

UTN: FACULTAD REGIONAL ROSARIO

DEPARTAMENTO DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Cátedra: Simulación

Trabajo Práctico MM1

Autores:

Andrés J. Botello	Legajo: 43697
Santiago Ciriaci	Legajo:43828
Joaquín Romero	Legajo:43740

Supervisado por:
Ing. Torres, Juan Ignacio

Trabajo Práctico de Simulación

Andrés Julián Botello, Joaquín Romero, Santiago Ciriaci

Andrés Botello: 43697, Joaquín Romero: 43740, Santiago Ciriaci: 43828

Abstracto

Las líneas de espera generan malestar, ineficiencia, retraso y otros problemas, lo que origina un coste de tiempo y un coste económico. Es muy importante evaluar el balance entre el aumento del nivel de servicio y el tamaño de las colas de espera. Por tanto, es necesario entender la relación entre el número de servidores en un sistema (o eficacia de los mismos) y la cantidad de tiempo gastado en la cola (o cantidad de clientes en la misma).

En sistemas de colas sencillos dichas relaciones se pueden encontrar analíticamente. En sistemas más complejos se pueden analizar mediante simulación.

Palabras Clave:

Simulación, Colas, M/M/1.

Índice

1. Introducción	3
1.1. Aplicación de los sistemas de espera	3
2. Desarrollo	4
3. Conclusiones	5
4. Anexo	6
5. Bibliografía	8

1. Introducción

En Teoría de Colas, disciplina dentro de la teoría matemática de la probabilidad, una cola $M/M/1$ representa la longitud de la cola en un sistema que posee un servidor, donde los arribos son determinados por una distribución de Poisson y los tiempos de servicio por una exponencial. Este tipo de modelo es el más elemental de los modelos de colas y su nombre está escrito haciendo uso de la notación de Kendall.

Como se ha mencionado anteriormente, hay sistemas de colas sencillos a los cuales se los puede resolver analíticamente encontrando una solución determinística, como lo es el $M/M/1$.

Habiendo presentado esta idea, en este informe procederemos a programar un sistema de colas de estas características mediante el uso de la técnica de simulación de eventos discretos.

El hecho de poseer solución analítica, nos permitirá realizar la comparación de ambos resultados con el fin de contrastar sus diferencias y similitudes.

En la mayoría de los problemas de líneas de espera que se presentan en la vida real, la tasa de demanda varía; es decir, los clientes llegan a intervalos imprevisibles. Lo más común es que también haya variaciones en el ritmo de producción del servicio, dependiendo de las necesidades del cliente.

Los pacientes que aguardan al médico en su consultorio y los taladros descompuestos que esperan en una instalación de reparación tienen mucho en común desde una perspectiva de Producción y Operaciones. Ambas utilizan recursos humanos y recursos de equipos para mantener los valiosos activos de producción (gente y máquinas) en buenas condiciones.

Los administradores de operaciones reconocen el trueque que se lleva a cabo entre el costo de ofrecer un buen servicio y el costo del tiempo de espera del cliente o la máquina. Los administradores desean que las filas de espera sean lo suficientemente cortas, de tal forma que los clientes no se sientan descontentos y se vayan sin comprar, o que compren pero nunca regresen. Sin embargo, los administradores están dispuestos a permitir alguna espera, si ésta es proporcional a un ahorro significativo en los costos del servicio. Cuando la empresa intenta elevar su nivel de servicio, se observa un incremento en los costos.

