TP 3.0 - SIMULACIÓN DE UN MODELO DE COLA M/M/1

Antonelli, Nicolás
Departamento de Ingeniería en Sistemas
Universidad Tecnológica Nacional - FR Rosario
Rosario, Zeballos 1341
niconelli2@gmail.com

Acciarri, Joshua
Departamento de Ingeniería en Sistemas
Universidad Tecnológica Nacional - FR Rosario
Rosario, Zeballos 1341
acciarrijoshua@gmail.com

Recalde, Alejandro
Departamento de Ingeniería en Sistemas
Universidad Tecnológica Nacional - FR Rosario
Rosario, Zeballos 1341
alejandrorecalde5@gmail.com

11 de julio de 2020

ABSTRACT

Simulación vs Modelo analítico: Se desarrolla un modelo de cola única con un solo servidor desde dos enfoques diferentes; uno calculado a partir de parámetros matemáticos-estadísticos y otro como resultado de una simulación con los mismos parámetros. Análisis de ventajas y desventajas, inconvenientes y beneficios de usar un modelo puramente matemático o bien dejarlo correr en un programa de computadora.

***K*eywords** Teoría · Cola · Modelo · Servidor · Analítico · Simulación · Artículo Científico

1. Introducción

En pocas palabras, una *cola* es una colección ordenada de *ítems* donde la adición de nuevos ítems tiene lugar en uno de los extremos. La cola puede estar compuesta teóricamente de cualquier cosa según sea el *modelo* que necesitemos: personas, procesos, solicitudes, vehículos, etc.

Prácticamente convivimos con diferentes modelos de colas todo el tiempo, aún desde nuestras casas con la computadora, el sistema operativo implementa un sistema con colas para asignar recursos a todos los programas que utilizaremos.

Es por eso que es importante estudiar qué origina una cola, y además saber cómo modelar correctamente la situación para poder analizar y posteriormente optimizar dicho modelo, haciendo que los recursos de la computadora sean utilizados de forma eficiente (ejemplo anterior), como también hacer que la gente espere lo menos posible en una fila en un supermercado (**Nota**: ¡A menos que sea un súper de *Amazon Go* sin colas!).

¿Por qué se forma una Cola? Las causas fundamentales que hacen inevitables la aparición de una cola son dos: la primera es que la capacidad de los *servidores* que poseemos sea menor que la demanda requerida, la otra causa posible se relaciona con la variabilidad de los *tiempos de servicio*, o bien suceden ambas simultáneamente. Las colas producidas por la primera causa no representa una gran complejidad para enfrentarlas [3], mientras que si sucede la segunda representará un desafío de diseño y gestión.

¿Como modelar correctamente un sistema de cola? Para responder esta pregunta en detalle, se deben definir unos conceptos básicos (algunos mencionados en esta introducción) primero, y de eso nos encargaremos en la sección de *marco teórico* que sigue a esta sección.

2. Marco Teórico

2.1. Conceptos Fundamentales

2.1.1. Sistema

Un *objeto* (fenómeno real) cuyas partes o componentes son una colección de *entidades* que se relacionan con al menos alguno de los demás componentes, es decir, interactúan entre ellos y cumplen un fin o propósito. Un sistema no necesariamente es material, también puede ser *conceptual*.

Para analizar como se comporta un sistema, si bien podemos experimentar con él directamente, esto no suele ser rentable (e inclusive factible) en la mayoría de los casos y es más conveniente experimentar con un *Modelo del sistema*.

2.1.2. Modelo

Representación *simplificada* de la realidad, que facilita su comprensión y el estudio de su comportamiento. Debe mantener un *equilibrio* entre sencillez y capacidad de representación.

Para representar un sistema, podemos utilizar un *modelo físico* o un *modelo matemático*.

Modelo físico: Normalmente son construcciones en escala reducida o simplificada [4] de obras, máquinas o sistemas de ingeniería para estudiar en ellos su comportamiento y permitir así perfeccionar los diseños, antes de iniciar la construcción de las obras u objetos reales. Por ese motivo, a este tipo de modelo se le suele llamar también modelo reducido o modelo simplificado.

Modelo matemático: Cualquier esquema simplificado e idealizado de un objeto (fenómeno real), constituido por símbolos y operaciones (relaciones) matemáticas; es decir, una forma de representar cada uno de los tipos de entidades que intervienen en un cierto proceso de la realidad mediante objetos matemáticos. Las relaciones matemáticas formales entre los objetos del modelo, deben representar de alguna manera las relaciones reales existentes entre las diferentes entidades o aspectos del sistema u objeto real [5]. Este tipo de modelo requerirá que se pueda seguir el camino inverso al modelado, permitiendo reinterpretar en la realidad las predicciones del modelo.

Planteado un modelo matemático, tendremos que optar por realizar una *Solución Analítica*, y/o una *Simulación por computadora*.

2.1.3. Soluciones Analíticas

Una vez ”traducido” o ”representado” cierto problema en forma de modelo matemático, si este no es extremadamente complejo y/o caótico, se puede proceder a aplicar el cálculo, el álgebra y otras herramientas matemáticas para deducir el comportamiento del sistema bajo estudio. De esta forma, se consiguen predicciones *exactas* que se reflejan en el sistema real (llamamos a eso solución analítica).

Una forma alternativa podría ser buscar una *solución heurística*, es decir buscar una solución aproximada a la exacta basado en una estrategia, método, o criterio usado para hacer más sencilla la predicción en modelos complejos.

