



## **Simulacion Contactos R0**

**Materia:**

**Simulación**

**Docente:**

**Ing. Diego Quisi**

**Estudiante:**

**Ricardo Vinicio Jara Jara**

# 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/>

(<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)

([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 [2]: from random import randrange # Obtener un numero randomico
import pygame

PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
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

def _run_prediction(valor_ro=4.0):
    #Parametros de inicio
    PROBA_MUERTE = 8.4 # Probabilidad de que la gente muera COVID

```

```

CONTAGION_RATE = valor_ro # Factor R0 para la simulacion COVID probabilida
d
PROBA_INFECT = CONTAGION_RATE * 10
SIMULACION_SPEED = 50 # Tiempo de un dia en milisegundos (Cada 25 es un di
a)

global display, myfont, states, states_temp #Declaracion de variables glob
ales
#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 simulacio
n 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")# Tit
ulo
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 recupera
                        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 perso
nas cercanas entre 10 y 20
                        neighbour = get_vecinos(x, y) #Obtenemos Los vecin
os 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 fond

```

```

0
    textsurface = font.render("Total muertes: "+ str(total_muerte), False,
(255,160,122)) #El numero de muertos
    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), 5)
            pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 100 + y *
12 + 4, 1, 1))
        #Escuchar 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
#_run_prediction()

```

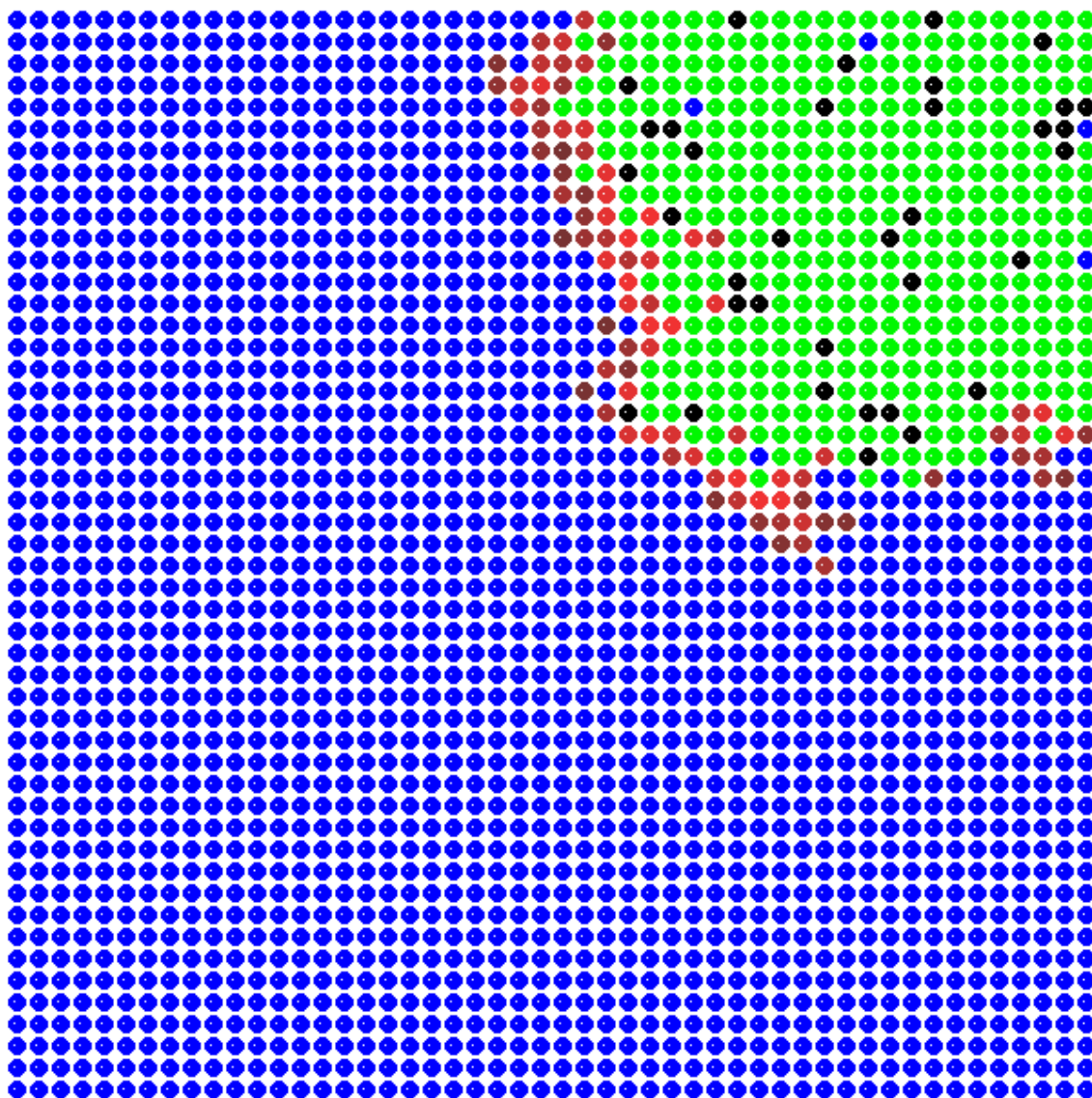
pygame 2.0.0 (SDL 2.0.12, python 3.8.3)

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

