


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: Computación		ASIGNATURA: Simulación	
NRO. PRÁCTICA:		TÍTULO PRÁCTICA: Practica Vacunación	
OBJETIVO ALCANZADO: comprensión práctica de las herramientas de simulación existentes			
ACTIVIDADES DESARROLLADAS			
1. Importamos las librerías necesarias, en este caso utilizaremos simpy <pre> # Imports import simpy import random import pandas as pd import numpy as np from matplotlib import pyplot as plt </pre> <p>Estas son las librerías que vamos a usar para esta simulación</p>			
2. Empezamos por instanciar las variables iniciales <pre> # Instanciar Variables NUM_MEDICOS = 6 NUM_PACIENTES = 150 TIEMPO_CONTROL = 2 # Hasta 5 TIEMPO_VACUNACION = 5 # Hasta 10 INTERVALO_LLEGADA = 15 POST_VACUNA = 1 # Hasta 3 ESPERA_INICIAL = 20 </pre> <p>Así mismo vamos a necesitar las variables que van a almacenar el proceso de la simulación, con el fin de obtener datos posibles de graficar al final.</p> <pre> vacunados = 0 no_vacunados = 0 pacientes = 0 pacientes_atendidos = 0 bandera = 1 tiempo = 0 proceso = pd.DataFrame(columns=['pacientes', 'vacunados', 'no_vacunados']) </pre>			
3. Creamos la clase inicial de la simulación <pre> class Centro_Vacunacion(object): def __init__(self, environment, num_medicos, num_pacientes): self.env = environment self.num_medicos = simpy.Resource(environment, num_medicos) self.num_pacientes = simpy.Resource(environment, num_pacientes) def llega_paciente(self, paciente): print("El %s llega a las %.2f. " % (paciente, env.now)) yield self.env.timeout(ESPERA_INICIAL) print("El %s entra a la fila a las %.2f. " % (paciente, env.now)) </pre> <p>Para iniciar el funcionamiento nuestro centro de vacunación necesitará médicos como recurso y pacientes.</p> <pre> def llega_paciente(self, paciente): print("El %s llega a las %.2f. " % (paciente, env.now)) yield self.env.timeout(ESPERA_INICIAL) print("El %s entra a la fila a las %.2f. " % (paciente, env.now)) </pre> <p>Así mismo tendremos nuestro primer método, que se encarga de la llegada del paciente, marca el tiempo y realiza la espera inicial que tienen los pacientes.</p>			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
def vacunar(self, paciente):
    global bandera
    global no_vacunados
    complicaciones = random.randint(1,100)
    print("El %s pasa a revisión en el padrón a las %.2f. " % (paciente, env.now))
    yield self.env.timeout(random.randint(TIEMPO_CONTROL, TIEMPO_CONTROL+3))
    if complicaciones < 95:
        print("El %s empieza la vacunación a las %.2f. " % (paciente, env.now))
        yield self.env.timeout(random.randint(TIEMPO_VACUNACION, TIEMPO_VACUNACION+5))
        print("El %s termina la vacunación a las %.2f. " % (paciente, env.now))
        bandera = 1
    else:
        print("!X!X!X! El %s no puede vacunarse y se retira a las %.2f." % (paciente, env.now))
        no_vacunados = no_vacunados+1
        bandera = 0
```

Procede a vacunarse, tiempo en el que se realiza la revisión del padrón y se verifican complicaciones por medio de un número aleatorio, que el 5% no podrá vacunarse. Si no puede vacunarse se suma el contador de los no vacunados y se define la bandera como false para evitar que haga el procedimiento post-vacunación.

```
def salida(self, paciente):
    global bandera
    global pacientes_atendidos
    if bandera == 1:
        print("El %s es llevado a la salida a las %.2f. " % (paciente, env.now))
        yield self.env.timeout(random.randint(POST_VACUNA, POST_VACUNA+2))
        print("El %s se va a las %.2f. " % (paciente, env.now))
        pacientes_atendidos = pacientes_atendidos+1
    else:
        print("El %s se va a las %.2f. " % (paciente, env.now))
```

Para la salida del paciente tenemos la validación de si se vacunó o simplemente fue expulsado por complicaciones. Imprimimos y guardamos los resultados

4. Procesos

Tenemos la llegada del paciente, que será la que aparta el recurso del médico para que pueda atender a un paciente a la vez. Así mismo llamamos a las demás actividades en el orden previsto

```
# Procesos
def llegada_paciente(env, nombre, vacunacion):
    yield env.process(vacunacion.llega_paciente(nombre))
    with vacunacion.num_medicos.request() as medico:
        yield medico
    yield env.process(vacunacion.vacunar(nombre))
    env.process(vacunacion.salida(nombre))

def ejecutar_simulacion(env, num_medicos, num_pacientes, intervalo):
    global pacientes
    global ESPERA_INICIAL
    vacunacion = Centro_Vacunacion(env, num_medicos, num_pacientes)

    for i in range(5):
        env.process(llegada_paciente(env, 'Paciente-%d' % (i+1), vacunacion))

    while True:
        yield env.timeout(random.randint(intervalo-3, intervalo+2))
        i+=1
        env.process(llegada_paciente(env, 'Paciente-%d' % (i+1), vacunacion))
        pacientes = i+1
```

El método de ejecutar simulación tomará los parámetros iniciales y los utilizará para el desarrollo de la simulación, así mismo se encargará de enviar los pacientes iniciales a que empiecen la cola y posteriormente cada tiempo variante enviará un paciente más. Estos valores serán guardados para su posterior interpretación

5. Simulación

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
# Simulación
print("Iniciando Vacunación")
random.seed(77)

env = simpy.Environment()
env.process(ejecutar_simulacion(env, NUM_MEDICOS, NUM_PACIENTES, INTERVALO_LLEGADA))
env.run(until=NUM_PACIENTES*14)
```

Cuando ya tengamos nuestras variables y métodos correctamente creados corremos la simulación, así mismo debemos poner la variable que hará que la simulación finalice.

RESULTADO(S) OBTENIDO(S): Obtención de conocimientos necesarios para la realización de una simulación que nos permita observar una actividad tan común hoy en día como es la vacunación.

CONCLUSIONES: Los datos de la simulación pueden variar drásticamente si no tenemos la precaución de agregar valores reales.

Nombre de los estudiantes: Alejandro Enríquez