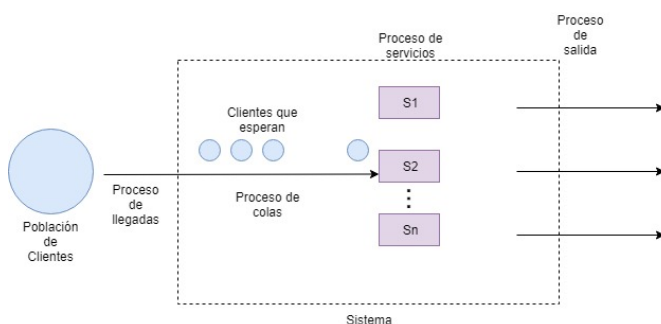


Figura 1: Componentes de Sistemas de espera

1.1. Aplicación de los sistemas de espera

La teoría de los sistemas de espera es aplicable a empresas de servicios o manufactureras, porque relaciona la llegada de los clientes y las características de procesamiento del sistema de servicios con las características de salida de dicho sistema. El sistema de servicio puede consistir en la operación de cortar el cabello en una peluquería, o bien, en el departamento de partes, con una máquina determinada para atender un pedido de producción. Otros ejemplos de clientes y servicios son las filas de los espectadores que esperan frente a un estadio de fútbol para comprar entradas, los camiones que aguardan para ser descargados en una planta de acopio de cereales, las máquinas en espera de ser reparadas por una cuadrilla de mantenimiento y los pacientes que hacen antesala para ser atendidos por un médico. Cualquiera que sea la situación, los problemas referentes a líneas de espera tienen algunos elementos en común.



Figura 2: Cola en aeropuerto

2. Desarrollo

A continuación, mostraremos el resultado obtenido a través del uso de una calculadora $M/M/1$ que está basada en el método analítico. Se podrá acceder a esta desde la sección “Bibliografía” en la que se adjuntará su Link.
Para los siguientes resultados, se han usado los parámetros:

- $te = 7,0$
- $ts = 9,0$

Siendo te el tiempo entre arribos y ts tiempo de servicio.

Resultados:

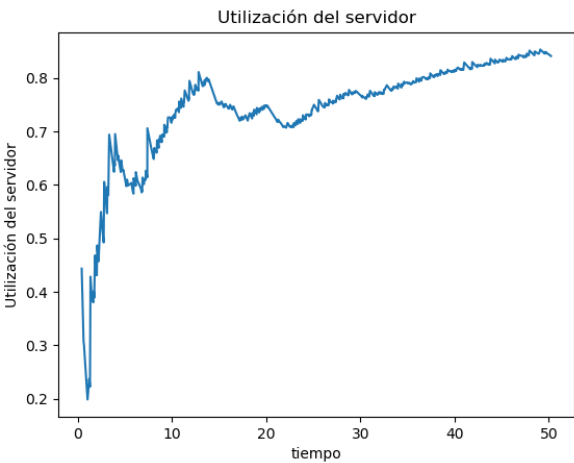
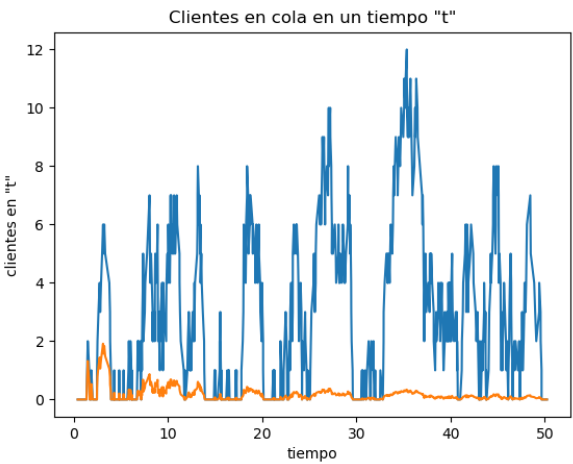
Estadístico	Valor
Número promedio de clientes en cola	2,722
Utilización promedio del servidor	0,7778
Demora promedio de cliente	0,3889

Tabla 1: Tabla de resultados (método analítico).

Ahora bien, utilizando estos mismos parámetros hemos corrido el programa de simulación varias veces y se han logrado las siguientes salidas.

