

Nombre: Jorge Arévalo

Docente: Ing. Diego Quisi

Materia: 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;
- 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). 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)

[.\(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 # Obtener un numero randomico
2  import pygame
3
4  #Parametros de inicio
5  PROBA_MUERTE = 8.4 # Probabilidad de que la gente muera COVID
6  CONTAGION_RATE = 4.0 # 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 = 50 # Tiempo de un dia en milisegundos (Cada 25 es un dia)
10 nb_rows = 50 #Numero de filas
11 nb_cols = 50 #Numero de columnas
12
13 global display, myfont, states, states_temp #Declaracion de variables globales
14
15 #Declaro colores en formato RGB
16 WHITE = (255, 255, 255)
17 BLUE = (0, 0, 255)
18 GREEN = (0, 247, 0)
19 BLACK = (0, 0, 0)
20
21 #Obtiene los vecinos dado un punto x,y
22 def get_vecinos(x, y):
23     incx = randrange(3)
24     incy = randrange(3)
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:
31         x2 = 0
32     if x2 >= nb_cols:
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
41 def vacunar():
42     for x in range(nb_cols):
43         for y in range(nb_rows):
44             if randrange(99) < PROBA_VACU:
45                 states[x][y] = 1
46
47 #Funcion que permite contar el numero de muertos de la matriz states == -1
48 def contar_muertes():
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()
59 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
62 vacunar() #Llamar a La funcion vacunar
63
64 pygame.init() #Incializo el motor de juegos pygame
65 pygame.font.init() #Inicializo el tipo de Letra
66 display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
67 pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
68 font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
69 display.fill(WHITE) # Color de fondo
70
71 while True:
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]
80                 if state == -1:
81                     pass
82                 if state >= 10: # Numero de dias de contagio
83                     states_temp[x][y] = state + 1
84                 if state >= 20:
85                     if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi
86                         states_temp[x][y] = -1 # Muere
87                     else:
88                         states_temp[x][y] = 1 # Cura o recupera
89                 if state >= 10 and state <= 20: # Rango de infectado
90                     if randrange(99) < PROBA_INFECT: # Infecto a Las personas cercanas
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]
95                         if neigh_state == 0: #Verifico que este sano
96                             states_temp[x2][y2] = 10 # Contagia
97             states = states_temp.copy()
98             total_muerte = contar_muertes() # contar el numero de muertos
99
100     pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
101     textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,12
102     display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
103     #Graficar el estado del paciente matriz
104     for x in range(nb_cols):
105         for y in range(nb_rows):
106             if states[x][y] == 0:
107                 color = BLUE # No infectado
108             if states[x][y] == 1:
109                 color = GREEN # Recupero
110             if states[x][y] >= 10:
111                 color = (states[x][y] * 12, 50, 50) # Inyectado - Rojo
112             if states[x][y] == -1:
113                 color = BLACK # Muerto
114             pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
115             pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1
116     #Escuchar 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()
124     states[5][5] = 10
125     it = 0
126     total_muerte = 0
127     vacunar() #Llamar a la funcion vacunar
128
129     pygame.display.update()# Mandar actualizar la ventana

```

Practica

En consecuencia, generar 5 simulaciones:

1. R0 investigar el valor de la varianza del RO 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 algún tipo de software que permita 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 simulacion.

Implementar

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

El “número reproductivo básico”, (R_0) refleja la media del número de casos secundarios que provocará un enfermo primario en la población que no tiene previa inmunidad.⁴ Para determinar el número reproductivo (R_0) se utiliza la fórmula siguiente en el que el producto ($? So$) es el número de nuevos casos que surgen (incidencia) y ($1/ ?$) que es el promedio de duración de la infección. Con el valor de R_0 se determina cuando un agente patógeno puede invadir o persistir en una población susceptible sin inmunidad. Los agentes patógenos pueden evolucionar aumentando R_0 .

Este valor es un indicador válido para medir la efectividad de las medidas de control que se requieren para eliminar las epidemias, cuando $R_0=1$ ó $R_0 < 1$ no hay epidemias.

$R_0 = ? So (1/ ?)$

El número de contagios va ir incrementando en provincias como Guayas, en un 10 a 20% y disminuyendo en Pichincha. El último reporte del índice RO en Pichincha es de 0.86, en Guayas estuvo en 1.3 a mediados de octubre, a finales de octubre estaba a 1.08 y en provincias como Bolivar estaba en 1.29.

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

In []:

