

```
In [14]: 1 import simpy
2 import math
3 import random
4 import tweepy
5 import pandas as pd
6 import numpy as np
7 import random
8 import matplotlib.pyplot as plt
```

```
In [15]: 1 sufragantes = 6962
2 ausentismo = 1428
3 votar_personas = sufragantes + ausentismo
4 print(votar_personas)
```

8390

```
In [16]: 1 juntasM = 12
2 juntasF = 13
3 juntas = juntasM + juntasF
4 print(juntas)
```

25

- Tomar los resultados de la regresión para la vacuna según la llegada.

-1042.01927509	-744.43750402	-440.29343233	-140.13334007
150.028371	448.19028267	746.35219435	1044.51410602
1342.67601769	1640.83792937	1938.99984104	2237.16175271
2535.32366439	2833.48557606	3131.64748773	3429.80939941
3727.97131108	4026.13322275	4324.29513443	4622.4570461
4920.61895777	5218.78086945	5516.94278112	5815.10469279
6113.26660446	6411.42851614	6709.59042781	7007.75233948
7305.91425116	7604.07616283	7902.2380745	8200.39998618
8498.56189785	8796.72380952	9094.8857212	9393.04763287

8498 vacunas en 29 días

```
In [17]: 1 vacunas_dia = 8498/(29)
2 print(vacunas_dia)
```

293.0344827586207

- Se tiene un promedio que el 80% de personas realizara n el proceso de vacunación dentro del Ecuador.

```
In [18]: 1 promedio80 = (votar_personas * 0.8)
          2 print("80% de personas: ", promedio80)
```

80% de personas: 6712.0

```
In [19]: 1 vacunas = 8498 * 2
          2 print(vacunas)
```

16996

- Dentro del procesos se tiene que alrededor del 5% – 10 % no podrán vacunarse.

```
In [20]: 1 promedio10 = (votar_personas * 0.1)
          2 print("10% de personas: ", promedio10)
```

10% de personas: 839.0

```
In [21]: 1 tiempo_traslado_ambulacion = random.randint(5, 10)
          2 ab = 10
          3 for a in range(10):
          4     a = random.randint(5, 10)
          5     print(a)
          6 print(tiempo_traslado_ambulacion)
```

8  
7  
9  
7  
8  
5  
10  
7  
9  
7  
10

RECUERDA EL HORARIO DE ATENCION EN EL CENTRO DE SALUD ES DE  
5:00 A 24:00

```
In [32]: 1 tiempo_verificar = 20
          2 tiempo_vacunacion = random.randint(5, 10)
```

```
In [35]: 1 tiempo_vacunacion_total = tiempo_vacunacion + tiempo_verificar
          2 print(tiempo_vacunacion_total)
```

25

```
In [36]: 1 # El lugar de vacunacion tiene horario de atencion de 19 horas,
2 horario_atencion = 60 * (19)
3 print(horario_atencion)
```

1140

```
In [37]: 1 # Días de vacuna
2 horario_atencion_completa = horario_atencion * 31
3 print(horario_atencion_completa)
```

35340

```
In [38]: 1 personas_vacunar = horario_atencion/tiempo_vacunacion_total
2 personas_vacunar_dia = personas_vacunar*juntas
3 print(personas_vacunar)
4 print(personas_vacunar_dia)
```

45.6

1140.0

```
In [39]: 1 intervalo_llegada = horario_atencion/vacunas_dia
2 print(intervalo_llegada, "en minutos son " , 3)
3 intervalo_llegada = 3
```

3.890327135796658 en minutos son 3

```
In [40]: aux = 0
atendidos = 0
tiempos_espera = {}
tiempos_llegada = {}
personas_esperando = []
vacunados = 0
no_vacunado = []
horas = []
lista = []
10
11 class Recinto(object):
12     def __init__(self, environment, num_juntas, tiempo_espera):
13         # Guardamos como variable el entorno de ejecución
14         self.env=environment
15         # Creamos la variable que representa el numero de juntas pa
16         self.num_juntas = simpy.Resource(environment, num_juntas)
17         # Tiempo de consulta
18         self.tiempo_espera = tiempo_espera
19
20     # Método que dicta el tiempo que el paciente espera en la fila
21     def espera_paciente(self, paciente):
22         espera = random.randint(tiempo_verificar-intervalo_llegada
23         yield self.env.timeout(espera)
24
25     # Traslado de la ambulancia
26     def traslado_ambulancia(self, paciente):
27         espera = random.randint(5, 10)
28         yield self.env.timeout(espera)
29
30     # Alta medica
```

```

31 def alta_medica(self, paciente):
32     espera = random.randint(60, 120)
33     yield self.env.timeout(espera)
34
35 def no_vacunarse(paciente, promedio10):
36     no_vacuna = random.randint(1,2)
37     global aux
38
39     if(no_vacuna == 1):
40         aux += 1
41         if(aux >= promedio10 ):
42             no_vacuna = 2
43             return no_vacuna
44         else :
45             return no_vacuna
46     else :
47         return no_vacuna
48
49 def llegada_paciente(env, paciente, recinto, no_vacunados):
50     # Usamos el reloj de la simulacion (env.now()) para indicar a
51     # hora que llega el paciente con el nombre pasado como parametro
52
53
54
55     global tiempos_llegada
56     global vacunados
57
58     tiempos_llegada[paciente] = env.now
59
60     print("Llega el paciente "+str(paciente)+" a las "+str(env.now))
61
62     no_puede_vacurse = no_vacunarse(paciente,promedio10)
63
64     # Especificamos que vamos a usar un recurso (Resource) que representa
65     with recinto.num_juntas.request() as lugar:
66         yield env.process(recinto.espera_paciente(paciente))
67         yield lugar
68         global atendidos
69
70     #usamos try para para agarrar una excepcion
71     try:
72         tiempos_espera[paciente] = env.now - tiempos_llegada[paciente]
73     except:
74         a=1
75
76     if no_puede_vacurse == 1:
77
78         print("El paciente ",paciente," no puede vacunarse, se va a casa")
79         #y = paciente.replace("Paciente-", "")
80         yield env.process(recinto.translado_ambulancia(paciente))
81         print("El paciente ",paciente," es traslado en ambulancia")
82         yield env.process(recinto.alta_medica(paciente))
83         print("El paciente ",paciente," alta médica ", (paciente))
84         no_vacunado.append(paciente)
85         horas.append(int(env.now))
86
87

```

```

def ejecutar_simulacion(env, num_juntas, paciente, intervalo):
89     recinto = Recinto(env,num_juntas,tiempo_verificar)
90
91     promedio10 = (votar_personas * 0.1)
92
93     print("Personas que tienen sintomas: ",promedio10,"%")
94
95     for i in range(juntas):
96         env.process(llegada_paciente(env,i,recinto,promedio10))
97
98     while True:
99         yield env.timeout(random.randint(intervalo-intervalo_llega
100         i+=1
101         env.process(llegada_paciente(env,i,recinto,promedio10))
102

```

```

In [42]: 1 # Creamos el entorno de simulacion
2 env=simpy.Environment()
3 print("Pacientes ", votar_personas)
4 env.process(ejecutar_simulacion(env, juntas, votar_personas, in
5 # Ejecutamos el proceso durante el tiempo de simulacion
6
7 env.run(until = horario_atencion)

```

```

Pacientes 8390
Personas que tienen sintomas: 839.0 %
Llega el paciente 0 a las 0
Llega el paciente 1 a las 0
Llega el paciente 2 a las 0
Llega el paciente 3 a las 0
Llega el paciente 4 a las 0
Llega el paciente 5 a las 0
Llega el paciente 6 a las 0
Llega el paciente 7 a las 0
Llega el paciente 8 a las 0
Llega el paciente 9 a las 0
Llega el paciente 10 a las 0
Llega el paciente 11 a las 0
Llega el paciente 12 a las 0
Llega el paciente 13 a las 0
Llega el paciente 14 a las 0
Llega el paciente 15 a las 0
Llega el paciente 16 a las 0
Llega el paciente 17 a las 0

```

- Deben calcularse las siguientes métricas del sistema de simulación de eventos discretos :
  - Total de de personas que realizaron el proceso de vacunación.

```

In [43]: 1 person_vacunas = vacunados
2 print("El total de personas que realizaron el proceso de vacuna

```

```

El total de personas que realizaron el proceso de vacunación: 0

```

- El 10% de las personas que presentan complicaciones fallecen.

In [52]: `1 print("Personas que presentan complicaciones fallecen: ",int(pr`

Personas que presentan complicaciones fallecen: 83

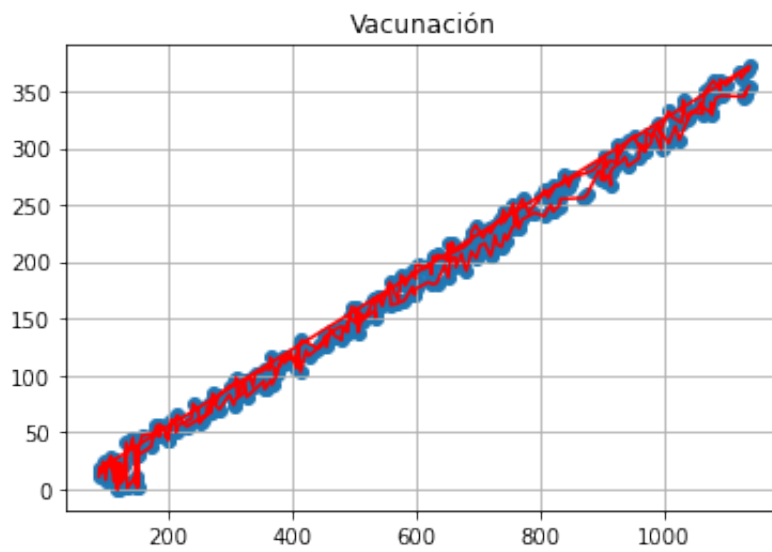
- Grafico del porcentaje de personas que no recibier on la vacuna.

In [44]: `1 dict = {'Horas':horas, 'No_Vacunacion':no_vacunado}  
2 df = pd.DataFrame(dict)  
3 df.to_csv('vacunacion.csv')  
4`

In [45]: `1 no = len(no_vacunado)  
2 si = vacunados  
3 plt.figure(figsize=(8,5))  
4  
5 etiquetas = ['No vacunados', 'Vacunados']  
6 colores = ['#a0522d', '#aabbcc']  
7 plt.pie([no, si, ], labels=etiquetas, colors=colores, autopct='  
8 plt.title('Vacunación')  
9 plt.show()`



```
In [46]: 1 plt.plot(horas,no_vacunado,color='red')
2 plt.scatter(horas,no_vacunado)
3 plt.title("Personas con complicaciones en base a las 24 horas")
4 plt.grid(True)
5 plt.show()
```



```
In [ ]: 1
```