No siempre existe un método heurístico o es conveniente utilizar uno para hallar una solución a un problema complejo, o bien este es muy caótico para modelarlo solo con elementos matemáticos tradicionales y que tenga una solución realista, o quizás el tiempo requerido o el costo de recursos hace que no sea factible utilizar uno.

Para todos estos casos, la mejor opción es optar por hacer una *simulación* del modelo.

2.1.4. Simulación por computadora

Es un intento de modelar situaciones de la vida real por medio de un programa de computadora, lo que requiere ser estudiado para ver cómo es que trabaja el sistema y hacer predicciones sobre su comportamiento y que si bien no son exactas, se apunta a llegar a soluciones realistas y aproximadas a lo que en el sistema sucede. Su comportamiento puede cambiar en cada simulación según el conjunto de parámetros iniciales supuestos por el entorno.

En una simulación también se puede buscar generar una muestra de escenarios representativos para un modelo en que una enumeración analítica completa de todos los estados posibles serían prohibitivos o imposibles.

La simulación por computadora se ha convertido en una parte útil del modelado de muchos sistemas naturales en física, química y biología, y sistemas humanos como la economía y las ciencias sociales (sociología computacional), etc...

2.2. Teoría de Colas

2.2.1. Definición

La teoría de colas [2] es el estudio matemático de las colas o líneas de espera dentro de un sistema. Esta teoría estudia factores como el tiempo de espera medio en las colas o la capacidad de trabajo del sistema sin que llegue a colapsar. Dentro de las matemáticas, la teoría de colas se engloba en la *investigación de operaciones* y es un complemento muy importante a la *teoría de sistemas* y la *teoría de control*. Se trata así de una teoría que encuentra aplicación en una amplia variedad de situaciones como negocios, comercio, industria, ingenierías, transporte y logística o telecomunicaciones. El matemático danés *Agner Krarup Erlang*, publicó el primer artículo sobre la teoría de colas en 1909.

Un sistema de colas puede modelarse a través de un modelo matemático, en el cuál realizaremos predicciones analíticamente y/o a partir de una simulación.

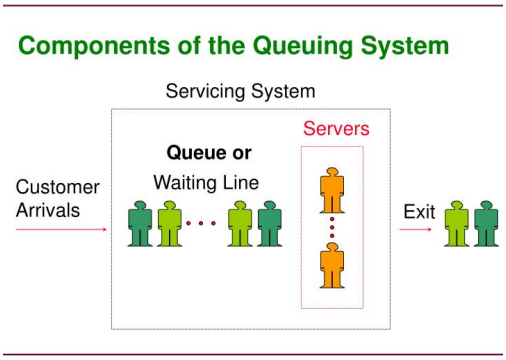

Figura 1: Componentes básicos de un sistema de cola.

2.2.2. Objetivos

Algunos de los objetivos de modelar un sistema de colas consisten en:

- Identificar el nivel óptimo de capacidad del sistema que minimiza su coste (el coste puede ser económico, u otro recurso como el tiempo de espera de los elementos de la cola).
- Evaluar el impacto que las posibles alternativas de modificación en capacidad del sistema tendrían en el coste.
- Establecer un balance óptimo entre consideraciones cuantitativas de costes y cualitativas de servicio.
- Prestar atención al tiempo de permanencia en el sistema o en la cola de espera.

2.2.3. Elementos existentes en un Modelo de Colas

Para modelizar un sistema de colas, hay que tener en cuenta al menos estas 8 características:

- Proceso básico de colas [2]: Los clientes que requieren un servicio se generan en una fase de entrada. Estos clientes entran al sistema y se unen a una cola. En determinado momento se selecciona un miembro de la cola, para proporcionarle el servicio, mediante alguna regla conocida como disciplina de servicio. Luego, se lleva a cabo el servicio requerido por el cliente en un mecanismo de servicio, después de lo cual el cliente sale del sistema de colas.
- Fuente de entrada o población potencial: Una característica de la fuente de entrada es su tamaño. El tamaño es el número total de clientes que pueden requerir servicio en determinado momento. Puede suponerse que el tamaño es infinito o finito.
- Cliente: Es todo individuo de la población potencial que solicita servicio como por ejemplo una lista de trabajo esperando para imprimirse.
- Capacidad de la cola: Es el máximo número de clientes que pueden estar haciendo cola (antes de comenzar a ser servidos). De nuevo, puede suponerse finita o infinita.
- Disciplina de la cola: La disciplina de la cola se refiere al orden en el que se seleccionan sus miembros para recibir el servicio. Por ejemplo, puede ser:

    - FIFO (first in first out) primero en entrar, primero en salir, según la cual se atiende primero al cliente que antes haya llegado.
    - LIFO (last in first out) también conocida como pila que consiste en atender primero al cliente que ha llegado el último.
    - RSS (random selection of service) que selecciona los clientes de manera aleatoria, de acuerdo a algún procedimiento de prioridad o a algún otro orden.
    - PS (Processor Sharing) sirve a los clientes igualmente. La capacidad de la red se comparte entre los clientes y todos experimentan con eficacia el mismo retraso.
- Mecanismo de servicio: El mecanismo de servicio consiste en una o más instalaciones de servicio, cada una de ellas con uno o más canales paralelos de servicio, llamados **servidores**.
- Redes de colas: Sistema donde existen varias colas y los trabajos fluyen de una a otra. Por ejemplo: las redes de comunicaciones o los sistemas operativos multitarea.
- El proceso de servicio: Define cómo son atendidos los clientes.

