

CARRERA: Computación**NRO. PRÁCTICA:** 4**ESTUDIANTE:** Doménica Merchán García**ASIGNATURA:** Simulación**TÍTULO:** Simulación del proceso de vacunación en Ecuador**ACTIVIDADES DESARROLLADAS**

```
In [3]: import simpy
import random
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

Se toma uno de los recintos en la parroquia El Batán de Cuenca. En esta zona hay 12738 personas que deberán votar en uno de los tres Recintos Electorales [1]. Suponiendo que todas las personas se reparten equitativamente en los 3 recintos, en cada recinto se atenderán a 4246 personas.

Puesto a que sólo el 80% de la población será vacunada, de las 4246 personas que deberían asistir al recinto, se espera que realmente asistan 3397 personas.

Puesto a que el horario de atención para la vacunación es de 08h00 a 17h00 [2], se define el tiempo de ejecución de la simulación diario en 540 (9 horas en minutos)

```
In [5]: TIEMPO = 540
PUNTOS_VACUNACION0 = 6
PUNTOS_VACUNACION = 6
PUNTOS_VACUNACION2 = int(PUNTOS_VACUNACION/2)
DIAS = 100

vacunados1 = []
no_vacunados1 = []

s1 = 0
n1 = 0

vacunados2 = []
no_vacunados2 = []

s2 = 0
n2 = 0

tiempo_espera = []
tiempo = 0
```

Se crea la clase Centro Covid donde se simularán todos los procesos a llevarse a cabo durante la vacunación de las personas del recinto.

```
In [6]: class Centro_Covid(object):
    def __init__(self, env, puntos_vacunacion):
        self.env = env
        self.puntos_vacunacion = simpy.Resource(env, puntos_vacunacion)

    def poner_vacuna(self, paciente, dosis):
        global s1
        global s2
        global tiempo
        print('> El %s va a vacunarse a las %.2f. %s' % (paciente, env.now, dosis))
        t = random.randint(5,10)
        tiempo += t
        yield self.env.timeout(t)
        print('> El %s ha sido vacunado a las %.2f. %s' % (paciente, env.now, dosis))
        if '1' in dosis:
            s1+=1
        else:
            s2+=1

    def esperar(self, paciente, dosis):
        global tiempo
        print('_ El %s comienza su tiempo de espera a las %.2f. %s' % (paciente, env.now, dosis))
        tiempo += 20
        yield self.env.timeout(20)
        print('_ El %s termina su tiempo de espera a las %.2f. %s' % (paciente, env.now, dosis))
        yield self.env.process(self.certificar(paciente, dosis))

    def certificar(self, paciente, dosis):
        global tiempo
        print('@ El %s va a recibir su certificado a las %.2f. %s' % (paciente, env.now, dosis))
        t = random.randint(2,3)
        tiempo += t
        yield self.env.timeout(t)
        print('@ El %s ha recibido su certificado a las %.2f. %s y sale del recinto' % (paciente, env.now, dosis))
```

Se crea también un método para simular la llegada del paciente al recinto para comenzar su proceso de vacunación.

```
In [7]: def llega_paciente(env, paciente, centro_covid, dosis):
    global n1
    global n2
    print('El %s llega al centro de vacunación a las %.2f. %s' % (paciente, env.now, dosis))
    with centro_covid.puntos_vacunacion.request() as punto_vacunacion:
        if random.randint(1,10) < 9:
            yield punto_vacunacion
            yield env.process(centro_covid.poner_vacuna(paciente, dosis))
            env.process(centro_covid.esperar(paciente, dosis))
        else:
            print('! El %s no ha podido vacunarse la %s y sale del recinto a las %.2f.' % (paciente, dosis, env.now))
            if '1' in dosis:
                n1+=1
            else:
                n2+=1
```

Se inicia la simulación con la llegada del paciente al recinto, realizando la reserva de los recursos y la asignación de tiempos de ejecución.