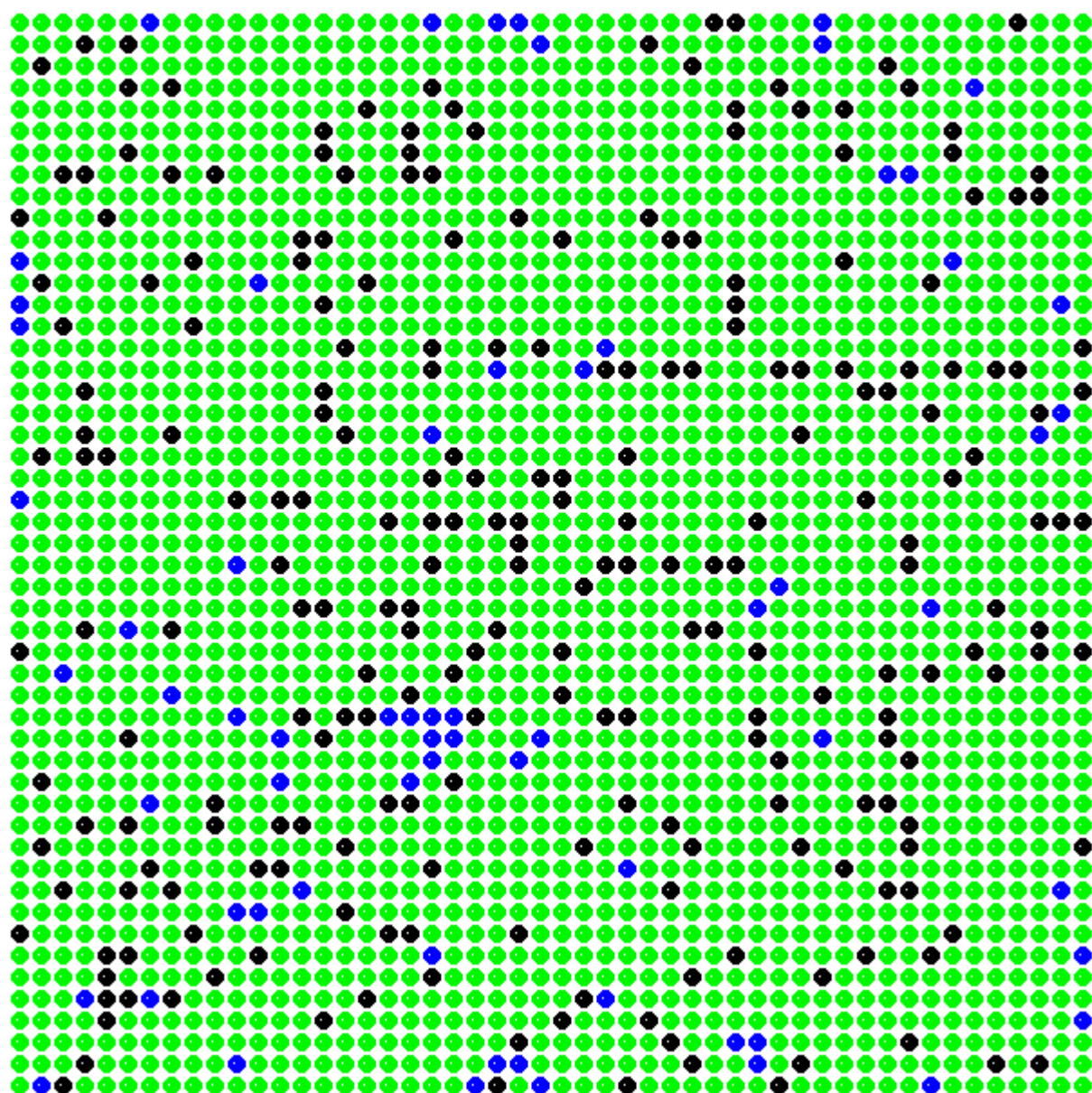
## COLORES

- Azul No infectado
- Verde Recuperado
- Rojo Infectado
- Negro Muerto

Total muertes: 34



Total muertes: 244



# Practica

**1. R0 investigar el valor de varianza del R0 dentro de Ecuador** Los científicos usan el R0 -el número de reproducción- para describir la intensidad de una enfermedad infecciosa.

- Quiénes son los superpropagadores y por qué tienen tanta importancia en un brote como el del coronavirus
- ¿Cuán preocupados debemos estar por el coronavirus?

Las estimaciones de R0 han sido una parte importante de la descripción de las pandemias o de los brotes muy conocidos, incluyendo la pandemia del Síndrome respiratorio agudo grave (SARS, por sus siglas en inglés) de 2003, la pandemia de influenza H1N1 de 2009 y la epidemia de ébola en África occidental de 2014.