```

1  from random import randrange # Obtener un numero randomico
2  import pygame
3
4  #Parametros de inicio
5  PROBA_MUERTE = 3.9 # Probabilidad de que la gente muera COVID
6  CONTAGION_RATE = 4.0 # 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 = 50 # Tiempo de un dia en milisegundos (Cada 25 es un dia)
10 nb_rows = 50 #Numero de filas
11 nb_cols = 50 #Numero de columnas
12
13 global display, myfont, states, states_temp #Declaracion de variables globales
14
15 #Declaro colores en formato RGB
16 WHITE = (255, 255, 255)
17 BLUE = (0, 0, 255)
18 GREEN = (0, 247, 0)
19 BLACK = (0, 0, 0)
20
21 #Obtiene los vecinos dado un punto x,y
22 def get_vecinos(x, y):
23     incx = randrange(3)
24     incy = randrange(3)
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:
31         x2 = 0
32     if x2 >= nb_cols:
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
41 def vacunar():
42     for x in range(nb_cols):
43         for y in range(nb_rows):
44             if randrange(99) < PROBA_VACU:
45                 states[x][y] = 1
46
47 #Funcion que permite contar el numero de muertos de la matriz states == -1
48 def contar_muertes():
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()
59 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
62 vacunar() #Llamar a La funcion vacunar
63
64 pygame.init() #Incializo el motor de juegos pygame
65 pygame.font.init() #Inicializo el tipo de letra
66 display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
67 pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
68 font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
69 display.fill(WHITE) # Color de fondo
70
71 while True:
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]
80                 if state == -1:
81                     pass
82                 if state >= 10: # Numero de dias de contagio
83                     states_temp[x][y] = state + 1
84                 if state >= 20:
85                     if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi
86                         states_temp[x][y] = -1 # Muere
87                     else:
88                         states_temp[x][y] = 1 # Cura o recupera
89                 if state >= 10 and state <= 20: # Rango de infectado
90                     if randrange(99) < PROBA_INFECT: # Infecto a Las personas cercanas
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]
95                         if neigh_state == 0: #Verifico que este sano
96                             states_temp[x2][y2] = 10 # Contagia
97             states = states_temp.copy()
98             total_muerte = contar_muertes() # contar el numero de muertos
99
100     pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
101     textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,12
102     display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
103     #Graficar el estado del paciente matriz
104     for x in range(nb_cols):
105         for y in range(nb_rows):
106             if states[x][y] == 0:
107                 color = BLUE # No infectado
108             if states[x][y] == 1:
109                 color = GREEN # Recupero
110             if states[x][y] >= 10:
111                 color = (states[x][y] * 12, 50, 50) # Inyectado - Rojo
112             if states[x][y] == -1:
113                 color = BLACK # Muerto
114             pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
115             pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1
116     #Escuchar 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()
124     states[5][5] = 10
125     it = 0
126     total_muerte = 0
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 3.7.6)