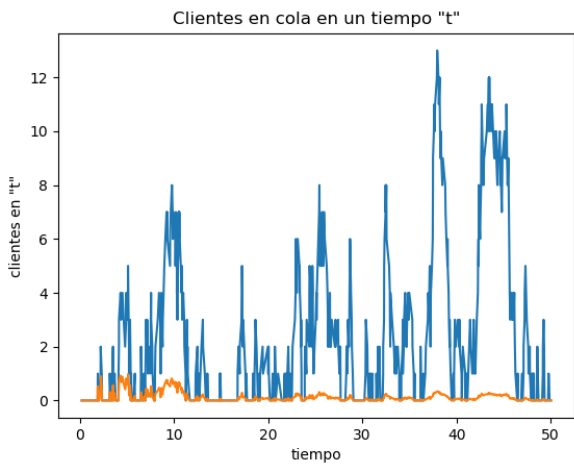
Resultados de la primera corrida:

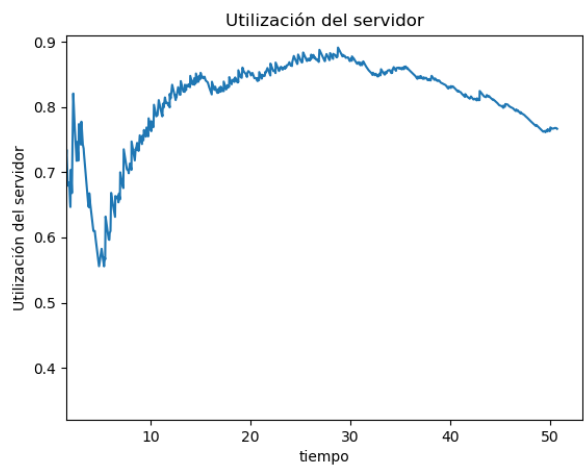
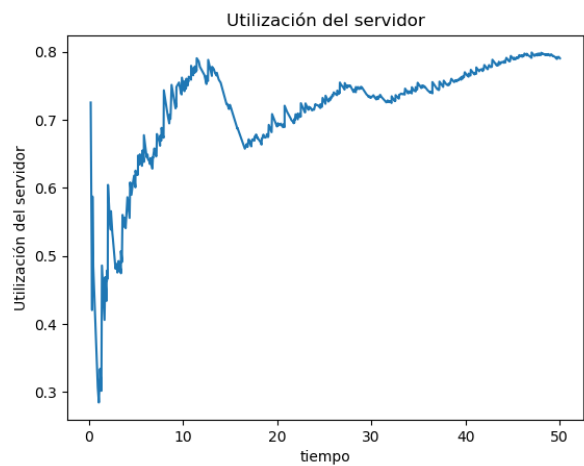
Estadístico	Valor
Número promedio de clientes en cola	3,0133
Utilización promedio del servidor	0,8413
Demora promedio de cliente	0,4147



Resultados de la segunda corrida:

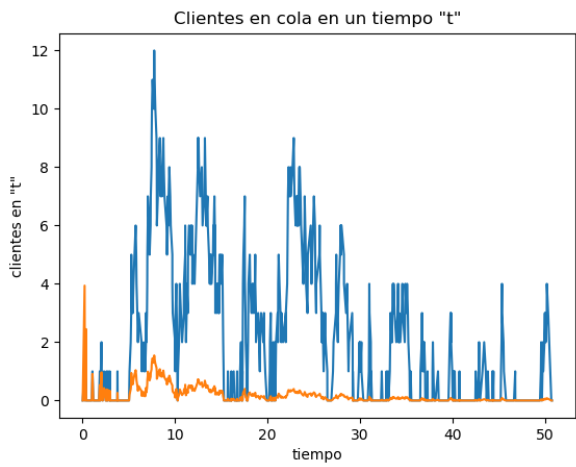
Estadístico	Valor
Número promedio de clientes en cola	2,3627
Utilización promedio del servidor	0,7904
Demora promedio de cliente	0,3204





Resultados de la tercera corrida:

Estadístico	Valor
Número promedio de clientes en cola	2,1463
Utilización promedio del servidor	0,7664
Demora promedio de cliente	0,3212



3. Conclusiones

Podemos observar que en las tres corridas del programa que hemos presentado anteriormente, los valores obtenidos se aproximan notablemente a los resultados analíticos proporcionados por la calculadora de $M/M/1$. Especialmente en la segunda y tercera corrida donde la variación no es tan notable.

