

Importamos librerias

```
In [63]: 1 import simpy
2 import random
3 import datetime as dt
4 from datetime import datetime
```

Inicializamos Variables

```
In [64]: 1 MESAS = 20
2 TIEMPO_VACUNACION = 5
3 SEMANAS = 4
4 TIEMPO_SIMULACION = SEMANAS * 3 * 9 * 60
5 CONTROL_SIGNOS = 5
6 TIEMPO_POST_VACUNA = 15
7
8 tiempo_vacunacion={}
9 cont={0:1}
10 FECHA_ACTUAL = datetime.now().strftime('%d/%m/%Y')
```

```
In [65]: 1 personas_reassignadas={}
2 tiempo_vacunacion={}
3 estado_personas={}
```

Metodos para la simulacion

```
In [66]: ▶ 1 class VacunasEstado():
2     def __init__(self, env):
3         self.env = env
4         self.dispensador = simpy.Resource(env, capacity=MESAS)
5         self.bodega = simpy.Container(env, init=350, capacity=800)
6         self.monitoreo = env.process(self.monitoreo_bodega())
7     def monitoreo_bodega(self,):
8         while True:
9             if self.bodega.level <= 10:
10                 self.logs_procesos('+++', 'Solicitar nuevas vacunas ', self.env.now)
11                 env.process(self.solicitar_vacunas())
12                 yield self.env.timeout(15)
13     def solicitar_vacunas(self,):
14         yield self.env.timeout(50)
15         self.logs_procesos(' ', 'llega suministro vacunas ', self.env.now)
16         nivel = self.bodega.capacity - self.bodega.level
17         if nivel == 0:
18             nivel += 1
19         yield self.bodega.put(nivel)
20
21     def logs_procesos(self, icon, accion, hora):
22         print(" %s accion: [%s] hora: [%d] "%(icon, accion, hora))
```

```

In [67]: 1 class Vacunacion():
2     def __init__(self, env, nombre, estadoVacunas):
3         self.env = env
4         self.nombre = nombre
5         self.estado_vacunas = estadoVacunas
6         self.proceso = env.process(self.proceso_vacunacion())
7         self.personas_vacunadas = 0
8     def control_signos(self):
9         yield self.env.timeout(random.randint(CONTROL_SIGNOS, CONTROL_SIGNOS+2))
10
11     def logs_procesos(self, icon, nombre, accion, hora):
12         print(" %s nombre: [%s] accion: [%s] hora: [%d]"%(icon, nombre, accion, hora))
13
14     def aplicar_vacuna(self):
15         yield self.env.timeout(random.randint(TIEMPO_VACUNACION, TIEMPO_VACUNACION+5))
16
17     def post_vacuna(self, tiempo):
18         yield self.env.timeout(tiempo)
19
20     def proceso_vacunacion(self,):
21         while True:
22             nombre= 'persona '+str(cont[0])
23             cont[0]=cont[0]+1
24             self.logs_procesos("--->", nombre, " llega al recinto", self.env.now)
25             estado_personas[nombre]='cola'
26             yield env.process(self.control_signos())
27             if random.randint(1,100) > 10:
28                 self.logs_procesos("", nombre, "pasa el control", env.now)
29                 estado_personas[nombre]='pasa control'
30                 inicio_vacunacion = self.env.now
31                 if self.estado_vacunas.bodega.level >=1:
32                     yield env.process(self.aplicar_vacuna())
33                     estado_personas[nombre]='vacunado'
34                     self.estado_vacunas.bodega.get(1)
35                     tiempo_vacunacion[nombre] = self.env.now - inicio_vacunacion
36                     self.logs_procesos("", nombre, "fue vacunada ", self.env.now)
37                     yield env.process(self.post_vacuna(TIEMPO_POST_VACUNA))
38                     self.logs_procesos("", nombre, "sin complicaciones", self.env.now)
39                     self.personas_vacunadas += 1
40                     estado_personas[nombre]='completo'
41
42             else:

```

```

43         prox_cita = datetime.strftime(datetime.strptime(FECHA_ACTUAL, '%d/%m/%Y')+dt.timedelta(days=1), '%d/%m/%Y')
44         print("--XX-- No hay vacunas para %s, reasignado fecha para: %s"%(nombre, prox_cita))
45         personas_reasignadas[nombre]='reasignado'
46         estado_personas[nombre]='reasignado'
47
48     else:
49         self.logs_procesos("", nombre,"no pasa el control", self.env.now)
50         estado_personas[nombre]='no control'

```

In [68]:

```

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

```

```

nombre: [persona 5132] accion: [pasa el control] hora: [6221]
nombre: [persona 5125] accion: [fue vacunada ] hora: [6222]
nombre: [persona 5127] accion: [fue vacunada ] hora: [6224]
nombre: [persona 5128] accion: [fue vacunada ] hora: [6224]
nombre: [persona 5133] accion: [pasa el control] hora: [6224]
nombre: [persona 5134] accion: [pasa el control] hora: [6224]
nombre: [persona 5116] accion: [sin complicaciones] hora: [6225]
---> nombre: [persona 5137] accion: [ llega al recinto] hora: [6225]
nombre: [persona 5117] accion: [sin complicaciones] hora: [6225]
---> nombre: [persona 5138] accion: [ llega al recinto] hora: [6225]
nombre: [persona 5129] accion: [fue vacunada ] hora: [6225]
nombre: [persona 5135] accion: [pasa el control] hora: [6226]
nombre: [persona 5136] accion: [pasa el control] hora: [6226]
nombre: [persona 5131] accion: [fue vacunada ] hora: [6226]
nombre: [persona 5132] accion: [fue vacunada ] hora: [6227]
nombre: [persona 5119] accion: [sin complicaciones] hora: [6229]
---> nombre: [persona 5139] accion: [ llega al recinto] hora: [6229]
nombre: [persona 5130] accion: [fue vacunada ] hora: [6229]
nombre: [persona 5133] accion: [fue vacunada ] hora: [6230]
nombre: [persona 5137] accion: [pasa el control] hora: [6230]

```

Grafica de los resultados

```
In [70]: 1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize=(10,3))
4 res1 = [len(vacunados), len(no_control), len(reasignados)]
5 etiquetas = ['Vacunados', 'No pasaron control', 'Reasignados']
6
7 plt.bar(etiquetas, res1,color=['blue', 'blue', 'blue'])
8
9 plt.title('Resultados del porceso de vacunacion')
10 plt.show()
```



Resultados obtenidos

```
In [69]: 1 no_control=[key for key in list(estado_personas.keys()) if estado_personas[key]=='no control']
2 vacunados=[key for key in list(estado_personas.keys()) if estado_personas[key]=='completo']
3 reasignados=[key for key in list(estado_personas.keys()) if estado_personas[key]=='reasignado']
4 print('Total personas: ', len(estado_personas))
5 print('Personas que no pasaron el control: ', len(no_control))
6 print('Personas vacunadas: ', len(vacunados))
7 print('Personas que fueron reasignados por falta de stock: ', len(reasignados))
```

Total personas: 5332

Personas que no pasaron el control: 483

Personas vacunadas: 4333

Personas que fueron reasignados por falta de stock: 496

In []: ▶ 1

In []: ▶ 1