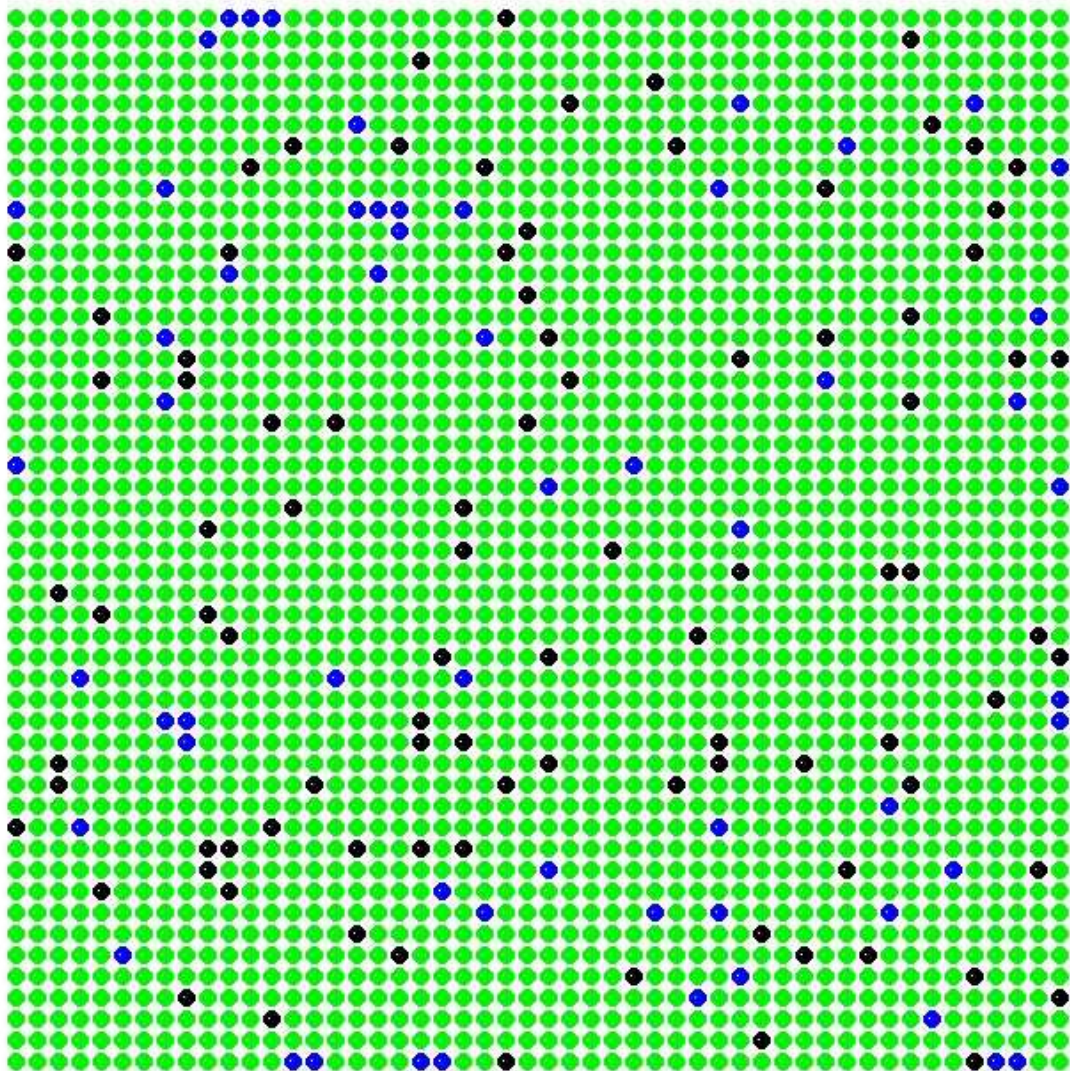
Hello from the pygame community. <https://www.pygame.org/contribute.html> (<https://www.pygame.org/contribute.html>)

