

# Practica No 4: Vacunación

**Nombre:**

**Fernando Sanchez**

**Enunciado:**

**El proceso de vacunacion es el siguiente:**

- La persona llega 20 minutos antes de la hora de vacuna.
- Ingresa a la cola/fila a que tome el turno.
- Tenemos un proceso de control en donde se toma los signos vitales y se valida que este dentro del patron de (2-5 minutos)
- Como ejemplificación menos del 5% no es vacunado ya que p resenta signo de alerta.
- Se procede al traslado al cama/silla de vacunación y se i nyecta la dosis de la vacuna este proceso tarda entre (5-10 minutos).
- Finalmente son llevados a la salida entre (1-3 minutos)

**Finalmente generar las siguientes metricas/graficas:**

- Tiempo de espera promedio para cada persona.
- Tiempo promedio total que requiere un paciente para ser v acunado.
- Cuántos personas se atendieron.
- Cuántos pacientes se quedaron sin atender.

**\* Puntos extras**

- Vincular su investigación con la llegada y distribución d e la vacuna.

In [49]:

```
1 # Importar librerias necesarias
2 import simpy
3 import random
4 import matplotlib.pyplot as plt
```

In [157]:

```
1 aux = 0
2 atendidos = 0
3 # Maximo de vacunas que puede recibir el negocio
4 MAX_VACUNAS = 200
5 # Total de numero de mesas para vacunacion
```

```

6 NUM_MESAS = 10
7 # Total de pacientes para vacunarse
8 NUM_PACIENTES = 200
9 promedio10 = (NUM_PACIENTES * 0.05)
10 # Intervalo de tiempo en que la persona llega 20 minutos
11 INTERVALO_LLEGADA = 20 #minutos
12 # Tiempo de simulación, el lugar de vacunacion tiene horario de
13 horario_atencion = 60 * (9)
14 TIEMPO_SIMULACION = horario_atencion
15 # Intervalo de llegada
16 INTERVALO = 3
17 tiempos_llegada = {}
18
19 # Diccionario para almacenar tiempo de espera de vacunacion de p
20 tiempo_espera_vacunacion=[]
21 # Diccionario para almacenar tiempo promedio total de vacunacion
22 tiempo_vacunacion={}
23 #Diccionario para el numero de personas vacunadas
24 vacunados = []
25 #Diccionario para el numero de personas no vacunadas
26 no_vacunados = []
27
28 horas = []
29
30
31 class Recinto(object):
32     def __init__(self, environment, num_juntas):
33         # Guardamos como variable el entorno de ejecución
34         self.env=environment
35         # Creamos la variable que representa el numero de juntas
36         self.num_juntas = simpy.Resource(environment, num_juntas)
37
38         # Método que dicta el tiempo que el paciente espera en la fi
39         def control_signos(self, paciente):
40             espera = random.randint(2, 5)
41             yield self.env.timeout(espera)
42
43         # Método que dicta el tiempo que el paciente espera la vacun
44         def espera_vacunacion (self, paciente):
45             espera = random.randint(5, 10)
46             yield self.env.timeout(espera)
47
48             k = int(paciente)
49             if k in tiempo_vacunacion:
50                 tiempo_vacunacion[k]=tiempo_vacunacion[k]+1
51             else:
52                 tiempo_vacunacion[k]=espera
53
54         # Método que dicta la salida
55         def espera_salida(self, paciente):
56             espera = random.randint(1, 3)
57             yield self.env.timeout(espera)
58
59
60 def no_vacunarse(paciente, promedio10):
61     no_vacuna = random.randint(1,2)
62     global aux
63

```

```

64     if(no_vacuna == 1):
65         aux += 1
66         if(aux >= promedio10 ):
67             no_vacuna = 2
68             return no_vacuna
69         else :
70             return no_vacuna
71     else :
72         return no_vacuna
73
74
75 def llegada_paciente(env, paciente, recinto, no_vacunado):
76     # Usamos el reloj de la simulacion (env.now()) para indicar
77     # hora que llega el paciente con el nombre pasado como parame
78
79
80     global tiempos_llegada
81     global timeentry
82     global timeexit
83
84     tiempos_llegada[paciente] = env.now
85     timeentry = paciente,tiempos_llegada[paciente];
86
87     tiempo_espera_vacunacion.append(timeentry)
88
89     print("Llega el paciente "+str(paciente)+" a las "+str(env.now))
90
91     no_puede_vacurse = no_vacunarse(paciente,promedio10)
92
93     # Especificamos que vamos a usar un recurso (Resource) que i
94     with recinto.num_juntas.request() as lugar:
95         yield env.process(recinto.control_signos(paciente))
96
97         yield lugar
98         global atendidos
99
100     #usamos try para para agarrar una excepcion
101     try:
102         tiempos_espera[paciente] = env.now - tiempos_llegada
103     except:
104         a=1
105
106     if no_puede_vacurse == 2:
107         yield env.process(recinto.espera_vacunacion(paciente))
108         #yield env.timeout(INTERVALO_LLEGADA)
109         print("El paciente ",paciente," se verifica que no t
110         atendidos+=1
111         vacunados.append(paciente)
112
113         yield env.process(recinto.espera_salida(paciente))
114         timeexit = (paciente,env.now);
115         print("El paciente ",paciente," sale del recinto ",
116
117         tiempo_espera_vacunacion.append(timeexit)
118
119
120     elif no_puede_vacurse == 1:

