

Practica No 11: Practica COVID 19 - Interrupción

Nombre:

Fernando Sanchez

Enunciado:

Simulacion de eventos discretos del COVID-19

Datos:

Datos de la provincia de Imbabura: Total: 3 hospitales, 11 camas, 11 respiradores, 11 equipos de radiología, personal 150, 1.13% de camas por cada 1000 personas.

Hospital 1: Camas 7, respiradores 7, 7 equipos de radiología, 0 ventiladores

Hospital 2: Camas 4, respiradores 4, 4 ventiladores

Hospital 3: (IESS no se tiene información)

Datos estadísticos:

Tasa de crecimiento: 7 (Modelo de probabilidad)

Infectados actuales: 99

Modelo Propuesto

Generar un modelo del uso de los recursos de los hospitales para la atención de las personas que tienen el COVID-19 en base a los datos presentados, por cada uno de los días.

Practica

En base a la practica del COVID vacunación, realizar el proceso de interrupción en base a que el 5% de los pacientes que estan en la sala de espera presentan complicaciones de salud y deben ser trasladados inmediatamente al centro de salud más cercano y ser atendidos . Realizar este proceso y generar una segunda linea de simulación para el traslado, atencion y alta del paciente que presenta complicaciones, en base a los siguientes datos:

- * Traslado de la ambulancia: 5 – 15 minutos.
- * Atención centro de salud: 5 – 24 horas
- * Alta: 1 – 2 horas
- * El 10% de las personas que presentan complicaciones fallecen.

Generar graficas que indiquen las personas que presentaron complicaciones en base a los tiempos, estado de las personas y respuesta.

```
In [2]: 1 # Importar las librerias para el analisis
        2 import simpy
        3 import random
        4 import matplotlib.pyplot as pp
        5 import numpy as np
        6
```

```
In [3]: 1 #PARAMETROS
        2 HOSPITALES = 3
        3 HOSPITAL_CAMA_A = 7
        4 HOSPITAL_CAMA_B = 4
        5 HOSPITAL_VENT_A = 7
        6 HOSPITAL_VENT_B = 4
        7 INFECTADOS = 20
        8 TASA_CRECIMIENTO = 7
        9 PERSONAL_MEDICO = 150
       10 DIAS_INTERNADO = 10
       11 DIAS_SIMULACION = 20
       12
       13 #Diccionario para almacenar los resultados
       14 persona_recuperadas={}
       15 persona_fallecidas={}
       16
```

```

In [4]: 1 class Hospital(object):
2         #constructor
3         def __init__(self, env, num_cama, name):
4             self.env = env
5             self.num_cama = num_cama
6             self.camas = simpy.Resource(env, num_cama)
7             self.name = name
8
9         def ingresar_paciente(self, paciente):
10            yield self.env.timeout(random.randint(DIAS_INTERNADO-5,
11            print("El paciente se termino de atender: ", paciente,
12
13 def llegada_paciente(env, hospital, paciente):
14     arrive = env.now
15     estado = random.randint(1,100)
16     if (estado < 60):
17         with hospital.camas.request() as cama:
18             dias_esperando = random.randint(1,5)
19             requerimiento = yield cama | env.timeout(dias_esper
20             wait = env.now - arrive
21             if cama in requerimiento:
22                 print("Al paciente: ", paciente, " se le asigna
23                 yield env.process(hospital.ingresar_paciente(pa
24                 estado = random.randint(1,120)
25                 if (estado < 8) :
26                     persona_fallecidas[env.now] = persona_falle
27                 else:
28                     persona_recuperadas[env.now] = persona_recu
29             else:
30                 print("El paciente " , paciente, " en el hospit
31                 persona_fallecidas[env.now] = persona_fallecida
32     else :
33         print("El paciente no tiene COVID : ", paciente, " hosp

