

Laboratorio 7

- Derek Arreaga - 22537
- Paula Barillas - 22764
- Mónica Salvatierra - 22249

Link del repositorio: <https://github.com/FabianKel/ModSim-LAB7>

Parte 1

Teoría

1. Describa con sus propias palabras la diferencia fundamental entre una Entidad de DES y un Agente de MBA. ¿Por qué esta diferencia hace que cada paradigma sea adecuado para tipos de problemas distintos? Proporcione un ejemplo de un problema de modelado para cada uno.

La diferencia principal está en el tipo de comportamiento que representan. En un modelo de DES (Simulación de Eventos Discretos), una entidad es algo pasivo que pasa por diferentes etapas o procesos del sistema, como si siguiera un recorrido predefinido. Las entidades no deciden nada por sí mismas, simplemente reaccionan a los eventos del sistema. En cambio, en un MBA (Modelado Basado en Agentes), los agentes son activos y autónomos. Cada agente puede tomar decisiones, interactuar con otros y adaptarse a su entorno.

DES se usa más en sistemas operativos o logísticos, donde hay flujos de trabajo definidos (por ejemplo, una fila en un banco o una línea de producción). MBA se usa cuando los comportamientos individuales influyen en el resultado global, como en simulaciones de tráfico o en la propagación de enfermedades.

- ****Ejemplo DES****: modelar el flujo de pacientes en una clínica para analizar tiempos de espera y uso de recursos. - ****Ejemplo MBA****: modelar cómo las personas reaccionan ante una epidemia, considerando decisiones como aislarse o vacunarse.

2. Explique por qué las colas son una consecuencia inevitable en la mayoría de los sistemas modelados con DES. ¿Qué tipo de información valiosa podemos extraer al analizar la longitud de una cola y el tiempo de espera en ella a lo largo de una simulación?

Las colas aparecen porque los recursos disponibles son limitados y las llegadas o demandas son variables. Si en algún momento llegan más entidades de las que pueden ser atendidas,

se genera una acumulación en espera. Por eso, en la mayoría de los sistemas modelados con DES, las colas son una consecuencia natural.

Analizar la longitud de la cola y el tiempo de espera nos da información muy útil, por ejemplo:

- Identificar los puntos donde se forman cuellos de botella.
- Determinar si el sistema tiene suficientes recursos.
- Evaluar la calidad del servicio y el nivel de satisfacción de los clientes.

Por ejemplo, si en una cafetería la cola promedio supera los diez minutos, se podría analizar si conviene agregar otro empleado o modificar el proceso de atención. Así, los datos de la simulación ayudan a tomar decisiones basadas en evidencia.

3. En clase se mencionaron los recursos móviles, estáticos y portátiles. Describa un escenario (diferente al hospital) que involucre los tres tipos de recursos. Identifique qué elementos serían cada tipo de recurso y cómo interactuarían.

Un ejemplo podría ser un sistema de reparto de paquetes en una empresa de mensajería, como FedEx o Cargo Expreso. En este caso, se pueden identificar los tres tipos de recursos claramente:

- ****Recursos estáticos****: los centros de distribución o bodegas principales, donde se clasifican y almacenan los paquetes antes de ser enviados.

- **Recursos móviles**: los camiones o motocicletas que transportan los paquetes desde los centros de distribución hasta los puntos de entrega.
- **Recursos portátiles**: los escáneres, dispositivos móviles o etiquetas electrónicas que los repartidores usan para registrar cada entrega y confirmar la recepción.

Estos tres recursos interactúan constantemente. Los centros de distribución (estáticos) asignan rutas y organizan la salida de los vehículos (móviles), mientras que los repartidores usan los escáneres portátiles para actualizar el sistema en tiempo real. Así, se puede analizar la eficiencia del proceso, detectar retrasos, optimizar rutas y mejorar la gestión del inventario y la logística.

4. Imagine que está modelando la propagación de una enfermedad en una ciudad (un problema clásico de MBA). Ahora, quieres añadir el proceso de vacunación en centros de salud. ¿Cómo podría combinar MBA y DES? ¿Qué parte del modelo sería MBA y cuál DES? ¿Qué información pasaría de un modelo a otro?

