



El proceso de vacunacion es el siguiente.

- La persona llega 20 minutos antes de la hora de vacunacion
- Ingresa a la cola/fila a que toque su turno
- Tenemos un proceso de control en donde se toma los signos vitales y se valida que este dentro del padron (2-5 minutos)
- Se procede al traslado a la cama/silla de vacunacion y se inyecta la dosis de la vacuna este proceso tarde entre (5-10 minutos).
- Finalmente 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 vacunado.
- Cuantos pacientes se atendieron.
- Cuantos pacientes se quedaron sin atende

```
import simpy
import random
import matplotlib.pyplot as plt

# Maximo de vehiculos que puede recibir el negocio
MAX_VACUNAS = 57
# Total de numero de mesas para vacunacion
NUM_MESAS = 3
PACIENTES = 400
# Intervalo de tiempo en que llegan vehiculos (minutos)
INTERVALO_LLEGADA = 20 #minutos
# Tiempo de simulación
TIEMPO_SIMULACION = 480 #Minutos = 8 horas
INTERVALO = 10

# Creamos un diccionario para almacenar las horas en que se vacuna los pacientes
tiempo_vacunacion={}
#Diccionario para el numero de pacientes que han sido atendidos
p_vacunado = []
#Diccionario para el numero de pacientes que no han sido atendidos
p_no_vacunado = []

class Centro(object):
    def __init__(self, environment, num_mesas, nombre):
        # Guardamos como variable el entorno de ejecucion
        self.env=environment
        # Creamos el recurso que representa las maquinas
        self.mesas = simpy.Resource(environment, num_mesas)
        self.nombre = nombre
```

```

#Proceso de control, para toma de signos vitales y valida estar dentro del padron
def control(self, paciente):
    t_con = random.randint(2,5) #Tenemos un proceso de control en donde se toma los si
    yield self.env.timeout(t_con)

#Para traslado a la cama/silla de vacunacion y se inyecta
def vacunacion(self, paciente):
    t_vacunacion = random.randint(5,10) #Se procede al traslado a la cama/silla de vac
    print("Tiempo tomado: ",t_vacunacion," para paciente: ",paciente)
    yield self.env.timeout(t_vacunacion)

    k = int(paciente.strip('Paciente-'))
    print("Paciente ",k)
    if k in tiempo_vacunacion:
        tiempo_vacunacion[k]=tiempo_vacunacion[k]+1
    else:
        tiempo_vacunacion[k]=t_vacunacion

#Traslado para la salida del paciente
def salida(self,paciente):
    t_salida = random.randint(1,3) #Finalmente llevados a la salida entre (1-3 minutos
    yield self.env.timeout(t_salida)

def llegada(env, nombre, centro):
    # Usamos el reloj de la simulacion (env.now()) para indicar a la
    # hora que llega el vehiculo con el nombre pasado como parametro
    print('Llega el paciente: %s a la hora %.2f.' % (nombre, env.now))

    # Especificamos que vamos a usar un recurso (Resource) que representa
    # la maquina de lavado
    with centro.mesas.request() as mesa:
        # Ocupamos la mesa de vacunacion
        yield mesa
        #Indicamos que el paciente se esta trasladando para ser vacunado
        print('Se traslada el paciente: %s a la hora %.2f.' % (nombre,env.now))
        #Procesamos el traslado del paciente
        yield env.process(centro.control(nombre))

    estado = random.randint(1,2)
    print("Sintomas: ",estado)
    if estado == 1:
        p_no_vacunado.append(nombre)
    else:
        # Indicamos que paciente ingresa a la mesa de vacunacion
        print('Ingresa el paciente a las %s a la hora %.2f.' % (nombre, env.now))

        # Se realiza la vacunacion
        yield env.process(centro.vacunacion(nombre))
        # Una vez que termina la llamada con 'yield', se indica que se ha lavado el ve
        print('Paciente [%s] vacunado a las %.2f.' % (nombre, env.now))
        p_vacunado.append(nombre)

    yield env.process(centro.salida(nombre))

```

```

print('Paciente [%s] trasladado a la salida a las %.2f.' %(nombre,env.now))

def ejecucion(env, num_mesas, pacientes, intervalo):
    centro = Centro(env,num_mesas,"Coliseo UPS")
    for i in range(pacientes):
        env.process(llegada(env,'Paciente-%d'%(i+1), centro))
    while True:
        yield env.timeout(random.randint(intervalo-10,intervalo+10))
        i+=1
        env.process(llegada(env,'Paciente-%d'%(i+1), centro))

# Creamos el entorno de simulacion
env=simpy.Environment()
print("pacientes ", PACIENTES)
env.process(ejecucion(env, NUM_MESAS, PACIENTES, INTERVALO))
# Ejecutamos el proceso durante el tiempo de simulacion
env.run(until = TIEMPO_SIMULACION)