El número básico de reproducción representa el máximo potencial epidémico de un patógeno. Describe lo que ocurriría si una persona infectada entra en contacto con una comunidad totalmente susceptible y, por tanto, es un estimado que se base en un escenario idealizado.

El número efectivo de reproducción depende de la verdadera susceptibilidad de la población. Esta medida del potencial de transmisión es probablemente menor que el número de reproducción básica, sobre la base de factores como si algunas personas están vacunadas en contra de la enfermedad o si algunos tienen inmunidad debido a que previamente estuvieron expuestos al patógeno.

**Por tanto, el R0 efectivo cambia con el tiempo y es un estimado que se basa en las condiciones de la población.**

## Ejemplificacion

Si R0 es menor que 1, **cada infección existente causaría menos de una infección nueva:**

- por ejemplo: 100 personas enfermas infectarán a menos de otras 100.

Cada generación sucesiva de infecciones será más pequeña que la anterior. Es decir la enfermedad irá disminuyendo y finalmente desaparecerá.

Si el valor de R0 es igual a 1, **cada infección existente causa una nueva infección.**

- La infección causa más de una infección nueva. La enfermedad se transmitirá entre las personas y podría haber un brote o epidemia.

El término mal llamado **Tasa Básica reproductiva** viene de la demografía donde se utiliza para describir las tasas de natalidad. De ahí que la R se refiere a la reproducción y el 0 a la generación cero, como en el paciente "cero". Por eso se le conoce de manera conjunta como Número básico reproductivo o R0

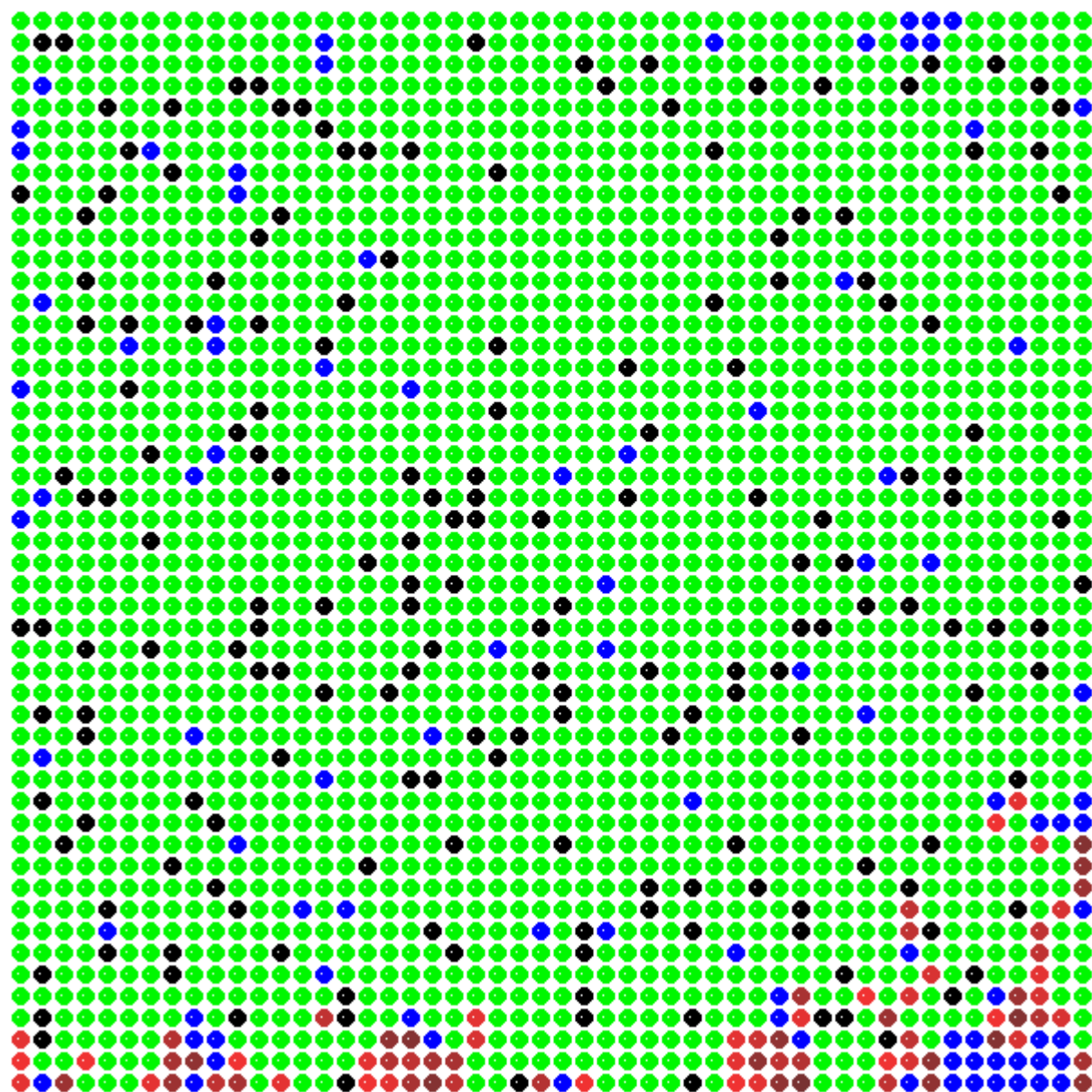
**Un equipo del Imperial College calcula que el R0 está entre 1,5 y 3,5. Científicos de la Academia China del Instituto de Ciencias de la Automatización y la Universidad de la Academia de Ciencias de China estiman una cifra mucho más alta, de 4,08.**



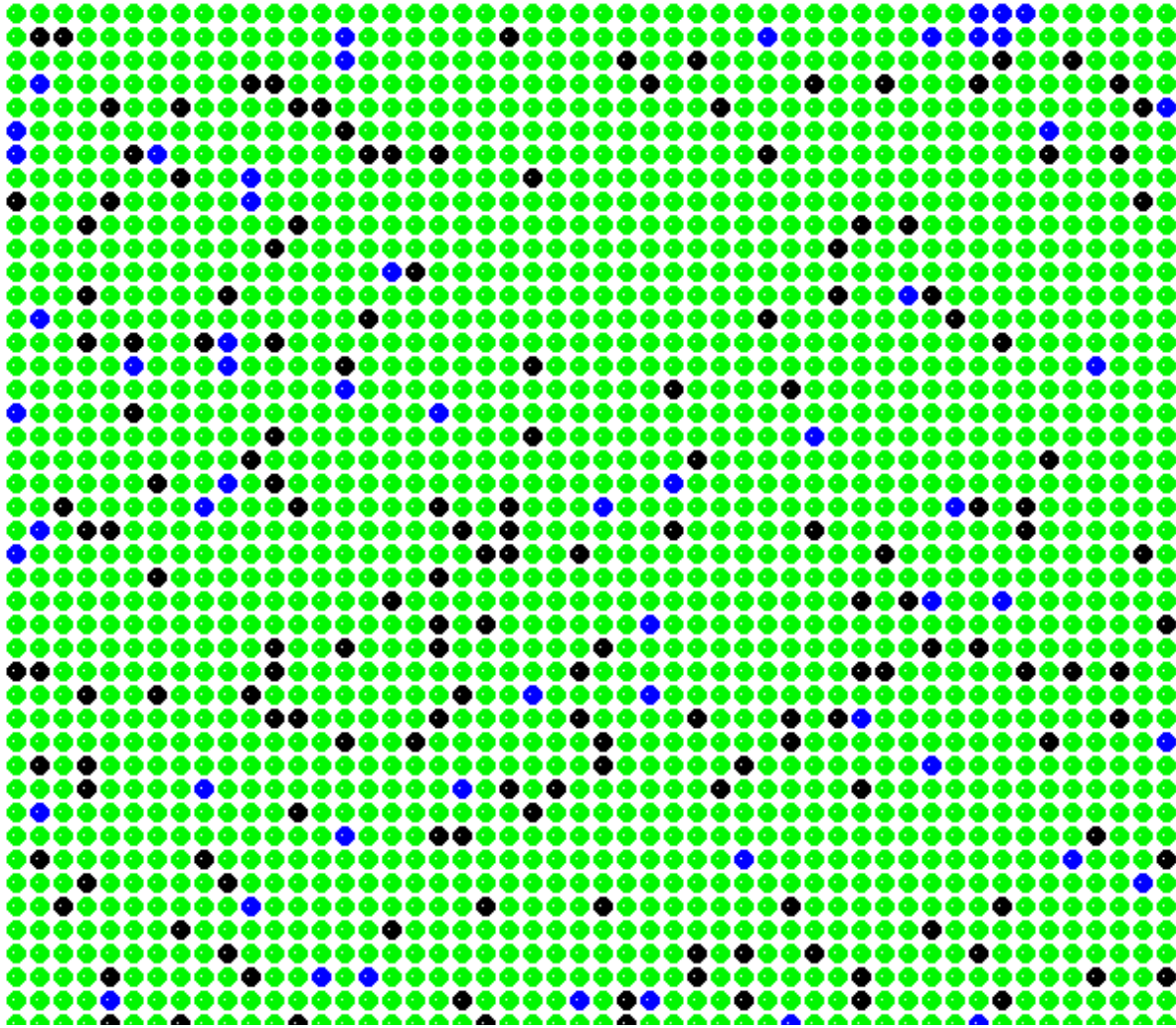
### 3. El valor 4, el cual representaría el peor de los casos.