2.2.4. Notación de Kendall

David G. Kendall [2] introdujo una notación para describir las colas y sus características, que originalmente era de la forma **A/B/C** y que luego ha sido extendida a **1/2/3/(4/5/6)** donde los números se reemplazan con:

1. Un código que describe el proceso de llegada. Los códigos usados son:
    - M para "Markoviano"(la tasa de llegadas sigue una distribución de Poisson), significando una distribución exponencial para los tiempos entre llegadas.
    - D para unos tiempos entre llegadas deterministas, es decir, no siguen un proceso probabilista a la hora de su determinación.
    - G para una "distribución general"de los tiempos entre llegadas, o del régimen de llegadas.
2. Un código similar que representa el proceso de servicio (tiempo de servicio). Se usan los mismos símbolos.
3. El número de canales de servicio (o servidores).
4. La capacidad del sistema, o el número máximo de clientes permitidos en el sistema incluyendo esos en servicio. Cuando el número está al máximo, las llegadas siguientes son rechazadas.
5. Disciplina de cola, es decir, el orden de prioridad en la que los trabajos en la cola son servidos:
    - FIFO (First In First Out)
    - LIFO (Last In First Out)
    - RSS (Random selection of service)
    - PS (Processor Sharing)
6. El tamaño del origen de las llamadas. El tamaño de la población desde donde los clientes vienen. Esto limita la tasa de llegadas.

Los números 4/5/6 son opcionales; si alguno de estos no son especificados por convención se toma que 4 (límite de clientes) es ∞, 5 (disciplina de cola) es *FIFO*, y 6 (tamaño de la población de origen) es también ∞. La utilización de una *n* no indica infinito, sino un valor finito no especificado.

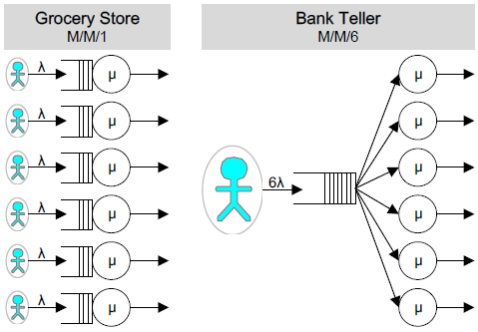

Figura 2: Modelos con su representación en la notación de Kendall.

2.2.5. Enfoque de Modelo M/M/1

En este modelo se dispone sólo de un canal para dar servicio, las llegadas siguen un proceso de Poisson y la distribución del tiempo de servicio es exponencial. En este modelo la tasa de promedio de arribos al sistema se denomina con λ y la tasa promedio de servicio con μ . La capacidad del sistema es ilimitada, es decir, el tamaño de la cola se toma teóricamente como infinita y la disciplina de la cola es FIFO. El factor de utilización es conocido y es $\rho = \frac{\lambda}{\mu}$

La condición necesaria y suficiente para que un modelo M/M/1 tenga solución de equilibrio, es que $\rho < 1$, también denominada **condición de estabilidad**.

2.2.6. Medidas de Desempeño en un Modelo de Cola M/M/1

Teniendo en cuenta que la cantidad de clientes atendidos en el servidor por unidad de tiempo la denominamos con μ y la cantidad de llegadas por unidad de tiempo con λ, podemos definir las siguientes medidas de rendimiento:

- **Utilización del servidor**: $\rho = \frac{\lambda}{\mu}$
- **Tiempo promedio en la cola**: se denomina con W_q
- **Número promedio de clientes en la cola:** se calcula como $L_q = \lambda W_q$
- **Tiempo promedio en el sistema:** se denomina con W_s y se calcula $W_s = W_q + \frac{1}{\mu}$
- **Número promedio de clientes en el sistema:** se calcula como $L_s = \lambda W_s = L_q + \frac{\lambda}{\mu}$
- **Probabilidad de n clientes en el sistema:** $P_n = (1-p)p^n$
- **Probabilidad de denegación de servicio:** es la probabilidad de que haya más clientes de lo que la cola puede soportar, por lo tanto se le denega el servicio al cliente. El mismo se calcula como: $1 - \sum_0^n P_i$

2.2.7. Otros Tipos de Modelos de Colas

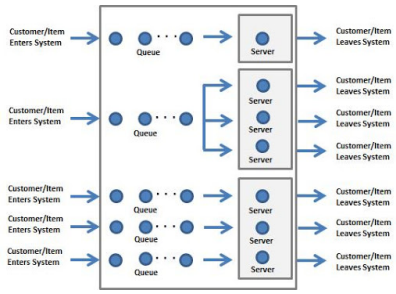

Figura 3: Modelos con: Cola y servidor único, única cola y 3 servidores, 3 colas y 3 servidores

Modelo M/M/c/c o Erlang-B: Modelo de colas exponencial con un Nº limitado de servidores y con pérdidas. Consideramos el Nº de fuentes = ∞, tendremos una tasa de llegadas aleatorias λ y una tasa de servicio μ cte. en cada servidor. Cada estado representa el Nº de unidades en cola de espera. El sistema tiene c servidores con un Nº máximo de unidades de c elementos. Entonces no hay cola de espera, y las unidades que se encuentren los servidores ocupados se perderán sin posibilidad de ser almacenadas. Se llama *modelo con pérdidas LCC*, y lo utilizan las centrales telefónicas.