En conclusión, habiendo obtenido valores tan cercanos a los exactos podemos afirmar que la técnica de simulación es una herramienta muy valiosa que nos facilitará obtención de “buenas” soluciones para problemas muy complejos, los cuales, no pueden ser resueltos analíticamente.

4. Anexo

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 class sim():
5     reloj, ts_acumulado, demora_acumulada,
6     area_q_t, tiempo_ultimo_evento =
7     0.0,0.0,0.0,0.0,0.0
8     nro_clientes_cola, completaron_demora,
9     paso = 0,0,0
10    tm_entre_arribos = 7.0
11    tm_servicio = 9.0
12    iniciado = False
13    estado_servidor = "" #D - disponible | O -
14    ocupado
15    proximo_evento = "" #A - Arribo | P -
16    partida
17    lista_eventos = []
18    cola = []
19    #las listas siguientes se utilizan s lo
20    con el fin de generar las gr ficas
21    cliColaEnT = []
22    relojEnT = []
23    tsAcuEnT = []
24
25 def inicializar():
26     sim.estado_servidor = "D"
27     sim.proximo_evento = ""
28     sim.reloj, sim.ts_acumulado, sim.
29     demora_acumulada, sim.area_q_t, sim.
30     tiempo_ultimo_evento =
31     0.0,0.0,0.0,0.0,0.0
32     sim.nro_clientes_cola, sim.
33     completaron_demora, sim.paso = 0,0,0
34
35     #Tiempo del primer arribo
36     sim.lista_eventos.append(np.random.
37     exponential(1/sim.tm_entre_arribos))
38
39     #Numero grande para asegurar que el primer
40     evento sea un arribo
41     sim.lista_eventos.append(99999999)
42     sim.iniciado = False
43
44 def run():
45     print("Iniciando simulacion")
46     inicializar()
47     while sim.reloj < 50 or sim.
48         nro_clientes_cola != 0 or sim.
49         estado_servidor != "D":
50         print("reloj: "+ str(sim.reloj))
51         tiempos()
52
53         if sim.proximo_evento == "A":
54             arribo()
55         else:
56             partida()
57
58     reportes()

```

```

46 def tiempos():
47     sim.tiempo_ultimo_evento = sim.reloj
48     if sim.lista_eventos[0] <= sim.
49         lista_eventos[1]:
50         sim.reloj = sim.lista_eventos[0]
51         sim.proximo_evento = "A"
52     else:
53         sim.reloj = sim.lista_eventos[1]
54         sim.proximo_evento = "P"
55
56 def arribo():
57     sim.lista_eventos[0] = sim.reloj + np.
58     random.exponential(1/sim.
59     tm_entre_arribos)
60     if sim.estado_servidor == "D":
61         sim.estado_servidor = "O"
62         sim.lista_eventos[1] = sim.reloj + np.
63         random.exponential(1/sim.
64         tm_servicio)
65         sim.ts_acumulado += (sim.lista_eventos
66         [1] - sim.reloj)
67         sim.completaron_demora += 1
68         #grafica
69         sim.relojEnT.append(sim.reloj)
70         sim.cliColaEnT.append(sim.
71         nro_clientes_cola)
72         sim.tsAcuEnT.append(sim.ts_acumulado)
73     else:
74         sim.area_q_t += sim.nro_clientes_cola
75         * (sim.reloj - sim.
76         tiempo_ultimo_evento)
77         sim.nro_clientes_cola += 1
78         sim.cola.append(sim.reloj)
79         #grafica
80         sim.cliColaEnT.append(sim.
81         nro_clientes_cola)
82         sim.relojEnT.append(sim.reloj)
83         sim.tsAcuEnT.append(sim.ts_acumulado)
84
85 def partida():
86     if sim.nro_clientes_cola > 0:
87         sim.lista_eventos[1] = sim.reloj + np.
88         random.exponential(1/sim.
89         tm_servicio)
90         sim.demora_acumulada += sim.reloj -
91         sim.cola[0]
92         sim.completaron_demora += 1
93         sim.ts_acumulado += sim.lista_eventos
94         [1] - sim.reloj
95         sim.area_q_t += (sim.nro_clientes_cola
96         * (sim.reloj - sim.
97         tiempo_ultimo_evento))
98         sim.nro_clientes_cola -= 1
99         sim.cola.pop(0)
100        #grafica
101        sim.cliColaEnT.append(sim.
102        nro_clientes_cola)
103        sim.relojEnT.append(sim.reloj)
104        sim.tsAcuEnT.append(sim.ts_acumulado)
105    else:
106        sim.estado_servidor = "D"

```