```
In [ ]: CONTAGION_RATE = 4.0  
        _run_prediction(CONTAGION_RATE)
```

Total muertes: 144

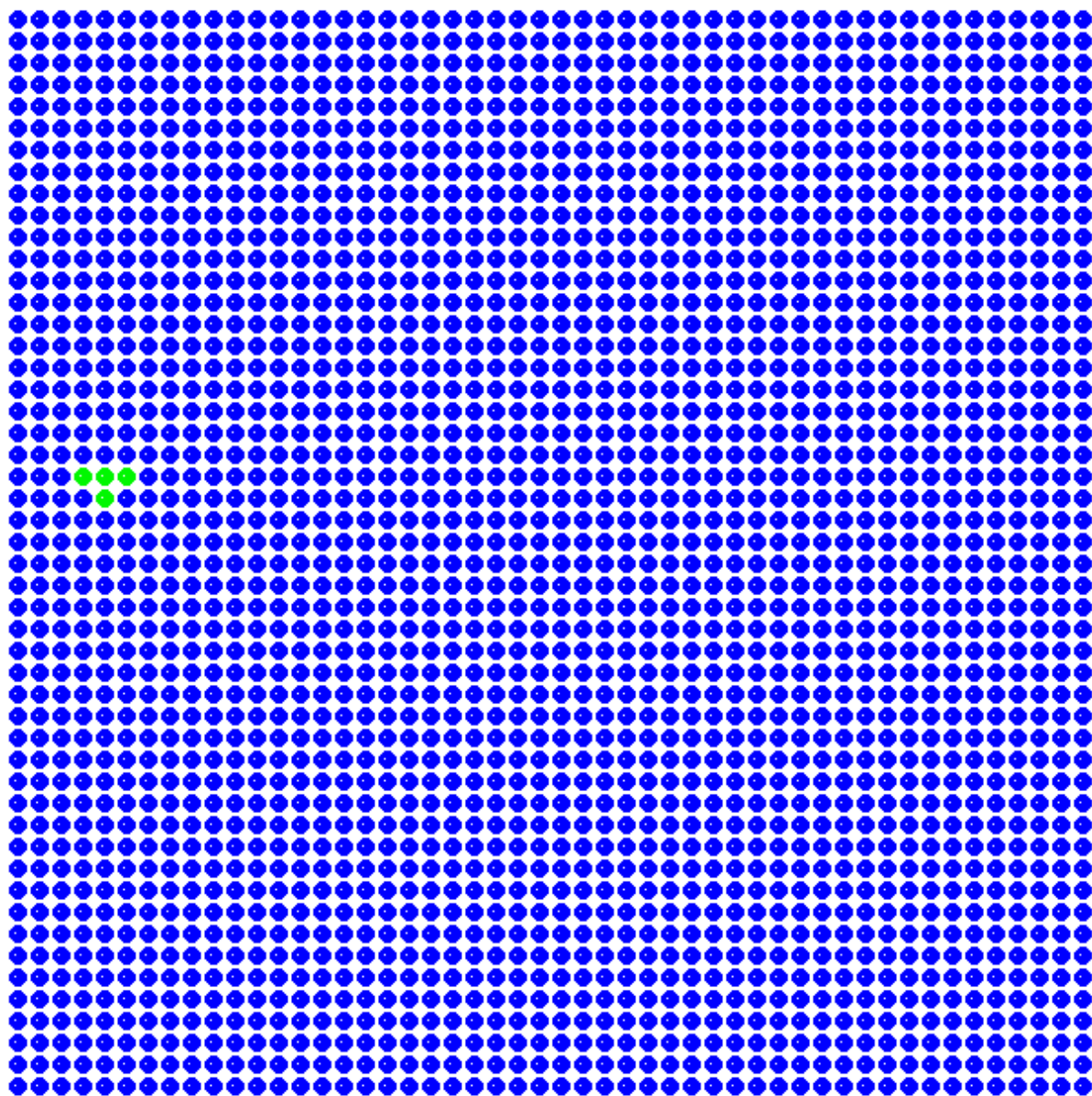


Total muertes: 144