Modelo M/M/c o Erlang-C: Modelo con un Nº finito de c servidores, cola infinita. Consideramos población infinita con una tasa de llegadas aleatoria exponencial λ y una tasa de servicio μ en cada servidor. Las llamadas que no se puedan servir pasarán a la cola de espera. Cada estado representa el Nº de unidades en el sistema. En consecuencia, habrá hasta c unidades atendidas simultáneamente y un Nº ilimitado de unidades esperando en cola. Las c primeras unidades serán atendidas por los c servidores. La primera unidad que irá a la cola será la $c+1$. Entonces cuando el estado del sistema sea superior a c, todos los c servidores estarán activos, y, por lo tanto, la tasa de servicio será constante, de valor $c\mu$. Las llegadas se producen con una tasa λ, independientemente del Nº de unidades en el sistema.

Modelo M/M/∞: Este modelo es un caso particular del modelo anterior, M/M/c, donde no hay tiempo de espera ni rechazo de unidades, ya que el sistema siempre dispone de recursos libres. Consideraremos pues, al igual que en el caso anterior, una tasa de llegadas λ y una tasa de servicio μ para cada servidor. A partir de la notación de Kendall, podemos observar que este sistema es multiservidor con infinitos servidores; por lo tanto, nunca habrá ninguna unidad en la cola de espera, ya que siempre habrá un servidor libre para atender a la unidad que llegue. Este sistema no necesita ninguna cola, pues se trata, simplemente, de un conjunto de servidores atendiendo a todas las unidades recibidas.

3. Metodología

Para componer este trabajo práctico, se han utilizado 2 herramientas: primero una simulación íntegramente en el lenguaje **Python**[7], escrito el código en el IDE **Visual Studio Code**[8] con su correspondiente *Plugin*.

Luego, otra simulación en **AnyLogic**[9] con la versión gratuita del mismo. También se realizó un modelo analítico para comparar con los resultados de las simulaciones

3.1. Librerías y Módulos de Python Utilizados

Numpy [10] Se hizo uso del generador Mersenne-Twister de esta librería para obtener números aleatorios en una distribución uniforme normalizada $\mathscr{U}(0,1)$, necesario para nuestra función que a partir de una lista de uniformes, generamos números en distribución exponencial. También se utilizaron funciones varias para manipulación de arrays.

Pyplot [11] Este módulo de la librería **matplotlib** se utilizó para graficar las *Funciones de distribución acumulada* de una distribución obtenida del resultado de la simulación contra la misma función calculada en forma analítica.

3.2. Convenciones

Se supuso que la cola arranca vacía y el servidor desocupado. La simulación termina al llegar a n demoras, con $n \in \mathbb{Z}$. Además, se estableció las variables aleatorias correspondientes a los arribos y a las partidas se distribuyan de forma exponencial.

Las modalidades de cola tanto infinita como finita son soportadas, teniendo en cuenta en el caso de la finita que el sistema dejara sin servicio a los clientes que lleguen una vez la cola este completa.

Cabe aclarar que interesa únicamente sacar conclusiones sobre la simulación cuando esta entra a su estado estable, que es justamente donde se observan los valores calculados de forma teórica (2.2.5) y por ende es posible compararlos con los resultados de la simulación.

3.3. Métodos de Resolución Aplicados

Para la implementación de esta simulación se adopto un enfoque de programación orientada a eventos, en la que primero se **inicializa un conjunto de variables** que irán evolucionando a la largo de la ejecución de la simulación, luego se **ejecuta un bucle** mientras que el numero de clientes demorados sea menor a la cantidad requerida de demoras.

En el loop, primero se **decide si el siguiente evento sera un arribo o una partida**, según cual sea el mínimo (el mas próximo en fin) y se avanza el reloj de simulación a ese tiempo.

Después, se **computan las estadísticas** correspondientes al área debajo de la función de numero clientes en cola ($Q(t)$) y de utilización del servidor ($U(t)$). Lo cual se logra sumando al valor obtenido en la iteración anterior el producto entre el numero de clientes en cola con el tiempo que transcurrió entre ese momento y el actual. Notando que si bien esto es equivalente a la formula original

$$\sum_{i=0}^{\infty} i\, T_i \tag{1}$$

donde i es el numero de clientes en cola y T_i es el tiempo en el que la cola tuvo esa longitud durante la simulación.

En cambio

$$\int_0^{T(n)} Q(t)dt \tag{2}$$

sugiere que el calculo de un promedio continuo, es decir de calculo incremental. A diferencia de (1) que evoca una suma de términos en cada momento.

Presentándose una situación análoga en el caso de $U(t)$, donde se define una función $B(t)$

$$B(t) = \begin{cases} 0 & \text{si el servidor esta ocupado en ese momento t} \\ 1 & \text{si el servidor esta desocupado en ese momento t} \end{cases} \tag{3}$$