Resultado

 Simulacion de Epidemia Covid-19 Ecuador

— □ ×

Total muertes: 9 



3. El valor 1.4 en el mejor de los casos

In [*]:

```

1  from random import randrange # Obtener un numero randomico
2  import pygame
3
4  #Parametros de inicio
5  PROBA_MUERTE = 1.4 # Probabilidad de que la gente muera COVID
6  CONTAGION_RATE = 4.0 # 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 = 50 # Tiempo de un dia en milisegundos (Cada 25 es un dia)
10 nb_rows = 50 #Numero de filas
11 nb_cols = 50 #Numero de columnas
12
13 global display, myfont, states, states_temp #Declaracion de variables globales
14
15 #Declaro colores en formato RGB
16 WHITE = (255, 255, 255)
17 BLUE = (0, 0, 255)
18 GREEN = (0, 247, 0)
19 BLACK = (0, 0, 0)
20
21 #Obtiene los vecinos dado un punto x,y
22 def get_vecinos(x, y):
23     incx = randrange(3)
24     incy = randrange(3)
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:
31         x2 = 0
32     if x2 >= nb_cols:
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
41 def vacunar():
42     for x in range(nb_cols):
43         for y in range(nb_rows):
44             if randrange(99) < PROBA_VACU:
45                 states[x][y] = 1
46
47 #Funcion que permite contar el numero de muertos de la matriz states == -1
48 def contar_muertes():
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()
59 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
62 vacunar() #Llamar a La funcion vacunar
63
64 pygame.init() #Incializo el motor de juegos pygame
65 pygame.font.init() #Inicializo el tipo de letra
66 display=pygame.display.set_mode((800,750),0,32) #Tamanio de La ventana
67 pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
68 font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
69 display.fill(WHITE) # Color de fondo
70
71 while True:
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]
80                 if state == -1:
81                     pass
82                 if state >= 10: # Numero de dias de contagio
83                     states_temp[x][y] = state + 1
84                 if state >= 20:
85                     if randrange(99) < PROBA_MUERTE: # Genero un randomico para verifi
86                         states_temp[x][y] = -1 # Muere
87                     else:
88                         states_temp[x][y] = 1 # Cura o recupera
89                 if state >= 10 and state <= 20: # Rango de infectado
90                     if randrange(99) < PROBA_INFECT: # Infecto a Las personas cercanas
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]
95                         if neigh_state == 0: #Verifico que este sano
96                             states_temp[x2][y2] = 10 # Contagia
97             states = states_temp.copy()
98             total_muerte = contar_muertes() # contar el numero de muertos
99
100     pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico el fondo
101     textsurface = font.render("Total muertes: "+ str(total_muerte), False, (255,160,12
102     display.blit(textsurface, (250, 30)) # Graficar el texto de muertes
103     #Graficar el estado del paciente matriz
104     for x in range(nb_cols):
105         for y in range(nb_rows):
106             if states[x][y] == 0:
107                 color = BLUE # No infectado
108             if states[x][y] == 1:
109                 color = GREEN # Recupero
110             if states[x][y] >= 10:
111                 color = (states[x][y] * 12, 50, 50) # Inyectado - Rojo
112             if states[x][y] == -1:
113                 color = BLACK # Muerto
114             pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5
115             pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1
116     #Escuchar 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