```
In [ ]: CONTAGION_RATE = 1.4  
_run_prediction(CONTAGION_RATE)
```

Total muertes: 0



5. 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 RO con los datos del pais.

Parameter	Symbol	Typical
Reproduction number	$R_0$	2.4
Incubation period (days)	$\tau_{incubation}$	5.1
Infectious period (days)	$\tau_{infectious}$	3.3
Campus population	$N$	20,000
Initial number exposed	$n$	10
Mitigation by social distancing	$u$	0.2
Start of social distancing following exposure (weeks)	$t_{sd}$	2

```

In [10]: R0 = 4.0
t_incubation = 5.1
t_infective = 3.3
N = 16000000
n = 10
t_social_distancing = 2
u_social_distancing = 40

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
import seaborn as sns

e_initial = n/N
i_initial = 0.00
r_initial = 0.00
s_initial = 1 - e_initial - i_initial - r_initial

alpha = 1/t_incubation
gamma = 1/t_infective
beta = R0*gamma

def step(t):
    return 1 if t >= 7*t_social_distancing else 0

# SEIR model differential equations.
def deriv(x, t, u, alpha, beta, gamma):
    s, e, i, r = x
    dsdt = -(1-u*step(t)/100)*beta * s * i
    dedt = (1-u*step(t)/100)*beta * s * i - alpha * e
    didt = alpha * e - gamma * i
    drdt = gamma * i
    return [dsdt, dedt, didt, drdt]

t = np.linspace(0, 210, 210)
x_initial = s_initial, e_initial, i_initial, r_initial
s, e, i, r = odeint(deriv, x_initial, t, args=(u_social_distancing, alpha, beta, gamma)).T
s0, e0, i0, r0 = odeint(deriv, x_initial, t, args=(0, alpha, beta, gamma)).T

# plot the data
fig = plt.figure(figsize=(12, 10))
ax = [fig.add_subplot(311, axisbelow=True),
      fig.add_subplot(312)]

pal = sns.color_palette()

ax[0].stackplot(t/7, N*s, N*e, N*i, N*r, colors=pal, alpha=0.6)
ax[0].set_title('Susceptible and Recovered Populations with {0:3.0f}% Effective Social Distancing'.format(u_social_distancing))
ax[0].set_xlabel('Weeks following Initial Campus Exposure')
ax[0].set_xlim(0, t[-1]/7)
ax[0].set_ylim(0, N)
ax[0].legend(['Susceptible',

```

```

    'Exposed/no symptoms',
    'Infectious/symptomatic',
    'Recovered'],
    loc='best')
ax[0].plot(np.array([t_social_distancing, t_social_distancing]), ax[0].get_yli
m(), 'r', lw=3)
ax[0].plot(np.array([0, t[-1]])/7, [N/R0, N/R0], lw=3, label='herd immunity')
ax[0].annotate("Start of social distancing",
    (t_social_distancing, 0), (t_social_distancing + 1.5, N/10),
    arrowprops=dict(arrowstyle='->'))
ax[0].annotate("Herd Immunity without social distancing",
    (t[-1]/7, N/R0), (t[-1]/7 - 8, N/R0 - N/5),
    arrowprops=dict(arrowstyle='->'))

ax[1].stackplot(t/7, N*i0, N*e0, colors=pal[2:0:-1], alpha=0.5)
ax[1].stackplot(t/7, N*i, N*e, colors=pal[2:0:-1], alpha=0.5)
ax[1].set_title('Infected Population with no Social Distancing and with {0:3.0
f}% Effective Social Distancing'.format(u_social_distancing))
ax[1].set_xlim(0, t[-1]/7)
ax[1].set_ylim(0, max(0.3*N, 1.05*max(N*(e + i))))
ax[1].set_xlabel('Weeks following Initial Campus Exposure')
ax[1].legend([
    'Infective/Symptomatic',
    'Exposed/Not Symptomatic'],
    loc='upper right')
ax[1].plot(np.array([t_social_distancing, t_social_distancing]), ax[0].get_yli
m(), 'r', lw=3)