```

```

pacientes 400
Llega el paciente: Paciente-1 a la hora 0.00.
Llega el paciente: Paciente-2 a la hora 0.00.
Llega el paciente: Paciente-3 a la hora 0.00.
Llega el paciente: Paciente-4 a la hora 0.00.
Llega el paciente: Paciente-5 a la hora 0.00.
Llega el paciente: Paciente-6 a la hora 0.00.
Llega el paciente: Paciente-7 a la hora 0.00.
Llega el paciente: Paciente-8 a la hora 0.00.
Llega el paciente: Paciente-9 a la hora 0.00.
Llega el paciente: Paciente-10 a la hora 0.00.
Llega el paciente: Paciente-11 a la hora 0.00.
Llega el paciente: Paciente-12 a la hora 0.00.
Llega el paciente: Paciente-13 a la hora 0.00.
Llega el paciente: Paciente-14 a la hora 0.00.
Llega el paciente: Paciente-15 a la hora 0.00.
Llega el paciente: Paciente-16 a la hora 0.00.
Llega el paciente: Paciente-17 a la hora 0.00.
Llega el paciente: Paciente-18 a la hora 0.00.
Llega el paciente: Paciente-19 a la hora 0.00.
Llega el paciente: Paciente-20 a la hora 0.00.
Llega el paciente: Paciente-21 a la hora 0.00.
Llega el paciente: Paciente-22 a la hora 0.00.
Llega el paciente: Paciente-23 a la hora 0.00.
Llega el paciente: Paciente-24 a la hora 0.00.
Llega el paciente: Paciente-25 a la hora 0.00.
Llega el paciente: Paciente-26 a la hora 0.00.
Llega el paciente: Paciente-27 a la hora 0.00.
Llega el paciente: Paciente-28 a la hora 0.00.
Llega el paciente: Paciente-29 a la hora 0.00.
Llega el paciente: Paciente-30 a la hora 0.00.
Llega el paciente: Paciente-31 a la hora 0.00.
Llega el paciente: Paciente-32 a la hora 0.00.
Llega el paciente: Paciente-33 a la hora 0.00.
Llega el paciente: Paciente-34 a la hora 0.00.
Llega el paciente: Paciente-35 a la hora 0.00.
Llega el paciente: Paciente-36 a la hora 0.00.
Llega el paciente: Paciente-37 a la hora 0.00.
Llega el paciente: Paciente-38 a la hora 0.00.
Llega el paciente: Paciente-39 a la hora 0.00.

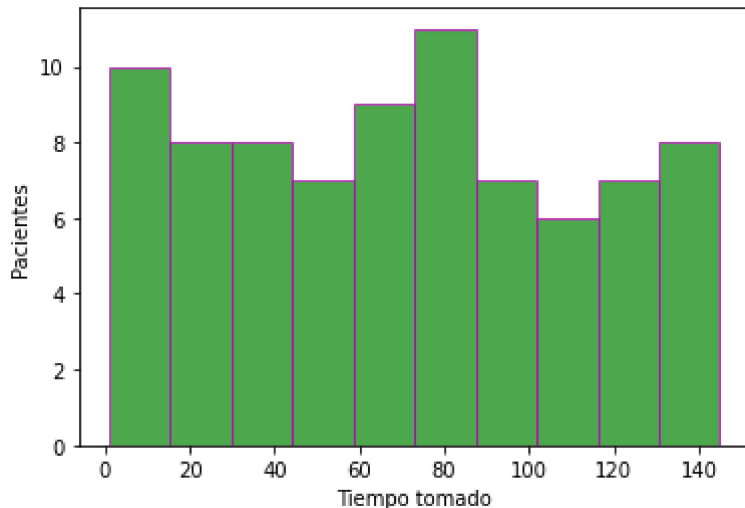
```

```
Llega el paciente: Paciente-40 a la hora 0.00.
Llega el paciente: Paciente-41 a la hora 0.00.
Llega el paciente: Paciente-42 a la hora 0.00.
Llega el paciente: Paciente-43 a la hora 0.00.
Llega el paciente: Paciente-44 a la hora 0.00.
Llega el paciente: Paciente-45 a la hora 0.00.
Llega el paciente: Paciente-46 a la hora 0.00.
Llega el paciente: Paciente-47 a la hora 0.00.
Llega el paciente: Paciente-48 a la hora 0.00.
Llega el paciente: Paciente-49 a la hora 0.00.
Llega el paciente: Paciente-50 a la hora 0.00.
Llega el paciente: Paciente-51 a la hora 0.00.
Llega el paciente: Paciente-52 a la hora 0.00.
Llega el paciente: Paciente-53 a la hora 0.00.
Llega el paciente: Paciente-54 a la hora 0.00.
Llega el paciente: Paciente-55 a la hora 0.00.
Llega el paciente: Paciente-56 a la hora 0.00.
Llega el paciente: Paciente-57 a la hora 0.00.
Llega el paciente: Paciente-58 a la hora 0.00.
```

```
# SE CREA LA GRAFICA
```

```
datos = sorted(tiempo_vacunacion.items())
print(datos)
x, y = zip(*datos)
result = plt.hist(x,color='green',edgecolor='m',alpha=0.7)
plt.ylabel("Pacientes")
plt.xlabel("Tiempo tomado")
```

```
[(1, 10), (2, 10), (4, 9), (5, 6), (8, 8), (9, 10), (10, 8), (11, 5), (13, 6), (15, 8), (17, 11), (19, 7), (21, 6), (23, 7), (25, 8)]
Text(0.5, 0, 'Tiempo tomado')
```



```
suma = 0
for val in tuple(tiempo_vacunacion.values()):
    suma = suma+val
print("Promedio de tiempo vacunacion",suma/len(tiempo_vacunacion))
```

```
Promedio de tiempo vacunacion 7.265822784810126
```

```
print("Total de pacientes que llegaron ",PACIENTES)
print("Vacunados ", len(p_vacunado))
print("No vacunados ", len(p_no_vacunado))
```

```
n_atendidos = PACIENTES-(len(p_vacunado)+len(p_no_vacunado))  
print("Pacientes no atendidos",n_atendidos)
```

```
Total de pacientes que llegaron 400  
Vacunados 79  
No vacunados 76  
Pacientes no atendidos 245
```