```



```

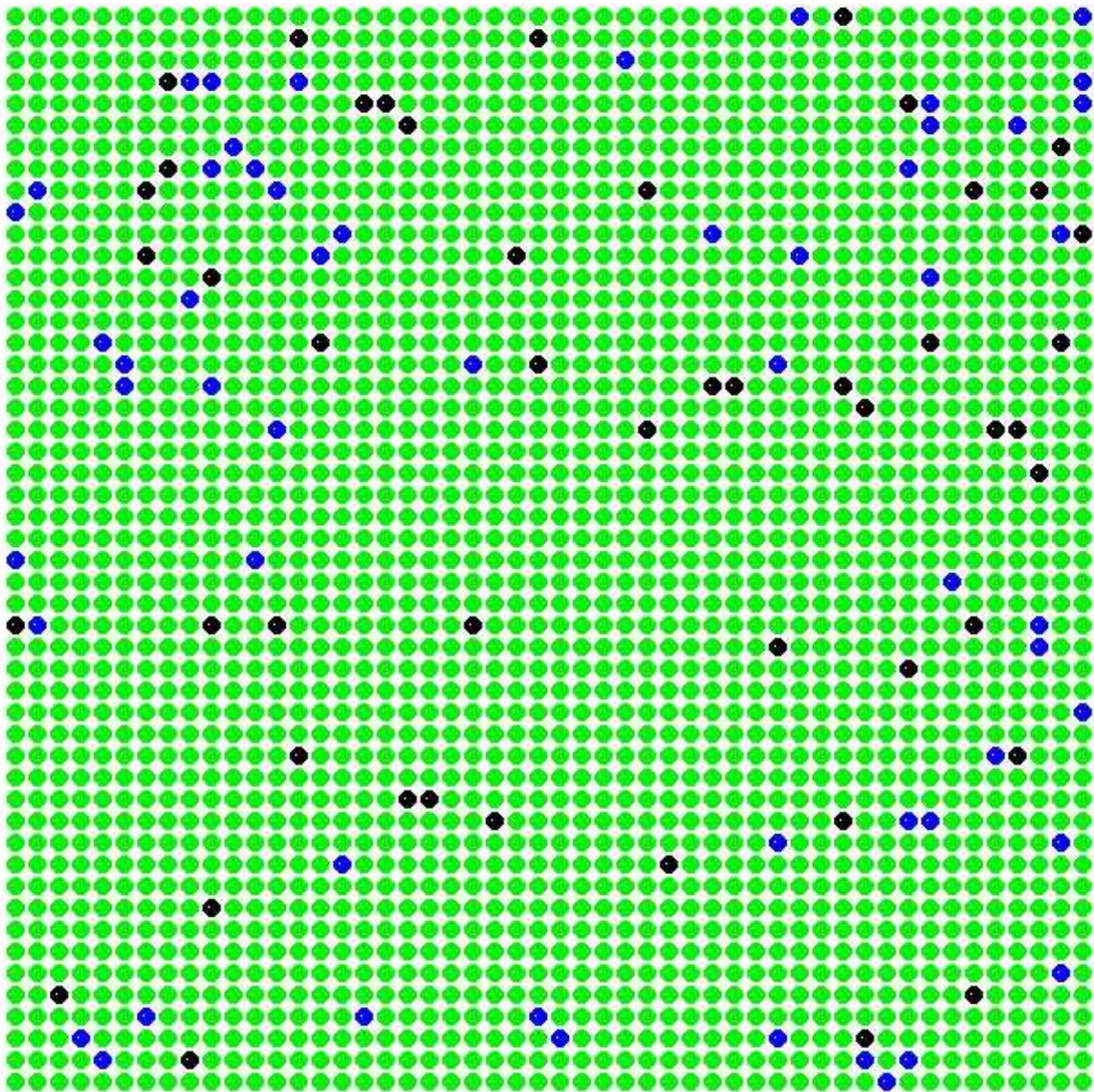
121     #Reiniciamos valores
122     states = [[0] * nb_cols for i1 in range(nb_rows)]
123     states_temp = states.copy()
124     states[5][5] = 10
125     it = 0
126     total_muerte = 0
127     vacunar() #Llamar a la funcion vacunar
128
129     pygame.display.update()# Mandar actualizar la ventana

```

pygame 2.0.0 (SDL 2.0.12, python 3.7.6)

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

Total muertes: 4 🏴



4. Revisar e investigar algún tipo de software que permita simular la tasa de contagio en una epidemia,

aplicar a los datos del Ecuador y obtener un R0 con los datos del país.