La propagación de una enfermedad en una ciudad es un problema típico de MBA, porque cada persona (agente) puede tener un comportamiento distinto. Por ejemplo, algunos se aíslan, otros siguen saliendo, algunos se vacunan, etc. Cada decisión individual influye en la dinámica de contagio general. Si ahora se quiere agregar el proceso de vacunación en centros de salud, esa parte se modelaría mejor con DES, ya que implica flujos ordenados de personas que llegan, esperan su turno, son atendidas y luego se retiran.

En este modelo combinado:

- El MBA representaría la dinámica de la población: quién se contagia, quién decide vacunarse o moverse por la ciudad.

- El DES modelaría los procesos dentro de los centros de vacunación: tiempo de espera, capacidad de atención, personal disponible, etc.

La información pasaría de un modelo al otro. Por ejemplo, el MBA enviaría al DES el número de personas que decidieron vacunarse, y el DES devolvería los resultados de cuántas fueron atendidas efectivamente. Luego, esa información regresaría al MBA para actualizar la tasa de inmunidad y modificar la propagación de la enfermedad en la población.

Práctica

Librerías

```
In [ ]: import simpy
import random
import statistics
from typing import List
```

Parámetros globales

```
In [ ]: # Configuración
NUM_RECEPCIONISTAS = 2
NUM_MEDICOS = 2
TIEMPO_REGISTRO_PROMEDIO = 2.0
TIEMPO_CONSULTA_PROMEDIO = 7.0
TASA_LLEGADA_PACIENTES = 5.0
TIEMPO_SIMULACION = 120.0
```

```

# Contenedor global para los tiempos totales en clínica (por paciente)
tiempos_de_espera_totales: List[float] = []

#control de verbosidad
VERBOSE = True

# Semilla para reproducibilidad
RANDOM_SEED = 16
random.seed(RANDOM_SEED)

```

Definición del proceso paciente

```

In [3]: def paciente(env: simpy.Environment, nombre: str, recepcionistas: simpy.Resource, m
        tiempo_de_llegada = env.now
        if VERBOSE:
            print(f'{env.now:.2f}: {nombre} llega a la clínica')

        # Etapa de registro
        with recepcionistas.request() as req_registro:
            yield req_registro
            if VERBOSE:
                print(f'{env.now:.2f}: {nombre} inicia registro')
            tiempo_registro = random.expovariate(1.0 / TIEMPO_REGISTRO_PROMEDIO)
            yield env.timeout(tiempo_registro)
            if VERBOSE:
                print(f'{env.now:.2f}: {nombre} termina registro (duró {tiempo_registro}

        # Etapa de consulta
        with medicos.request() as req_consulta:
            yield req_consulta
            if VERBOSE:
                print(f'{env.now:.2f}: {nombre} inicia consulta')
            tiempo_consulta = random.expovariate(1.0 / TIEMPO_CONSULTA_PROMEDIO)
            yield env.timeout(tiempo_consulta)
            if VERBOSE:
                print(f'{env.now:.2f}: {nombre} termina consulta (duró {tiempo_consulta}

        # Fin del proceso del paciente, se calcula tiempo total en clínica
        tiempo_total = env.now - tiempo_de_llegada
        tiempos_de_espera_totales.append(tiempo_total)
        if VERBOSE:
            print(f'{env.now:.2f}: {nombre} sale de la clínica (tiempo total: {tiempo_t

```

Generador de pacientes

```

In [4]: def generador_pacientes(env: simpy.Environment, recepcionistas: simpy.Resource, med
        """Generador de llegadas: crea procesos 'paciente' con tiempos exponenciales en
        i = 0
        while True:
            intervalo = random.expovariate(1.0 / TASA_LLEGADA_PACIENTES)
            yield env.timeout(intervalo)
            i += 1
            nombre = f'Paciente_{i}'

```

```
env.process(paciente(env, nombre, recepcionistas, medicos))
if VERBOSE:
    print(f'{env.now:.2f}: Generado {nombre} (siguiente llegada en ~{interv
```

Ejecución de la simulación

```
In [5]: # Creación de entorno y recursos
env = simpy.Environment()
recepcionistas = simpy.Resource(env, capacity=NUM_RECEPCIONISTAS)
medicos = simpy.Resource(env, capacity=NUM_MEDICOS)

# generador de pacientes
env.process(generator_pacientes(env, recepcionistas, medicos))

# simulación
env.run(until=TIEMPO_SIMULACION)

print('Simulación finalizada')
```