```

```

121         print("El paciente ",paciente," no puede vacunarse,
122         #y = paciente.replace("Paciente-", "")
123         no_vacunados.append(paciente)
124         horas.append(int(env.now))
125
126
127 def ejecutar_simulacion(env, num_juntas, paciente, intervalo):
128     recinto = Recinto(env,num_juntas)
129     promedio10 = (NUM_PACIENTES* 0.05)
130
131     print("Personas que tienen sintomas: ",promedio10,"%")
132
133     for i in range(NUM_MESAS):
134         env.process(llegada_paciente(env,i,recinto,promedio10))
135
136     while True:
137         yield env.timeout(random.randint(intervalo-INTERVALO, in
138         i+=1
139         env.process(llegada_paciente(env,i,recinto,promedio10))
140

```

In [158]:

```

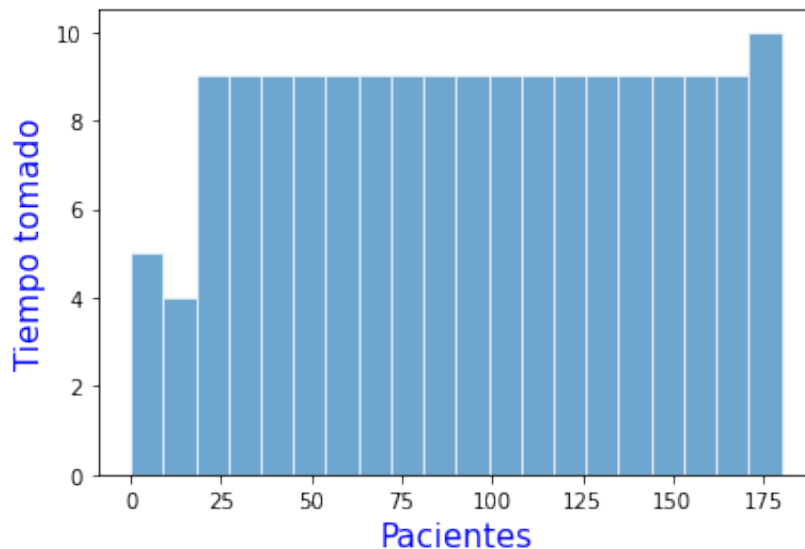
1  # Creamos el entorno de simulacion
2  env=simpy.Environment()
3  print("Pacientes ", NUM_PACIENTES)
4  env.process(ejecutar_simulacion(env, NUM_MESAS, NUM_PACIENTES,
5  # Ejecutamos el proceso durante el tiempo de simulacion
6
7  env.run(until = TIEMPO_SIMULACION)

```

')  
 El paciente 2 no puede vacunarse, sale del recinto a las (2, '3  
 ')  
 El paciente 9 no puede vacunarse, sale del recinto a las (9, '3  
 ')  
 Llega el paciente 11 a las 4  
 El paciente 6 no puede vacunarse, sale del recinto a las (6, '4  
 ')  
 El paciente 4 no puede vacunarse, sale del recinto a las (4, '5  
 ')  
 Llega el paciente 12 a las 8  
 El paciente 11 no puede vacunarse, sale del recinto a las (11,  
 '8')  
 El paciente 5 se verifica que no tenga problemas de salud por la  
 vacuna (5, '9')  
 El paciente 8 se verifica que no tenga problemas de salud por la  
 vacuna (8, '10')  
 El paciente 5 sale del recinto (5, '10')  
 El paciente 10 se verifica que no tenga problemas de salud por l  
 a vacuna (10, '11')

```
In [83]: 1 # Generamos la grafica
2 datos=sorted(tiempo_vacunacion.items())
3 x, y =zip(*datos)
4 result = pp.hist(x,bins=20,edgecolor='w',alpha=0.65)
5 pp.xlabel("Pacientes",fontsize = 15,color = 'b')
6 pp.ylabel("Tiempo tomado",fontsize = 15,color = 'b')
```

Out[83]: Text(0, 0.5, 'Tiempo tomado')



```
In [181]: 1 aux = 0
2 for val in tuple(tiempo_vacunacion.values()):
3     aux = aux+val
4 print("Tiempo de espera promedio vacunación ",aux/len(tiempo_vacuacion))

Tiempo de espera promedio vacunación  7.511627906976744
```

```
In [182]: 1 tiempo_espera_vacunacion1=[]
2 for val in tiempo_espera_vacunacion:
3     for val1 in tiempo_espera_vacunacion:
4         if(val[0] == val1[0] ):
5             if(val[1] != val1[1] ):
6                 if(val[1]<val1[1]):
7                     tiempo_espera_vacunacion1.append(val1[1]-val[1])
```

```
In [183]: 1 aux = 0
2 for val in tiempo_espera_vacunacion1:
3     aux = aux+val
4 print("Tiempo promedio total de vacunación ",suma/len(tiempo_espera_vacunacion1))

Tiempo promedio total de vacunación  12.91812865497076
```

```
In [184]: 1 print("Pacientes atendidos ", len(vacunados))
2 print("Pacientes no atendidos ", len(no_vacunados))

Pacientes atendidos  172
Pacientes no atendidos  9
```

