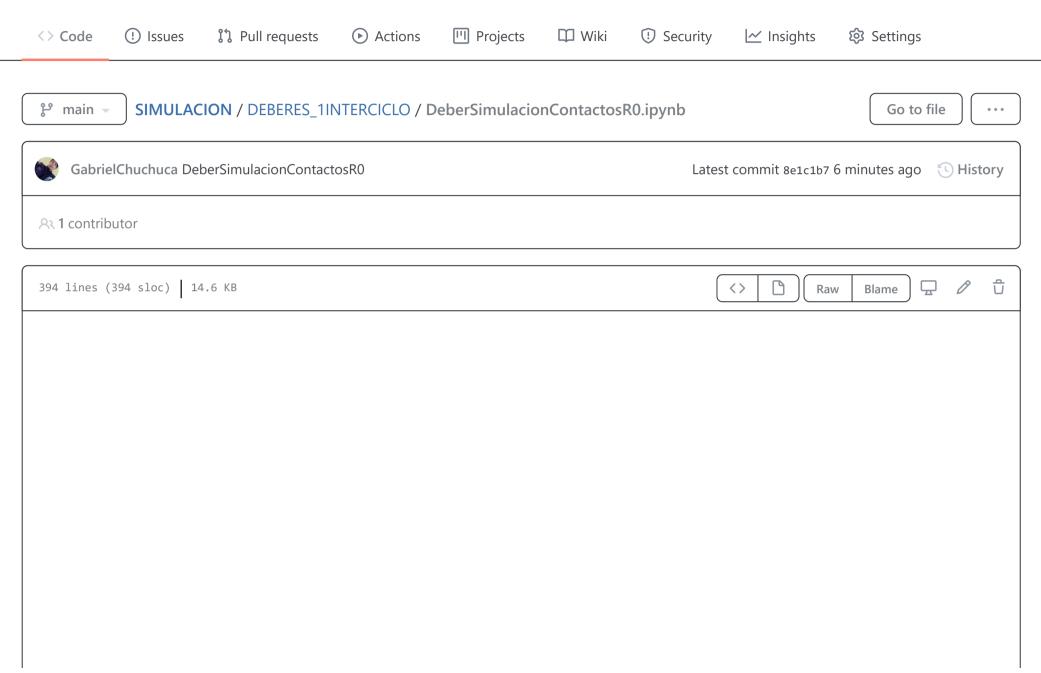
☐ GabrielChuchuca / SIMULACION



- Investigar el valor de varianza de 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%
- Valor 4, para el peor de los casos.
- Valor 1.4, en el mejor de los casos.

Valor R0 -> 4

```
In [1]: from random import randrange # Obtener un numero randomico
         import pygame
         pygame 2.0.1 (SDL 2.0.14, Python 3.8.3)
        Hello from the pygame community. https://www.pygame.org/contribute.html
In [2]: #Parametros de inicio
        PROBA MUERTE = 7.1 # Probabilidad de que la gente muera COVID
        CONTAGION RATE = 4 # 0.59 - 1.10 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)
In [3]: #Obtiene los vecinos dado un punto x,y
        def get_vecinos(x, y):
            incx = randrange(3)
            incy = randrange(3)
            incx = (incx * 1) - 1
            incv = (incv * 1) - 1
```

```
x2 = x + incx
            y2 = y + incy
             #Validar limites
             if x2 < 0:
                x2 = 0
            if x2 >= nb cols:
                x2 = nb cols - 1
             if v2 < 0:
                v2 = 0
             if y2 >= nb rows:
                 v2 = 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:</pre>
                         states[x][v] = 1
In [5]: #Funcion que permite contar el numero de muertosde la matriz states == -1
        def contar muertes():
             contador = 0
            for x in range(nb cols):
                 for y in range(nb rows):
                     if states[x][v] == -1:
                         contador += 1
             return contador
In [ ]: #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 del Infectado
         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) #Tamanio de La ventana

nuceum display set contien/"Simulacien de Enidemia Covid 10 Ecuador"\# Titula

```
phäquie.ntphtah.per_cabriou( primatacion ne chinemita conta-ia ccnanoi. )# Illinio
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 verificar si fallece o se rec
upera
                        states temp[x][y] = -1 \# Muere
                    else:
                        states_temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infectado</pre>
                    if randrange(99) < PROBA INFECT: # Infecto a las personas cercanas entre 10 y 20
                        neighbour = get vecinos(x, y) #Obtenemos los vecinos a contagiar
                        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, (255,160,122)) #EL numero de muert
os
   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 = RILIF # No infectado
```

```
if states[x][y] == 1:
            color = GREEN # Recupero
        if states[x][y] \Rightarrow= 10:
            color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
        if states[x][v] == -1:
            color = BLACK # Muerto
        pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5)
        pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1))
#Escuachar los eventos del teclado
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y Escape
        pygame.quit() #Termino simulacion
    if event.type == pygame.KEYDOWN and event.key == pygame.K SPACE: #Presiona y espacio
        #Reiniciamos valores
        states = [[0] * nb cols for i1 in range(nb rows)]
        states temp = states.copy()
        states[5][5] = 10
        it = 0
        total muerte = 0
        vacunar() #Llamar a la funcion vacunar
pygame.display.update()# Mandar actualizar La ventana
```

Valor R0 -> 1.4

```
In []: #Parametros de inicio

PROBA_MUERTE = 7.1  # Probabilidad de que la gente muera COVID

CONTAGION_RATE = 1.4  # 0.59 - 1.10 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)
In [1]: #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:
                 v2 = 0
             if y2 >= nb_rows:
                y2 = nb rows - 1
             return [x2, y2] # Nuevos contagiados
In [2]: #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:</pre>
                         states[x][y] = 1
In [3]: | #Funcion que permite contar el numero de muertosde 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
In [ ]: #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 del Infectado
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) #Tamanio de La ventana
pygame.display.set caption("Simulacion de Epidemia Covid-19 Ecuador")# Titulo
font=pygame.font.SysFont('Calibri', 40) # Tipo de Letra
display.fill(WHITE) # Color de fondo
while 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][v]
                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 verificar si fallece o se rec
upera
                        states temp[x][y] = -1 # Muere
                    else:
                        states temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infectado</pre>
                    if randrange(99) < PROBA INFECT: # Infecto a Las personas cercanas entre 10 y 20
                        neighbour = get vecinos(x, y) #Obtenemos los vecinos a contagiar
                        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, (255,160,122)) #EL numero de muert
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][v] == 0:
            color = BLUE # No infectado
        if states[x][y] == 1:
            color = GREEN # Recupero
       if states[x][v] >= 10:
            color = (states[x][y] * 12, 50, 50) # Injectado - Rojo
       if states[x][y] == -1:
            color = BLACK # Muerto
        pygame.draw.circle(display, color, (100 + x * 12 + 5, 100 + y * 12 + 5), 5)
        pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y * 12 + 4, 1, 1))
#Escuachar los eventos del teclado
for event in pygame.event.get():
    if event.type == pygame.KEYDOWN and event.key == pygame.K ESCAPE: #Presiona y Escape
        pygame.quit() #Termino simulacion
    if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE: #Presiona y espacio
        #Reiniciamos valores
        states - [[0] * nh cols for il in nango(nh nows)]
```