5.10: Generado Paciente_1 (siguiente llegada en ~5.10 min)
5.10: Paciente_1 llega a la clínica
5.10: Paciente_1 inicia registro
5.23: Generado Paciente_2 (siguiente llegada en ~0.13 min)
5.23: Paciente_2 llega a la clínica
5.74: Paciente_1 termina registro (duró 0.64 min)
5.74: Paciente_1 inicia consulta
5.74: Paciente_2 inicia registro
6.49: Generado Paciente_3 (siguiente llegada en ~1.26 min)
6.49: Paciente_3 llega a la clínica
8.00: Paciente_2 termina registro (duró 2.26 min)
8.00: Paciente_2 inicia consulta
8.00: Paciente_3 inicia registro
8.64: Paciente_2 termina consulta (duró 0.64 min)
8.64: Paciente_2 sale de la clínica (tiempo total: 3.41 min)
9.10: Paciente_3 termina registro (duró 1.10 min)
9.10: Paciente_3 inicia consulta
9.31: Paciente_3 termina consulta (duró 0.21 min)
9.31: Paciente_3 sale de la clínica (tiempo total: 2.82 min)
15.08: Paciente_1 termina consulta (duró 9.34 min)
15.08: Paciente_1 sale de la clínica (tiempo total: 9.98 min)
17.63: Generado Paciente_4 (siguiente llegada en ~11.14 min)
17.63: Paciente_4 llega a la clínica
17.63: Paciente_4 inicia registro
18.86: Generado Paciente_5 (siguiente llegada en ~1.23 min)
18.86: Paciente_5 llega a la clínica
18.99: Generado Paciente_6 (siguiente llegada en ~0.13 min)
18.99: Paciente_6 llega a la clínica
19.03: Paciente_4 termina registro (duró 1.41 min)
19.03: Paciente_4 inicia consulta
19.03: Paciente_5 inicia registro
20.10: Generado Paciente_7 (siguiente llegada en ~1.11 min)
20.10: Paciente_7 llega a la clínica
20.61: Paciente_5 termina registro (duró 1.57 min)
20.61: Paciente_5 inicia consulta
20.61: Paciente_6 inicia registro
21.35: Generado Paciente_8 (siguiente llegada en ~1.25 min)
21.35: Paciente_8 llega a la clínica
21.38: Generado Paciente_9 (siguiente llegada en ~0.03 min)
21.38: Paciente_9 llega a la clínica
23.92: Paciente_6 termina registro (duró 3.32 min)
23.92: Paciente_7 inicia registro
26.32: Paciente_7 termina registro (duró 2.40 min)
26.32: Paciente_8 inicia registro
26.38: Paciente_4 termina consulta (duró 7.35 min)
26.38: Paciente_4 sale de la clínica (tiempo total: 8.75 min)
26.38: Paciente_6 inicia consulta
26.84: Paciente_5 termina consulta (duró 6.23 min)
26.84: Paciente_5 sale de la clínica (tiempo total: 7.98 min)
26.84: Paciente_7 inicia consulta
27.15: Paciente_8 termina registro (duró 0.83 min)
27.15: Paciente_9 inicia registro
27.56: Paciente_6 termina consulta (duró 1.18 min)
27.56: Paciente_6 sale de la clínica (tiempo total: 8.57 min)
27.56: Paciente_8 inicia consulta
27.97: Paciente_9 termina registro (duró 0.82 min)