```
In [8]: def ejecutar(env, puntos_vacunacion, dosis):
    centro_covid = Centro_Covid(env, puntos_vacunacion)
    for i in range(PUNTOS_VACUNACION):
        env.process(llega_paciente(env, 'Paciente %d' %(i+1), centro_covid, dosis))

    while True:
        yield env.timeout(random.randint(3,15))
        i+=1
        env.process(llega_paciente(env, 'Paciente %d' %(i+1), centro_covid, dosis))
```

```
In [9]: random.seed(88)

for i in range(DIAS):
    print()
    print('----- DÍA ' + str(i+1) + ' -----')
    print()

    s1 = 0
    n1 = 0
    s2 = 0
    n2 = 0
    tiempo = 0

    env = simpy.Environment()
    if i >= 30:
        if sum(vacunados2) < 3397:
            env.process(ejecutar(env, PUNTOS_VACUNACION, '2da dosis'))
        if sum(vacunados1) < 3397:
            PUNTOS_VACUNACION = PUNTOS_VACUNACION2
        else:
            PUNTOS_VACUNACION = PUNTOS_VACUNACION0

    if sum(vacunados1) < 3397:
        env.process(ejecutar(env, PUNTOS_VACUNACION, '1ra dosis'))

    env.run(until=TIEMPO)

    vacunados1.append(s1)
    no_vacunados1.append(n1)
    vacunados2.append(s2)
    no_vacunados2.append(n2)
    tiempo_espera.append(tiempo)
```

```
El Paciente 1 llega al centro de vacunacion a las 0.00. 1ra dosis
El Paciente 2 llega al centro de vacunación a las 0.00. 1ra dosis
El Paciente 3 llega al centro de vacunación a las 0.00. 1ra dosis
El Paciente 4 llega al centro de vacunación a las 0.00. 1ra dosis
El Paciente 5 llega al centro de vacunación a las 0.00. 1ra dosis
! El Paciente 5 no ha podido vacunarse la 1ra dosis y sale del recinto a las 0.00.
El Paciente 6 llega al centro de vacunación a las 0.00. 1ra dosis
> El Paciente 1 va a vacunarse a las 0.00. 1ra dosis
> El Paciente 2 va a vacunarse a las 0.00. 1ra dosis
> El Paciente 3 va a vacunarse a las 0.00. 1ra dosis
> El Paciente 4 va a vacunarse a las 0.00. 1ra dosis
> El Paciente 6 va a vacunarse a las 0.00. 1ra dosis
> El Paciente 4 ha sido vacunado a las 6.00. 1ra dosis
> El Paciente 6 ha sido vacunado a las 6.00. 1ra dosis
_ El Paciente 4 comienza su tiempo de espera a las 6.00. 1ra dosis
_ El Paciente 6 comienza su tiempo de espera a las 6.00. 1ra dosis
> El Paciente 1 ha sido vacunado a las 7.00. 1ra dosis
_ El Paciente 1 comienza su tiempo de espera a las 7.00. 1ra dosis
> El Paciente 3 ha sido vacunado a las 8.00. 1ra dosis
_ El Paciente 3 comienza su tiempo de espera a las 8.00. 1ra dosis
```

```
In [14]: print('Cantidad de personas vacunadas con la primera dosis: ', sum(vacunados1))
print('Cantidad de personas no vacunadas durante la primera dosis: ', sum(no_vacunados1))
print()
print('Cantidad de personas vacunadas con la segunda dosis: ', sum(vacunados2))
print('Cantidad de personas no vacunadas durante la segunda dosis: ', sum(no_vacunados2))
print()
print('Tiempo de espera promedio', sum(tiempo_espera)/len(tiempo_espera))
```

Cantidad de personas vacunadas con la primera dosis: 3402
Cantidad de personas no vacunadas durante la primera dosis: 849

Cantidad de personas vacunadas con la segunda dosis: 3405
Cantidad de personas no vacunadas durante la segunda dosis: 826

Tiempo de espera promedio 2046.11

In []: