

Importamos Variables

```
In [35]: 1 import simpy
          2 import random
          3 import numpy as np
```

Inicializamos variables

```
In [36]: 1 TIEMPO_POST_VACUNA = 20
          2 MESAS = 5
          3 CONTROL_SIGNOS = 1
          4 SEMANAS = 2
          5 TIEMPO_VACUNACION = 5
          6 TIEMPO_SIMULACION = SEMANAS * 3 * 9 * 60
```

Metodos para la Simulacion

```

In [38]: 1 class Vacunacion():
2     def __init__(self, env, nombre, recinto):
3         self.env = env
4         self.nombre = nombre
5         self.complicacion = False
6         self.control = False
7         self.personas_vacunadas = 0
8         self.personas = 0
9         self.proceso = env.process(self.proceso_vacunar(recinto))
10        env.process(self.complicacion_persona())
11
12    def control_signos(self):
13        yield self.env.timeout(random.randint(CONTROL_SIGNOS, CONTROL_SIGNOS+2))
14
15    def logs_procesos(self, icon, nombre, accion, hora):
16        print(" %s =====nombre: [%s] accion: [%s] hora: [%d]"%(icon,nombre, accion, hora))
17
18    def aplicar_vacuna(self):
19        yield self.env.timeout(random.randint(TIEMPO_VACUNACION, TIEMPO_VACUNACION+5))
20
21    def post_vacuna(self, tiempo):
22        yield self.env.timeout(tiempo)
23
24
25    def proceso_vacunar(self, recinto):
26        while True:
27            try:
28                nombre= 'persona '+str(cont[0])
29                cont[0]=cont[0]+1
30                personas_total.append(nombre)
31                self.logs_procesos("#####",nombre," Llega al recinto",self.env.now)
32                yield env.process(self.control_signos())
33                if random.randint(1,100) > 10:
34                    self.logs_procesos("", nombre, "Pasa el control", env.now)
35                    inicio_vacunacion = self.env.now
36                    yield env.process(self.aplicar_vacuna())
37                    tiempo_vacunacion[nombre] = self.env.now - inicio_vacunacion
38                    self.logs_procesos("", nombre, "Fue vacunada ", self.env.now)
39
40                    tiempo_post_vacuna = TIEMPO_POST_VACUNA
41                    inicio = self.env.now
42                    self.control = True

```

```

43         yield env.process(self.post_vacuna(tiempo_post_vacuna))
44         tiempo_post_vacuna = 0
45         self.logs_procesos("", nombre, "El paciente no presenta complicaciones", self.env.now)
46         self.personas_vacunadas += 1
47     else:
48         self.logs_procesos("", nombre, "No pasa el control", self.env.now)
49     except simpy.Interrupt:
50         self.complicacion = True
51         self.logs_procesos("", nombre, "El paciente presenta complicaciones", self.env.now)
52         tiempo_post_vacuna -= self.env.now - inicio
53         personas_complicaciones.append(nombre)
54         with recinto.request(priority=1) as requerimiento:
55             yield requerimiento
56             env.process(self.proceso_traslado(nombre))
57
58         self.complicacion = False
59
60     def complicacion_persona(self):
61         while True:
62             yield self.env.timeout(25)
63             if random.randint(1,100) <=5:
64                 if not self.complicacion:
65                     self.proceso.interrupt()
66
67     def proceso_traslado(self, nombre):
68         self.logs_procesos(' ', nombre, ' Es trasladado en ambulancia ',self.env.now)
69
70         yield self.env.timeout(random.randint(5,15)) # tiempo traslado en ambulancia
71         self.logs_procesos(' ', nombre, ' El paciente es atendido', self.env.now)
72         yield self.env.timeout(random.randint(300,1200)) # tiempo en ser atendido 5-24 horas
73
74         if random.randint(1,100) <=10:
75             self.logs_procesos(' ',nombre, ' El paciente ha fallecido', self.env.now)
76             personas_muertas.append(nombre)
77         else:
78             yield self.env.timeout(random.randint(60,120)) # tiempo en dar alta 1-2 horas
79             self.logs_procesos('<---', nombre, 'El paciente recibe el alta', self.env.now)
80             personas_altas.append(nombre)

```

Realizamos el porceso de Simulacion

In [39]:

```

1 env = simpy.Environment()
2 recinto = simpy.PreemptiveResource(env, capacity=1)
3 vacunacion = [Vacunacion(env,"Mesa %d"% i, recinto) for i in range(MESAS)]
4 env.run(until=TIEMPO_SIMULACION)
5 print("Simulacion realizada despues de %d semanas"%SEMANAS)
6 for i in vacunacion:
7     print("La [%s] ha vacunado a [%d] personas"%(i.nombre,i.personas_vacunadas))