28.24: Paciente_8 termina consulta (duró 0.68 min)
28.24: Paciente_8 sale de la clínica (tiempo total: 6.90 min)
28.24: Paciente_9 inicia consulta
28.96: Paciente_9 termina consulta (duró 0.71 min)
28.96: Paciente_9 sale de la clínica (tiempo total: 7.58 min)
29.58: Generado Paciente_10 (siguiente llegada en ~8.19 min)
29.58: Paciente_10 llega a la clínica
29.58: Paciente_10 inicia registro
31.43: Paciente_10 termina registro (duró 1.85 min)
31.43: Paciente_10 inicia consulta
38.98: Generado Paciente_11 (siguiente llegada en ~9.40 min)
38.98: Paciente_11 llega a la clínica
38.98: Paciente_11 inicia registro
40.51: Paciente_11 termina registro (duró 1.54 min)
42.95: Paciente_10 termina consulta (duró 11.52 min)
42.95: Paciente_10 sale de la clínica (tiempo total: 13.37 min)
42.95: Paciente_11 inicia consulta
45.52: Generado Paciente_12 (siguiente llegada en ~6.54 min)
45.52: Paciente_12 llega a la clínica
45.52: Paciente_12 inicia registro
47.13: Paciente_12 termina registro (duró 1.61 min)
47.90: Generado Paciente_13 (siguiente llegada en ~2.38 min)
47.90: Paciente_13 llega a la clínica
47.90: Paciente_13 inicia registro
48.90: Paciente_7 termina consulta (duró 22.06 min)
48.90: Paciente_7 sale de la clínica (tiempo total: 28.80 min)
48.90: Paciente_12 inicia consulta
49.83: Paciente_13 termina registro (duró 1.93 min)
56.74: Generado Paciente_14 (siguiente llegada en ~8.84 min)
56.74: Paciente_14 llega a la clínica
56.74: Paciente_14 inicia registro
59.18: Paciente_14 termina registro (duró 2.44 min)
61.05: Generado Paciente_15 (siguiente llegada en ~4.31 min)
61.05: Paciente_15 llega a la clínica
61.05: Paciente_15 inicia registro
61.28: Generado Paciente_16 (siguiente llegada en ~0.23 min)
61.28: Paciente_16 llega a la clínica
61.56: Paciente_15 termina registro (duró 0.52 min)
61.56: Paciente_16 inicia registro
61.73: Paciente_16 termina registro (duró 0.17 min)
62.75: Paciente_12 termina consulta (duró 13.85 min)
62.75: Paciente_12 sale de la clínica (tiempo total: 17.23 min)
62.75: Paciente_13 inicia consulta
62.99: Generado Paciente_17 (siguiente llegada en ~1.71 min)
62.99: Paciente_17 llega a la clínica
62.99: Paciente_17 inicia registro
63.52: Generado Paciente_18 (siguiente llegada en ~0.53 min)
63.52: Paciente_18 llega a la clínica
63.64: Paciente_17 termina registro (duró 0.65 min)
63.64: Paciente_18 inicia registro
64.55: Paciente_18 termina registro (duró 0.91 min)
64.60: Paciente_13 termina consulta (duró 1.85 min)
64.60: Paciente_13 sale de la clínica (tiempo total: 16.70 min)
64.60: Paciente_14 inicia consulta
67.84: Paciente_14 termina consulta (duró 3.24 min)
67.84: Paciente_14 sale de la clínica (tiempo total: 11.10 min)

67.84: Paciente_15 inicia consulta
68.26: Paciente_11 termina consulta (duró 25.31 min)
68.26: Paciente_11 sale de la clínica (tiempo total: 29.28 min)
68.26: Paciente_16 inicia consulta
68.57: Generado Paciente_19 (siguiente llegada en ~5.05 min)
68.57: Paciente_19 llega a la clínica
68.57: Paciente_19 inicia registro
69.48: Paciente_15 termina consulta (duró 1.65 min)
69.48: Paciente_15 sale de la clínica (tiempo total: 8.44 min)
69.48: Paciente_17 inicia consulta
70.43: Paciente_16 termina consulta (duró 2.17 min)
70.43: Paciente_16 sale de la clínica (tiempo total: 9.15 min)
70.43: Paciente_18 inicia consulta
70.66: Paciente_19 termina registro (duró 2.09 min)
71.75: Paciente_18 termina consulta (duró 1.31 min)
71.75: Paciente_18 sale de la clínica (tiempo total: 8.23 min)
71.75: Paciente_19 inicia consulta
76.06: Paciente_17 termina consulta (duró 6.58 min)
76.06: Paciente_17 sale de la clínica (tiempo total: 13.07 min)
80.89: Paciente_19 termina consulta (duró 9.14 min)
80.89: Paciente_19 sale de la clínica (tiempo total: 12.32 min)
82.37: Generado Paciente_20 (siguiente llegada en ~13.80 min)
82.37: Paciente_20 llega a la clínica
82.37: Paciente_20 inicia registro
83.26: Generado Paciente_21 (siguiente llegada en ~0.89 min)
83.26: Paciente_21 llega a la clínica
83.32: Paciente_20 termina registro (duró 0.95 min)
83.32: Paciente_20 inicia consulta
83.32: Paciente_21 inicia registro
84.95: Paciente_21 termina registro (duró 1.63 min)
84.95: Paciente_21 inicia consulta
90.47: Paciente_20 termina consulta (duró 7.15 min)
90.47: Paciente_20 sale de la clínica (tiempo total: 8.11 min)
93.03: Paciente_21 termina consulta (duró 8.08 min)
93.03: Paciente_21 sale de la clínica (tiempo total: 9.77 min)
106.05: Generado Paciente_22 (siguiente llegada en ~22.79 min)
106.05: Paciente_22 llega a la clínica
106.05: Paciente_22 inicia registro
109.04: Paciente_22 termina registro (duró 2.99 min)
109.04: Paciente_22 inicia consulta
110.86: Paciente_22 termina consulta (duró 1.82 min)
110.86: Paciente_22 sale de la clínica (tiempo total: 4.81 min)
115.30: Generado Paciente_23 (siguiente llegada en ~9.25 min)
115.30: Paciente_23 llega a la clínica
115.30: Paciente_23 inicia registro
115.47: Generado Paciente_24 (siguiente llegada en ~0.16 min)
115.47: Paciente_24 llega a la clínica
116.06: Paciente_23 termina registro (duró 0.76 min)
116.06: Paciente_23 inicia consulta
116.06: Paciente_24 inicia registro
117.03: Generado Paciente_25 (siguiente llegada en ~1.56 min)
117.03: Paciente_25 llega a la clínica
117.72: Paciente_23 termina consulta (duró 1.66 min)
117.72: Paciente_23 sale de la clínica (tiempo total: 2.42 min)
Simulación finalizada

