguramente pasara en los proximos días en el país

Referencias

- <http://code.intef.es/simulamos-una-epidemia-virica/>
-