```

91     sim.lista_eventos[1] = 99999999          144
92     #grafica
93     sim.relojEnT.append(sim.reloj)           145
94     sim.cliColaEnT.append(sim.
95         nro_clientesCola)
96     sim.tsAcuEnT.append(sim.ts_acumulado)    146
97
98     def nro_prom_clientesCola():
99         if sim.reloj != 0:                    147
100             nro_clientes_promCola = sim.area_q_t/
101                 sim.reloj
102             return nro_clientes_promCola      148
103         else:
104             nro_clientes_promCola = 0         149
105             return nro_clientes_promCola      150
106
107     def utilizacion_prom_servidor():
108         if sim.reloj != 0:                    151
109             utilizacion_prom_servidor = sim.
110                 ts_acumulado/sim.reloj
111             return utilizacion_prom_servidor  152
112         else:
113             utilizacion_prom_servidor = 0     153
114             return utilizacion_prom_servidor  154
115
116     def demora_prom_cliente():
117         if sim.completaron_demora != 0:
118             demora_prom_cliente = sim.
119                 demora_acumulada/sim.
120                 completaron_demora
121             return demora_prom_cliente
122         else:
123             demora_prom_cliente = 0
124             return demora_prom_cliente
125
126     def graficaPromCliEnCola():
127         media = []
128         for i in range(len(sim.cliColaEnT)):
129             media.append(sim.cliColaEnT[i]/sim.
130                 relojEnT[i])
131         plt.title('Clientes en cola en un tiempo "
132             t"')
133         plt.plot(sim.relojEnT, sim.cliColaEnT)
134         plt.plot(sim.relojEnT, media)
135         plt.xlabel('tiempo')
136         plt.ylabel('clientes en "t"')
137         plt.show()
138
139     def graficaUtiServidor():
140         uti = []
141         for i in range(len(sim.relojEnT)):
142             uti.append(sim.tsAcuEnT[i]/sim.
143                 relojEnT[i])
144         plt.title('Utilizaci n del servidor')
145         plt.plot(sim.relojEnT, uti)
146         plt.ylabel('Utilizaci n del servidor')
147         plt.xlabel('tiempo')
148         plt.show()
149
150     def reportes():
151
152         print(" Estadsticos ".center(101,' '),'\n'
153             n')
154         print(("Cantidad promedio de clientes en
155             cola: " + str(nro_prom_clientesCola())
156                 ).center(100,' '))
157         print(("Utilizacion promedio del servidor:
158             " + str(utilizacion_prom_servidor())
159                 ).center(100,' '))
160         print(("Demora promedio de cliente: " + str
161             (demora_prom_cliente())
162                 ).center(100,' '))
163
164         #graficas
165         graficaPromCliEnCola()
166         graficaUtiServidor()
167
168     if __name__ == '__main__':
169         run()

```


5. Bibliografía

- *Calculadora M/M/1*: https://www.supositorio.com/rcalc/rcalc-lite_esp.htm
- “*Modelos de líneas de espera*” Universidad Nacional de Mar del Plata: http://nulan.mdp.edu.ar/1622/1/17_modelos_lineas_espera.pdf
- *Simulation Modeling and Analysis* by Law A. Kelton W