Métricas Simulación

```
In [6]: # Métricas básicas una vez finalizada la simulación
num_pacientes = len(tiempos_de_espera_totales)
if num_pacientes > 0:
    tiempo_promedio = statistics.mean(tiempos_de_espera_totales)
    tiempo_maximo = max(tiempos_de_espera_totales)
else:
    tiempo_promedio = 0.0
    tiempo_maximo = 0.0

print(f'Número total de pacientes procesados: {num_pacientes}')
print(f'Tiempo promedio en la clínica: {tiempo_promedio:.2f} minutos')
print(f'Tiempo máximo en la clínica: {tiempo_maximo:.2f} minutos')
print('Primeros 10 tiempos registrados:', tiempos_de_espera_totales[:10])
```

Número total de pacientes procesados: 23
Tiempo promedio en la clínica: 10.82 minutos
Tiempo máximo en la clínica: 29.28 minutos
Primeros 10 tiempos registrados: [3.4116167827793857, 2.8198625814273477, 9.978396837806514, 8.754275064290312, 7.9775670285303235, 8.56912344270054, 6.89686156142772, 7.576292513625809, 13.37140850974884, 28.79529520337851]

Visualizaciones

```
In [7]: import sys
import subprocess

plotly_available = False
try:
    import plotly.express as px
    plotly_available = True
except Exception:
    try:
        print('Plotly no encontrado, intentando instalar plotly...')
        subprocess.check_call([sys.executable, '-m', 'pip', 'install', 'plotly'])
        import plotly.express as px
        plotly_available = True
    except Exception as e_inst:
        print('No se pudo instalar plotly automáticamente. Usando Matplotlib como r
        plotly_available = False

if len(tiempos_de_espera_totales) == 0:
    print('No hay datos de pacientes. Ejecuta la simulación antes de generar las vi
else:
    # Preparar DataFrame opcionalmente
    try:
        import pandas as pd
        df = pd.DataFrame({'tiempo_total': tiempos_de_espera_totales})
    except Exception:
        df = None
    if plotly_available:
        # Histograma
        fig = px.histogram(df if df is not None else {'tiempo_total': tiempos_de_es
```

```

fig.show()
# Boxplot
fig2 = px.box(df if df is not None else {'tiempo_total': tiempos_de_espera_
fig2.show()
# Resumen
if df is not None:
    print(df['tiempo_total'].describe())
else:
    import matplotlib.pyplot as plt
    plt.figure(figsize=(8,4))
    plt.hist(tiempos_de_espera_totales, bins=30, color='C0', edgecolor='black')
    plt.title('Histograma: Tiempo total en clínica')
    plt.xlabel('Minutos')
    plt.ylabel('Frecuencia')
    plt.show()
    plt.figure(figsize=(4,6))
    plt.boxplot(tiempos_de_espera_totales, vert=True)
    plt.title('Boxplot: Tiempo total en clínica')
    plt.ylabel('Minutos')
    plt.show()

    import statistics
    try:
        mean = statistics.mean(tiempos_de_espera_totales)
        stdev = statistics.pstdev(tiempos_de_espera_totales) if len(tiempos_de_
        print(f'count={len(tiempos_de_espera_totales)}, mean={mean:.2f}, pop_st
    except Exception as e_stats:
        print('Error calculando estadísticas:', e_stats)