La cual se acumula a través de

$$\sum_{i=0}^{\infty} U_i \tag{4}$$

donde U_i son los fragmentos de tiempo en los que el servidor se encuentra en estado ocupado durante la simulación
Pero esto es posible computarlo mejor si se expresa con la ecuación

$$\int_0^{T(n)} B(t)dt \tag{5}$$

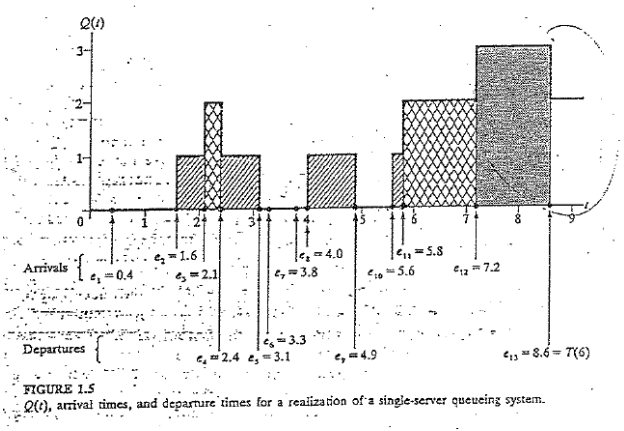

Figura 4: Q(t) - Tamaño de la cola a través del tiempo

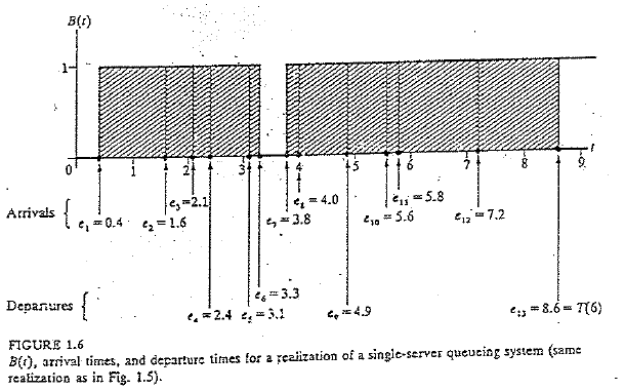

Figura 5: B(t) - Servidor ocupado/desocupado

Luego, se **ejecuta la rutina correspondiente** a el tipo de evento seleccionado [1].
Si es un **arribo**, consta de los siguientes pasos:

- Generar el próxima arribo
- Si el servidor esta desocupado (como lo esta en la fase inicial)
  - Sumar uno al numero de clientes demorados
  - Poner al servidor en estado ocupado
  - Agendar la próxima partida de este cliente
- Si el servidor esta ocupado
  - Agregar un cliente a la cola
  - Guardar el tiempo de arribo de este

Si es una **partida**, se secuencian los siguientes pasos:

- Si la cola esta vacía
  - Poner el servidor en estado desocupado
  - Quitar de consideración la posibilidad de que el próximo evento sea una partida
- Si la cola no esta vacía
  - Computar la demora para el cliente en que se va a servir y se suma esta a la demora total
  - Agregar uno al numero de clientes demorados
  - Agendar la partida para este cliente
  - Reducir la cola en uno y por ende el numero de clientes en cola

Cabe destacar que tanto en la ejecución de la rutina de arribo y la de partida, la demora total en cola es computada a través de hacer:

$$\sum_{i=1}^{n} D_i \tag{6}$$

Siendo en el arribo calculada cuando el servidor se encuentra en estado desocupado, sumando $D_i = 0$, por lo que es posible sacar esta parte de la simulación sin afectar el resultado final. El otro escenario en el que se computa la demora es al producirse una partida a la vez que la cola no esta vacía, por lo que el se calculara la demora del cliente que entra

en servicio, a partir de la diferencia entre el tiempo actual y el tiempo de arribo correspondiente al primer cliente de la cola de tiempos de arribo.

Mientras se va iterando, se procede a realizar el **reporte de los resultados de la simulación hasta ese momento**, que serian las medidas de rendimiento mencionadas en (2.2.6), para luego agregarlos a un arreglo que sera devuelto al final de la simulación para su posterior análisis.

Para Lq (cantidad promedio de clientes en cola) y U (utilización del servidor) se dividen los valores estadísticos computados anteriormente, área debajo de la curva de Q(t) y de U(t) respectivamente, por el tiempo que llevo la simulación para las n demoras requeridas T(n). Quedando los estimadores:

$$\hat{q}(n) = \frac{\int_0^{T(n)} Q(t)dt}{T(n)} \tag{7}$$

$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t)dt}{T(n)} \tag{8}$$

En cuanto a Wq esta se computa en la simulación a través del estimador:

$$\hat{d}(n) = \frac{\sum_{i=1}^{n} D_i}{n} \tag{9}$$

Además, se cuenta con los estimadores de las medidas de rendimiento para el sistema: W y L, que se calculan de forma similar a su calculo analítico (2.2.6) solo que a partir de los estimadores anteriores

Tiempo promedio de espera en el sistema

$$\hat{w}(n) = d(n) + \frac{1}{\mu} \tag{10}$$

donde μ es la tasa de servicio

Cantidad promedio de clientes en el sistema

$$\hat{l}(n) = \lambda \cdot w(n) \tag{11}$$

donde λ es la tasa de arribos

Estos resultados serán comparados con los valores esperados, que son obtenidos de forma analítica a través de las fórmulas de la sección nombrada, tanto en forma gráfica como a través de tablas. Esto se realiza observando el sistema en su estado estable, dado que es donde se evidencia la convergencia o no de los valores teóricos, determinando a su vez si la simulación esta bien realizada o no.

También se realizó una simulación sencilla en *AnyLogic* configurando todos los mismos parámetros, cuyos resultados también son tenidos en cuenta en la comparación.

