

# SIMULACION

## VINICIO VELETANGA

### Práctica

```
In [2]: 1 import simpy
2 import random
3 import matplotlib.pyplot as pp
4 import numpy as np
5 %matplotlib inline
6
7
8 #SE PARAMETRIZA LOS VALORES A USAR
9 HOSPITALES = 2
10 HOSPITAL_CAMA_A = 10
11 HOSPITAL_CAMA_B = 10
12 HOSPITAL_VENT_A = 4
13 HOSPITAL_VENT_B = 4
14 INFECTADOS = 20
15 TASA_CRECIMIENTO = 7
16 PERSONAL_MEDICO = 150
17 DIAS_INTERNADO = 10
18 DIAS_SIMULACION = 20
19 persona_recuperadas={}
20
```

```
In [ ]: 1 # Definición de la clase Hospital
```

```
In [4]: 1 class Hospital(object):
2     def __init__(self, env, num_cama, name):
3         self.env = env
4         self.num_cama = num_cama
5         self.camas = simpy.Resource(env, num_cama)
6         self.name = name
7
8     def ingresar_paciente(self, paciente):
9         yield self.env.timeout(random.randint(DIAS_INTERNADO-5,
10         print("El paciente terminó de revisarse: ", paciente, "
11
12
13
14 def ingreso_paciente(env, hospital, paciente):
15     arrive = env.now
16     estado = random.randint(1,100)
17     if (estado < 60):
18         with hospital.camas.request() as cama:
19             dias_esperando = random.randint(1,5)
20             requerimiento = yield cama | env.timeout(dias_esperando)
21             wait = env.now - arrive
22             if cama in requerimiento:
23                 print("Al paciente: ", paciente, " se le asigna
24                 yield env.process(hospital.ingresar_paciente(paciente,
```

```

25         estado = random.randint(1,100)
26         if (estado < 8) :
27             persona_fallecidas[env.now] = persona_falle
28         else:
29             persona_recuperadas[env.now] = persona_recu
30     else:
31         print("El paciente " , paciente, " en el hospita
32         persona_fallecidas[env.now] = persona_fallecidas
33     else :
34         print("El paciente no presenta covid : ", paciente, " h
35
36 def asignar_hospital(env, hospitalA, hospitalB, paciente):
37     hosp_esc = random.randint(1,2)
38     if (hosp_esc == 1):
39         print("Llega el paciente ", paciente, " al hospital 'A'
40         env.process(ingreso_paciente(env, hospitalA, paciente))
41     else:
42         print("Llega el paciente ", paciente, " al hospital 'B'
43         env.process(ingreso_paciente(env, hospitalB, paciente))
44
45 def ejecutar(env, tasa_crecimiento, infectados):
46     hospitalA = Hospital(env, HOSPITAL_CAMA_A, "A")
47     hospitalB = Hospital(env, HOSPITAL_CAMA_B, "B")
48     for i in range(infectados):
49         asignar_hospital(env, hospitalA, hospitalB, i)
50     paciente = infectados
51     while True:
52         yield env.timeout(1)
53         for i in range(tasa_crecimiento):
54             paciente += 1
55         asignar_hospital(env, hospitalA, hospitalB, paciente)

```

```

In [6]: 1 print("Simulación-Vinicio Veletanga")
2 env=simpy.Environment()
3 env.process(ejecutar(env,TASA_CRECIMIENTO, INFECTADOS))
4 env.run(until=DIAS_SIMULACION)
5
6
7
8 print("Pacientes recuperados:")
9 print(persona_recuperadas)
10 print("Pacientes fallecidos:")
11 print(persona_fallecidas)
12
13

```

Simulación-Vinicio Veletanga

Llega el paciente 0 al hospital 'B' en el tiempo 0

Llega el paciente 1 al hospital 'A' en el tiempo 0

```
In [7]: 1 datos=sorted(persona_recuperadas.items())
2 x, y =zip(*datos)
3 pp.figure(figsize=(15,10))
4 pp.plot(x,y,linewidth=2,color='red')
5 pp.scatter(x,y,color='blue')
6 pp.title("Pacientes Recuperados",color="red",size=20)
7 pp.grid(True)
8 pp.show()
9
10 if (persona_fallecidas):
11     datos=sorted(persona_fallecidas.items())
12     pp.figure(figsize=(15,10))
13     x, y =zip(*datos)
14     pp.plot(x,y,linewidth=2,color='red')
15     pp.scatter(x,y,color='black')
16     pp.title("Pacientes Fallecidos",color="red",size=20)
17     pp.grid(True)
18     pp.show()
```