```

```

count    23.000000
mean     10.816186
std       6.897958
min       2.416767
25%       7.776930
50%       8.754275
75%      12.695374
max      29.282146
Name: tiempo_total, dtype: float64

```

Preguntas de Análisis

1. Configure la simulación con los parámetros iniciales (NUM_RECEPCIONISTAS = 1, NUM_MEDICOS = 2) y proceda con su ejecución. Observe detenidamente la traza de eventos que su script imprime en la consola. ¿En qué etapa del proceso (registro o consulta) parece que los pacientes invierten la mayor parte de su tiempo de espera? ¿Cuál es el tiempo promedio total que un paciente permanece en la clínica según este escenario base?

En este caso, los pacientes pasan la mayor parte de su tiempo de espera en la etapa de consulta. En donde se observa que muchas consultas duran mucho más que los registros y generan cola en la segunda etapa por ejemplo, Paciente_1 inicia consulta a 5.74 y la termina a 15.08 en donde su consulta es de 9.34 min, Ahora bien, el Paciente_7 tuvo una consulta

muy larga que contribuyó a un tiempo total de 28.80 min, y también se observan consultas de 22.06 min y 25.31 min en la traza. En contraste, los tiempos de registro en la traza son cortos y típicamente entre 0.17 y 3.3 minutos como el Paciente_1 registro 0.64 min, Paciente_3 registro 1.10 min. En donde esa diferencia se refleja en las estadísticas obtenidas y con llegada media cada 5 min la tasa es 0.2 pacientes/min; Donde la utilización aproximada del recepcionista es $\rho_{\text{recep}} \approx 0.22/1 = 0.4$ (40%), mientras que la de los médicos es $\rho_{\text{med}} \approx 0.27/2 = 0.7$ (70%) es decir, los médicos están mucho más ocupados, por eso aparece espera en consulta.

En tiempo promedio se obtuvo que el tiempo en la clínica = 10.82 minutos, tiempo máximo = 29.28 minutos, y primer resumen estadístico (count=23, mean≈10.82, std≈6.90, mediana≈8.75, 75%≈12.70). En donde estos números muestran que el tiempo efectivo de servicio (registro ~2 min + consulta ~7 min ≈ 9 min) se ve aumentado por esperas en consulta, provocando una media algo por encima del servicio esperado y una cola con cola larga en algunos casos (cola/tiempos máximos elevan la media).

2. A continuación, usted deberá investigar cuál es el factor limitante principal del sistema. Mantenga el número de recepcionistas en NUM_RECEPCIONISTAS = 1 y realice un cambio significativo en el número de médicos, aumentándolo a NUM_MEDICOS = 5.

a. Tras ejecutar la simulación, ¿observa usted una disminución sustancial en el tiempo promedio que los pacientes pasan en la clínica?

Sí, en este caso aumentar los médicos a 5 produjo una disminución apreciable del tiempo promedio por paciente. Ya que el reporte final muestra que el número de pacientes = 26 y el tiempo promedio en la clínica = 9.02 minutos teniendo una desviación estándar ≈ 6.34, mediana ≈ 8.86. Comparado con la línea base (tiempo promedio ≈ 10.82 min), la media cayó 1.80 minutos, es decir una reducción relativa de ≈16.6%. En donde este efecto se aprecia en la traza, ya que por ejemplo hay más casos en los que las consultas comienzan y terminan pronto donde hay varios pacientes con tiempos totales por debajo de 10 min y la cola por consulta aparece menos prolongada que en la corrida base. La desviación estándar disminuyó ligeramente (de ≈6.90 a ≈6.34), lo que indica también una reducción moderada en la cola larga de la distribución.

b. Con base en este resultado, ¿cuál de los dos recursos (recepcionistas o médicos) concluye usted que es el cuello de botella del sistema? Por favor, explique su razonamiento