Los **escenarios a analizar** en este trabajo son:

Configuración	**Valor λ**	**Valor μ**	**Relación $\frac{\lambda}{\mu}$**
0	0.50	2.00	25 %
1	1.00	2.00	50 %
2	1.50	2.00	75 %
3	2.00	2.00	100 %
4	2.50	2.00	125 %

Se debe tener en cuenta para la comparación, que si

- λ (tasa de arribos) $\geq \mu$ (tasa de servicio)

Como ocurre en las configuraciones 3 y 4, ya no es posible calcular los valores esperados (2.2.5), a excepción de la utilizaciones del servidor que en estos casos convergerá siempre en 1. Ya que lógicamente al llegar mas clientes por unidad de tiempo de lo que el servidor puede atender, la longitud de la cola y la demora promedio en esta tiende a infinito.

4. Resultados

4.1. Gráficas Simulación Python

Gráficas obtenidas en Python para 10 corridas de programa con 10000 *delayed costumers* y un tamaño de cola infinita.

4.1.1. Primer caso

Arrival rate (λ) = 0.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 25\,\%$

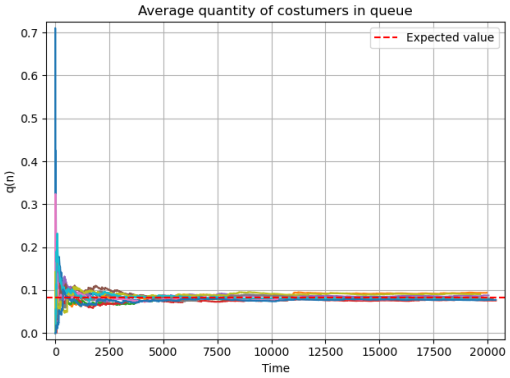

Figura 6: N^{o} prom clientes en cola, caso 1

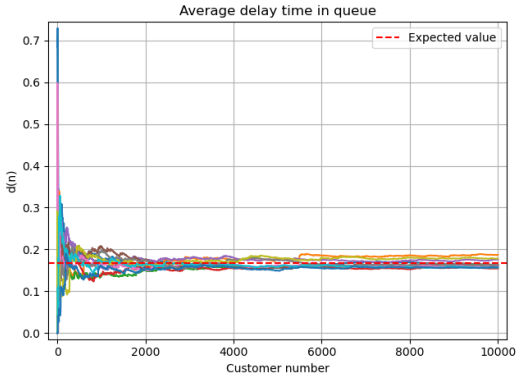

Figura 7: Tiempo promedio en cola, caso 1

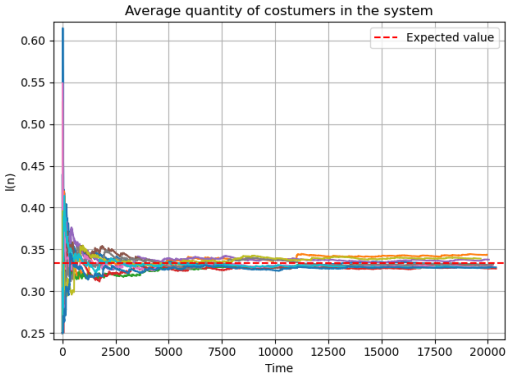

Figura 8: Núm. prom de clientes en sistema, caso 1

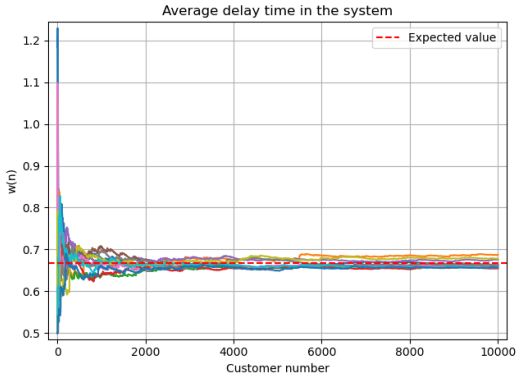

Figura 9: Tiempo promedio en sistema, caso 1

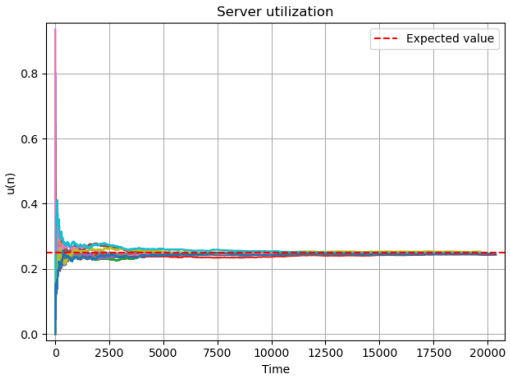

Figura 10: Utilización del servidor, caso 1

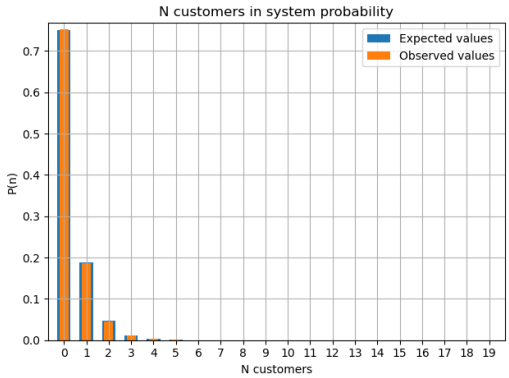

Figura 11: Prob N clientes en cola, caso 1

4.1.2. Segundo caso

Arrival rate (λ) = 1.0 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 50\,\%$