```

```

In [14]: 1 def ejecutar(env, tasa_crecimiento, infectados):
2         hospitalA = Hospital(env, HOSPITAL_CAMA_A, "A")
3         hospitalB = Hospital(env, HOSPITAL_CAMA_B, "B")
4         for i in range(infectados):
5             asignar_hospital(env, hospitalA, hospitalB, i)
6         paciente = infectados
7         while True:
8             yield env.timeout(1)
9             for i in range(tasa_crecimiento):
10                paciente += 1
11                asignar_hospital(env, hospitalA, hospitalB, pacient
12
13 def asignar_hospital(env, hospitalA, hospitalB, paciente):
14     hosp_esc = random.randint(1,2)
15     if (hosp_esc == 1):
16         print("Llega paciente nuevo : ", paciente, " hospital A
17         env.process(llegada_paciente(env, hospitalA, paciente))
18     else:
19         print("Llega paciente nuevo : ", paciente, " hospital B
20         env.process(llegada_paciente(env, hospitalB, paciente))

```

```
In [15]: 1 print("Practica COVID 19 - Interrupción")
2 env=simpy.Environment()
3 env.process(ejecutar(env,TASA_CRECIMIENTO, INFECTADOS))
4 env.run(until=DIAS_SIMULACION)
5
```

Practica COVID 19 - Interrupción

```
Llega paciente nuevo : 0 hospital B tiempo 0
Llega paciente nuevo : 1 hospital A tiempo 0
Llega paciente nuevo : 2 hospital A tiempo 0
Llega paciente nuevo : 3 hospital A tiempo 0
Llega paciente nuevo : 4 hospital A tiempo 0
Llega paciente nuevo : 5 hospital B tiempo 0
Llega paciente nuevo : 6 hospital B tiempo 0
Llega paciente nuevo : 7 hospital A tiempo 0
Llega paciente nuevo : 8 hospital B tiempo 0
Llega paciente nuevo : 9 hospital A tiempo 0
Llega paciente nuevo : 10 hospital A tiempo 0
Llega paciente nuevo : 11 hospital B tiempo 0
Llega paciente nuevo : 12 hospital A tiempo 0
Llega paciente nuevo : 13 hospital B tiempo 0
Llega paciente nuevo : 14 hospital A tiempo 0
Llega paciente nuevo : 15 hospital B tiempo 0
Llega paciente nuevo : 16 hospital A tiempo 0
Llega paciente nuevo : 17 hospital A tiempo 0
Llega paciente nuevo : 18 hospital A tiempo 0
```

```
In [8]: 1 print(" Resultados ")
2 print("Resultados pacientes :")
3 print("Recuperados: ")
4 print(persona_recuperadas)
5 print("Fallecidos: ")
6 print(persona_fallecidas)
```

Resultados

Resultados pacientes :

Recuperados:

```
{5: 2, 7: 3, 9: 2, 10: 1, 11: 2, 13: 3, 15: 2, 16: 2, 18: 3, 19: 3
, 12: 3, 17: 1}
```

Fallecidos:

```
{2: 3, 3: 3, 4: 4, 5: 8, 6: 5, 7: 3, 8: 10, 9: 4, 10: 9, 11: 7, 12
: 7, 13: 4, 14: 5, 15: 6, 16: 12, 17: 11, 18: 3, 19: 8, 1: 1}
```

```
In [11]: 1 datos=sorted(persona_recuperadas.items()) # Ordenamos los datos
2 x, y =zip(*datos) # Obtener x(tiempo - clave) y el y(Numero de
3 pp.plot(x,y,linewidth=2,color='blue') #Dibujamos las lineas
4 pp.scatter(x,y,color='r') # Dibujamos los puntos (x,y)
5 pp.title("Dias / Personas Recuperadas")
6 pp.grid(True) #Generamos una cuadrícula
7 pp.show() #Mostramos el grafico
```



```
In [13]: 1 if (persona_fallecidas):
2     datos=sorted(persona_fallecidas.items()) # Ordenamos los da
3     x, y =zip(*datos) # Obtener x(tiempo - clave) y el y(Numero
4     pp.plot(x,y,linewidth=2,color='b') #Dibujamos las lineas
5     pp.scatter(x,y,color='r') # Dibujamos los puntos (x,y)
6     pp.title("Personas Fallecidas / dias de hospitalizacion")
7     pp.grid(True) #Generamos una cuadrícula
8     pp.show() #Mostramos el grafico
```



```
In [ ]: 1
```