Con base en este resultado, el cuello de botella inicial eran los médicos. Ya que con la tasa de llegadas media de 1 paciente cada 5 min ($\lambda = 0.2$ pacientes/min), la utilización teórica de los médicos en la línea base era alta -> $\rho_{\text{med}} \approx (\lambda * \text{tiempo_consulta}) / \text{NUM_MEDICOS} = (0.2 * 7) / 2 \approx 0.70$ (70%), mientras que la utilización del recepcionista era $\rho_{\text{recep}} \approx (0.2 * 2) / 1 = 0.40$ (40%). Entonces al aumentar los médicos a 5 la utilización médica cae a ≈28%

$((0.2 \cdot 7)/5 \approx 0.28)$ en donde esto reduce la congestión en la etapa de consulta; por eso el tiempo promedio total baja. Entonces básicamente al reducir la carga / utilización de los médicos se logra una mejora notable en la métrica principal, por lo que se puede considerar que los médicos eran el recurso limitante en la configuración inicial.

3. Restablezca el número de médicos a su valor original (NUM_MEDICOS = 2). Ahora, dirija su atención al recurso que identificó como el cuello de botella en la pregunta anterior. Si fue el recepcionista, modifique el parámetro a NUM_RECEPCIONISTAS = 2 y ejecute la simulación.

a. Compare el tiempo promedio resultante en la clínica de este nuevo escenario con el obtenido en la línea base (pregunta 1).

En este caso, al restaurar NUM_MEDICOS = 2 y aumentar NUM_RECEPCIONISTAS a 2 se obtuvo un tiempo promedio en la clínica de 11.11 minutos ($n = 22$, mediana ≈ 9.53 , std ≈ 6.49). En donde si se compara con la línea base (tiempo promedio ≈ 10.82 min) este escenario no mejora el tiempo medio, ya que se observa que aumenta ligeramente $\approx +0.29$ min. Esto puede ser debido a que los registros suelen ser muy rápidos (ya que varios registros representarían < 1 min) y casi no se forman colas en recepción, pero también se siguen apareciendo consultas muy largas un ejemplo son las consultas que se observan que son de 22.06 min, 25.31 min que provocan esperas y tiempos totales elevados para algunos pacientes. Por lo cual al doblar recepcionistas reduce la congestión en la etapa de registro, pero no reduce la parte dominante de la espera, que sigue ocurriendo en consulta por eso el tiempo promedio total no baja.

b. ¿Qué intervención resultó más efectiva para mejorar el rendimiento global del sistema: el incremento de médicos (pregunta 2) o el de recepcionistas?

En este caso la intervención más efectiva fue aumentar el número de médicos. Ya que cuando se aumentaron los médicos a 5 el tiempo promedio cayó a 9.02 min, obteniendo una reducción $\approx 16.6\%$ respecto a la base, mientras que aumentar recepcionistas no mostró mejora y en la ejecución se observó un promedio de 11.11 min. En donde la utilización de recursos con $\lambda = 0.2$ pacientes/min y tiempos medios registro ≈ 2 min y consulta ≈ 7 min, la utilización médica en la línea base era alta: $\rho_{\text{med}} \approx (0.2 \cdot 7)/2 \approx 0.70$ (70%), mientras que la del recepcionista era $\rho_{\text{recep}} \approx (0.2 \cdot 2)/1 = 0.40$ (40%). Por lo cual al añadir recepcionistas la utilización de recepción cae mucho ($\approx 20\%$) pero la de los médicos permanece alta ($\approx 70\%$), de modo que la congestión se mantiene en consulta, ya que al aumentar médicos la utilización médica baja (por ejemplo a $\approx 28\%$ con 5 médicos) y por eso la cola de consulta y el tiempo total se reducen.

4. Asuma el rol de un consultor de procesos contratado para optimizar la operación de esta clínica. Basándose en la evidencia cuantitativa recopilada de sus tres experimentos (línea base, incremento de médicos, incremento de recepcionistas), redacte una recomendación profesional y concisa para la dirección de la clínica. ¿En qué recurso

debería invertir la clínica para lograr la reducción más significativa en los tiempos de espera, considerando el menor costo posible (bajo el supuesto de que el costo de un recepcionista es inferior al de un médico)?