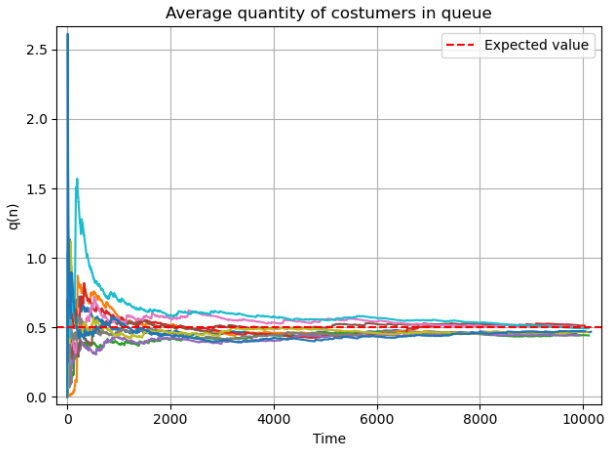

Figura 12: N^o prom clientes en cola, caso 2

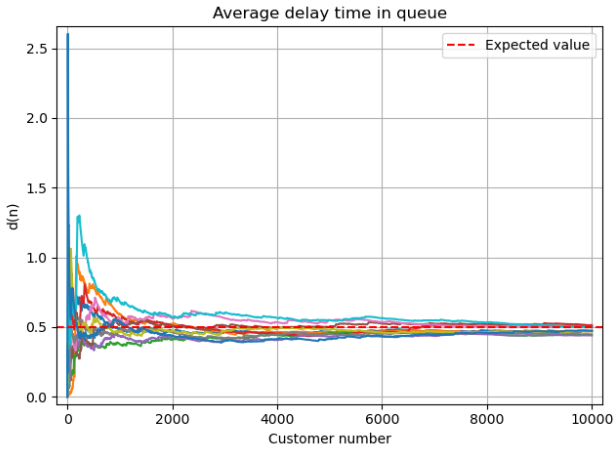

Figura 13: Tiempo promedio en cola, caso 2

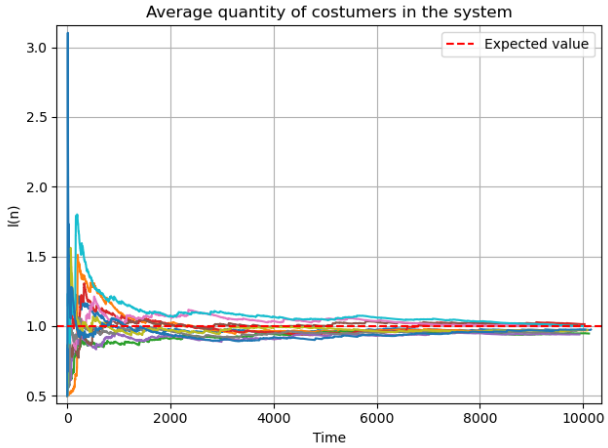

Figura 14: Núm. prom de clientes en sistema, caso 2

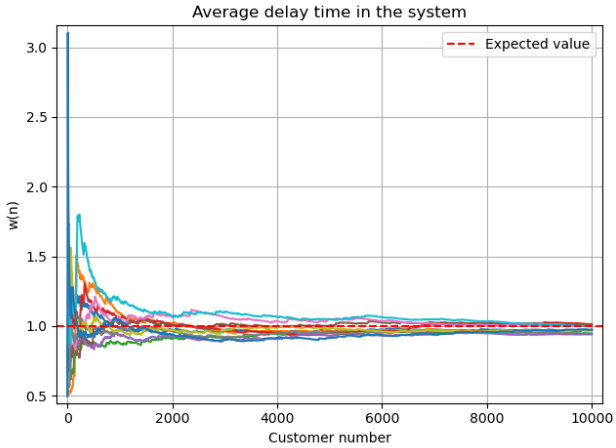

Figura 15: Tiempo promedio en sistema, caso 2

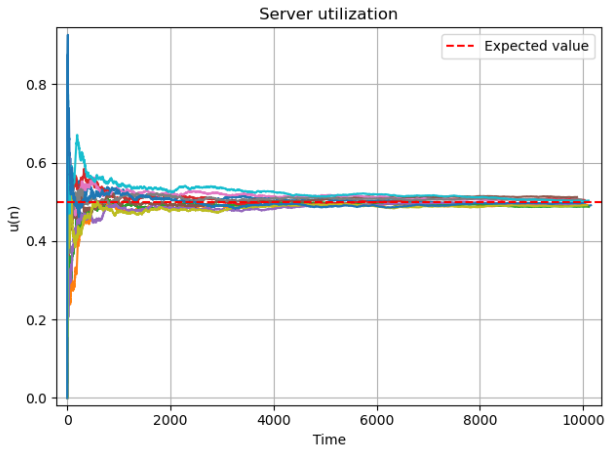

Figura 16: Utilización del servidor, caso 2

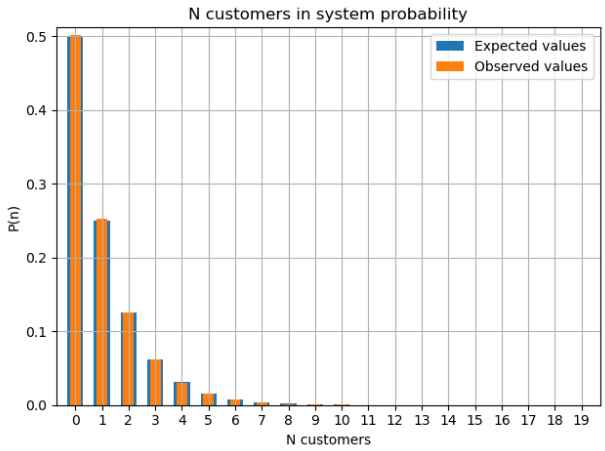

Figura 17: Prob N clientes en cola, caso 2

4.1.3. Tercer caso

Arrival rate (λ) = 1.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 75\,\%$