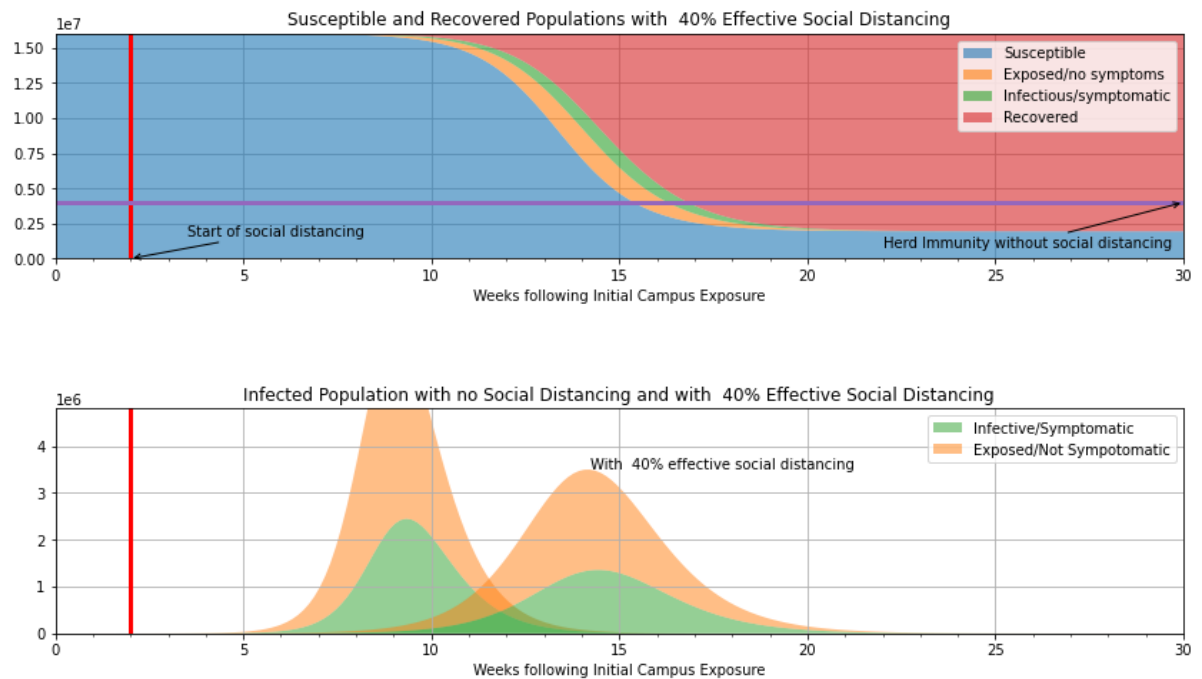
y0 = N*(e0 + i0)
k0 = np.argmax(y0)
ax[1].annotate("No social distancing", (t[k0]/7, y0[k0] + 100))

y = N*(e + i)
k = np.argmax(y)
ax[1].annotate("With {0:3.0f}% effective social distancing ".format(u_social_d
istancing), (t[k]/7, y[k] + 100))

for a in ax:
    a.xaxis.set_major_locator(plt.MultipleLocator(5))
    a.xaxis.set_minor_locator(plt.MultipleLocator(1))
    a.xaxis.set_major_formatter(plt.FormatStrFormatter('%d'))
    a.grid(True)

plt.tight_layout()

```



## Analisis

Realizar las distintas simulaciones sobre la epidemia en Ecuador y sobretodo aplicando el valor  $R_0$  se podría decir que este valor es muy importante para ver el crecimiento y cantidad de personas que van a ser afectadas ya que si es que existe un valor alto la cantidad de infectados sería mayor y si existe al contrario bajo la cantidad de infectados e incluso irá desapareciendo en el transcurso del tiempo sin embargo como vemos que el  $R_0$  del coronavirus Ecuador está entre 4.0 pues se puede evidenciar también a nivel real como es el crecimiento del virus

## Conclusiones

Las simulaciones con el valor de  $R_0$  nos ayudan a cuantificar y verificar cuál va a ser el avance de un virus incluso si este va a desaparecer por sí solo o de plano se va a requerir una vacuna incluso si se logra hacer una medición en el tiempo y con datos reales si la vacuna llegaría a en el tiempo necesario o si ya es tarde para la dosificación de la vacuna

## Opinion

A opinión personal estas simulaciones nos ayudan evidenciar casos reales o situaciones reales de los países que han sido más afectados por el coronavirus ya que con estas simulaciones podemos evidenciar los contagiados e incluso cuántas de estas personas van a requerir asistencia médica o de plano cuántas personas van a morir y claramente esto concuerda con los datos obtenidos de Ecuador, cómo se evidencia el porque de los hospitales saturados por casos de COVID.



# Referencias

- <http://code.intef.es/simulamos-una-epidemia-virica/> (<http://code.intef.es/simulamos-una-epidemia-virica/>)
- <https://github.com/jckantor/covid-19/blob/master/COVID-19.ipynb> (<https://github.com/jckantor/covid-19/blob/master/COVID-19.ipynb>)
- <https://cnnespanol.cnn.com/2020/05/18/alerta-ecuador-confirman-caso-de-covid-19-en-comunidad-indigena/> (<https://cnnespanol.cnn.com/2020/05/18/alerta-ecuador-confirman-caso-de-covid-19-en-comunidad-indigena/>)