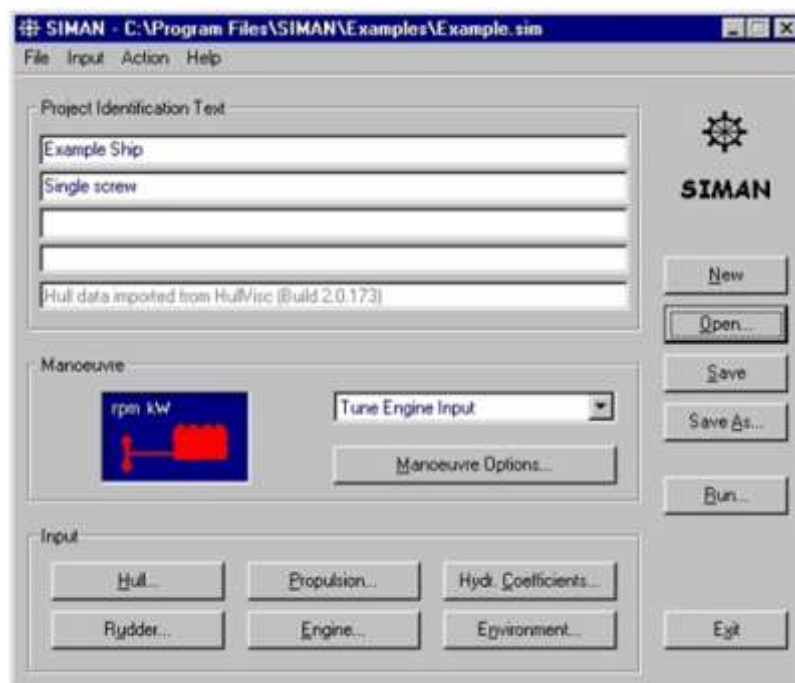
1. MATLAB
2. FlexSim
3. Simio
4. Anylogic
5. Sim3D

Su propósito es predecir el resultado de una situación que representa un riesgo o un peligro, lo cual puede tener distintos propósitos. El principal es tener la información necesaria para poder desarrollar planes de contingencia adecuados además de comprobar su eficacia. Otra utilidad es simular el riesgo para medir los posibles costes que causaría su materialización y de esta forma decidir si es rentable aplicar medidas preventivas para evitar que suceda.

Software SIMAN

DESCRIPCIÓN

Modela un sistema discreto usando la orientación al proceso, se estudian las entidades que se mueven a través del sistema. Una entidad para SIMAN es un cliente, un objeto que se mueve en la simulación y que posee características únicas conocidas como atributos. Los procesos denotan la secuencia de operaciones o actividades a través del que se mueven las entidades, siendo modeladas por el diagrama de bloques.



Analisis

Los modelos de simulación para el análisis de sistemas complejos, lo que ha generado la proliferación tanto como de lenguajes de Simulación como de libros de texto que faciliten el aprendizaje de los mismos.

Conclusiones

Los software de simulación han sido aplicados para modelar Sistemas dinámicos Complejos que no son posibles mediante otros métodos estocásticos, por su versatilidad y la posibilidad de experimentar con diferentes opciones sin que esto represente un alto costo o una violación de la realidad.

Opinion

facilidad de interpretación de resultados sin conocimientos previos, lo cual representa una herramienta poderosa en la toma de decisiones en las empresas de todo tipo y en donde sus líderes pueden manejar.

Referencias

- <http://code.intef.es/simulamos-una-epidemia-virica/> (<http://code.intef.es/simulamos-una-epidemia-virica/>)
- <https://softwareparatodo.com/software-de-simulacion/#MATLAB> (<https://softwareparatodo.com/software-de-simulacion/#MATLAB>)
- <https://astridmll.wordpress.com/2016/09/07/tipos-de-software-de-simulacion/> (<https://astridmll.wordpress.com/2016/09/07/tipos-de-software-de-simulacion/>)
- <https://www.redalyc.org/pdf/4962/496250974007.pdf> (<https://www.redalyc.org/pdf/4962/496250974007.pdf>)