Con base en los tres experimentos realizados línea base (tiempo promedio ≈ 10.82 min), el aumento de médicos a 5 (tiempo promedio ≈ 9.02 min, reducción $\approx 16.6\%$) y aumento de recepcionistas a 2 (tiempo promedio ≈ 11.11 min, sin mejora) nuestra recomendación es priorizar la inversión en capacidad médica antes que en recepción. Ya que los resultados cuantitativos y la explicación por utilización lo soportan: en la configuración inicial la utilización de los médicos era alta ($\approx 70\%$), mientras que la del recepcionista era moderada ($\approx 40\%$), por lo que la congestión principal ocurre en la etapa de consulta. Ya que por ejemplo, al aumentar médicos la media cayó cerca de 1.8 minutos por paciente y se redujo la cola larga, en cambio al doblar recepcionistas apenas disminuyó hasta consideramos que incluso aumentó por variabilidad en el tiempo total porque los registros ya no eran el cuello de botella. Por lo cuál si la clínica puede pagar el coste de al menos un médico adicional, eso dará el mayor beneficio en reducción de espera por paciente. Ahora bien si el presupuesto es muy limitado y un recepcionista cuesta, considerando una fracción de lo que cuesta un médico, una alternativa de menor coste es contratar médicos a tiempo parcial o por horas punta por ejemplo tener +1–2 médicos sólo en la franja horaria con mayor demanda o bien implementar un “fast-track” (consultas breves para casos rápidos) que aumente la capacidad efectiva de consulta sin contratar médicos fijos.

Prompt Utilizado

Uso de ChatGPT

Hola, estoy trabajando en mi laboratorio de modelación y simulación en donde necesito construir una simulación de eventos discretos para una clínica simple usando `simpy`. Por favor genera un esqueleto básico en Python que implemente lo siguiente, de manera que pueda realizar todos los ajustes necesarios a la implementación deseada:

Se tiene que definir parámetros globales en la parte superior: `NUM_RECEPCIONISTAS` (1), `NUM_MEDICOS` (2), `TIEMPO_REGISTRO_PROMEDIO` (2.0), `TIEMPO_CONSULTA_PROMEDIO` (7.0), `TASA_LLEGADA_PACIENTES` (5.0), `TIEMPO_SIMULACION` (120.0) y una lista vacía `tiempos_de_espera_totales`.

- Implementar la función generadora `paciente(env, nombre, recepcionistas, medicos)` que: captura `tiempo_de_llegada = env.now`; en donde se solicita un recepcionista (`with recepcionistas.request()` as `req`: / `yield req`), simula el tiempo de registro con `random.expovariate(1.0 / TIEMPO_REGISTRO_PROMEDIO)` y hace `yield env.timeout(...)`; luego solicita un médico (`with medicos.request()` as `req`: / `yield req`), simula la consulta con `random.expovariate(1.0 / TIEMPO_CONSULTA_PROMEDIO)` y hace `yield env.timeout(...)`; calcula `tiempo_total = env.now - tiempo_de_llegada` y lo

añade a `tiempos_de_espera_totales` . Incluye `print()` informativos en llegada, inicio/fin de registro, inicio/fin de consulta y salida para trazas. En el bloque principal: crear `env = simpy.Environment()` , instanciar `recepcionistas = simpy.Resource(env, capacity=NUM_RECEPCIONISTAS)` y `medicos = simpy.Resource(env, capacity=NUM_MEDICOS)` , arrancar el generador y ejecutar `env.run(until=TIEMPO_SIMULACION)` . Para que al terminar, se calcule y muestre el número total de pacientes procesados, tiempo promedio en la clínica y tiempo máximo registrado (usar `statistics.mean`).

Por qué funcionó este prompt

Este prompt funciona porque es específico y prescriptivo: lista las bibliotecas exactas, las variables de configuración, la estructura del proceso (llegada → registro → consulta → salida), el tipo de distribuciones estocásticas a usar (expovariate para tiempos), y las métricas que deben recopilarse para el análisis. En donde el prompt permite generar código que es reproducible, fácil de depurar y listo para experimentar (como cambiar parámetros y comparar escenarios).

Referencias

1. Discrete-Event Modeling. (n.d.). <https://www.anylogic.com/use-of-simulation/discrete-event-simulation/>
2. What is discrete event simulation and how does it work. (n.d.). Haskoning. <https://www.haskoning.com/en/twinn/blogs/2024/what-is-discrete-event-simulation-and-how-does-it-work>