```

```

nombre: [persona 598] accion: [Pasa el control] hora: [3103]
nombre: [persona 597] accion: [Pasa el control] hora: [3104]
nombre: [persona 599] accion: [Pasa el control] hora: [3104]
nombre: [persona 593] accion: [El paciente no presenta complicaciones] hora: [3106]
##### nombre: [persona 600] accion: [ Llega al recinto] hora: [3106]
nombre: [persona 594] accion: [ El paciente es atendido] hora: [3107]
nombre: [persona 595] accion: [Fue vacunada ] hora: [3108]
nombre: [persona 600] accion: [Pasa el control] hora: [3108]
nombre: [persona 598] accion: [Fue vacunada ] hora: [3109]
nombre: [persona 599] accion: [Fue vacunada ] hora: [3109]

nombre: [persona 597] accion: [Fue vacunada ] hora: [3111]
nombre: [persona 600] accion: [Fue vacunada ] hora: [3118]
<--- nombre: [persona 466] accion: [El paciente recibe el alta] hora: [3121]
nombre: [persona 595] accion: [El paciente no presenta complicaciones] hora: [3128]
##### nombre: [persona 601] accion: [ Llega al recinto] hora: [3128]
nombre: [persona 598] accion: [El paciente no presenta complicaciones] hora: [3129]
##### nombre: [persona 602] accion: [ Llega al recinto] hora: [3129]
nombre: [persona 599] accion: [El paciente no presenta complicaciones] hora: [3129]
##### nombre: [persona 603] accion: [ Llega al recinto] hora: [3129]

```

Resultados del proceso

```
In [44]: 1 complicaciones1 =len(personas_complicaciones)
2 muertes1 =len(personas_muertas)
3 altas1 =len(personas_altas)
4 total=len(personas_total)
5 print("Se vacunaron un total de: ",total)
6 print("Las personas que presentaron complicaiones fueron: ",complicaciones1)
7 print("Las personas que resivieron el alta medica fueron: ",altas1)
8 print("Las personas que fallcieron por las complicaiones fueron: ",muertes1)
```

Se vacunaron un total de: 626

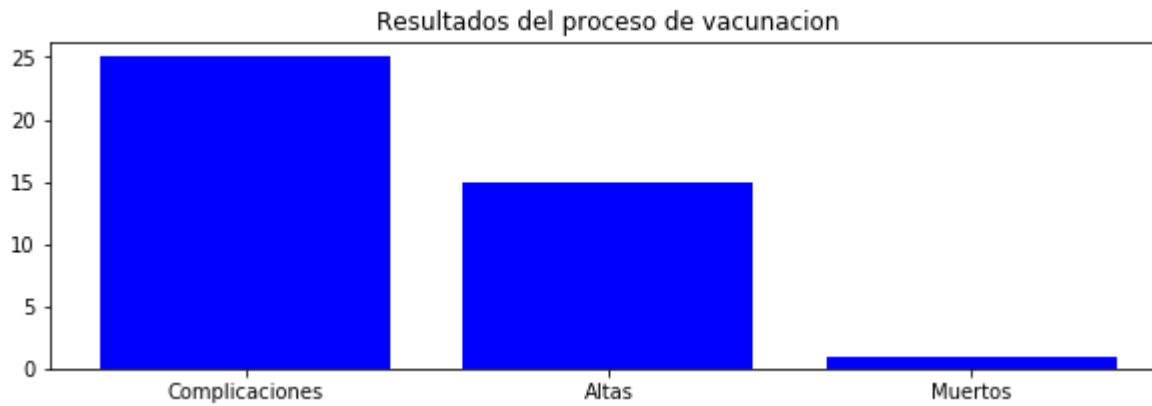
Las personas que presentaron complicaiones fueron: 25

Las personas que resivieron el alta medica fueron: 15

Las personas que fallcieron por las complicaiones fueron: 1

Graficas de los resultados

```
In [41]: 1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize=(10,3))
4 res1 = [len(personas_complicaciones), len(personas_altas), len(personas_muertas)]
5 etiquetas = ['Complicaciones', 'Altas', 'Muertos']
6 plt.bar(etiquetas, res1, color=['blue', 'blue', 'blue'])
7 plt.title('Resultados del proceso de vacunacion')
8 plt.show()
```



```
In [ ]: 1
```