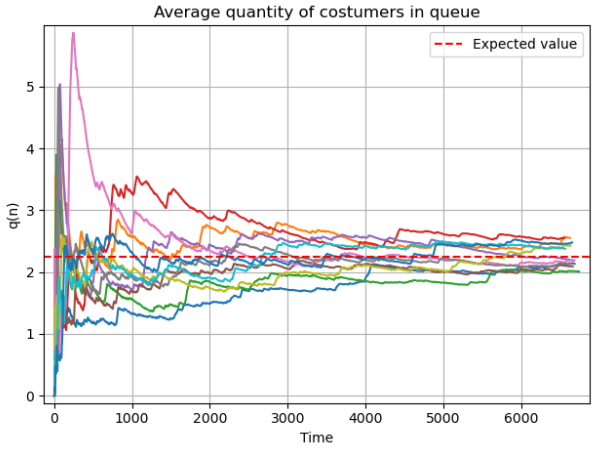

Figura 18: Nº prom clientes en cola, caso 3

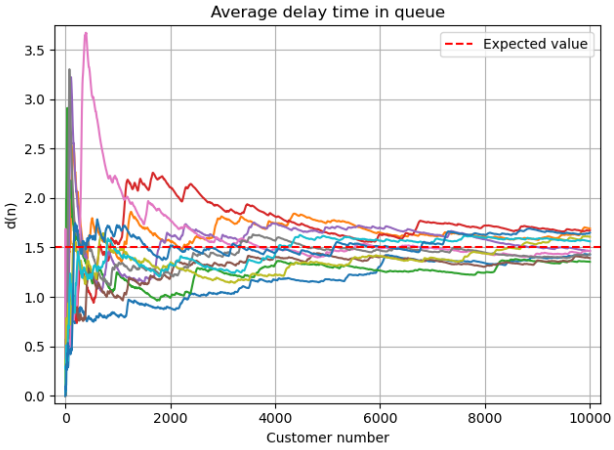

Figura 19: Tiempo promedio en cola, caso 3

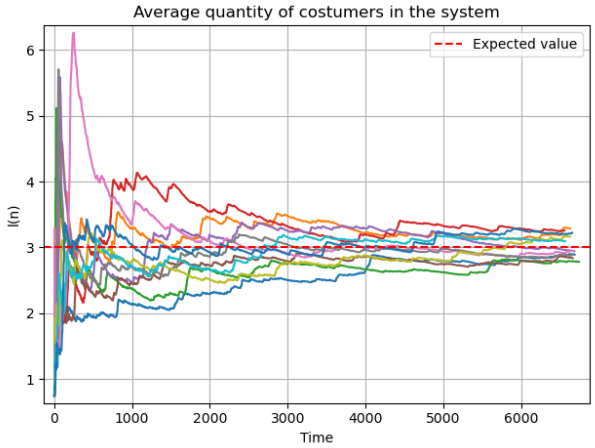

Figura 20: Núm. prom de clientes en sistema, caso 3

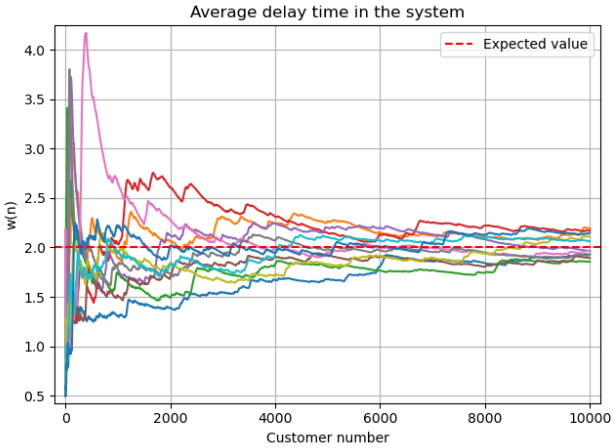

Figura 21: Tiempo promedio en sistema, caso 3

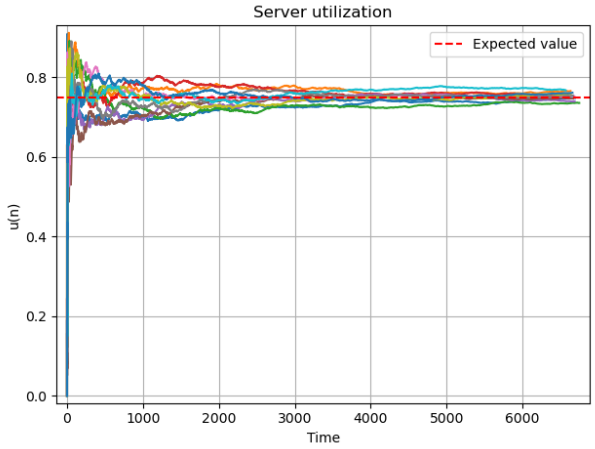

Figura 22: Utilización del servidor, caso 3

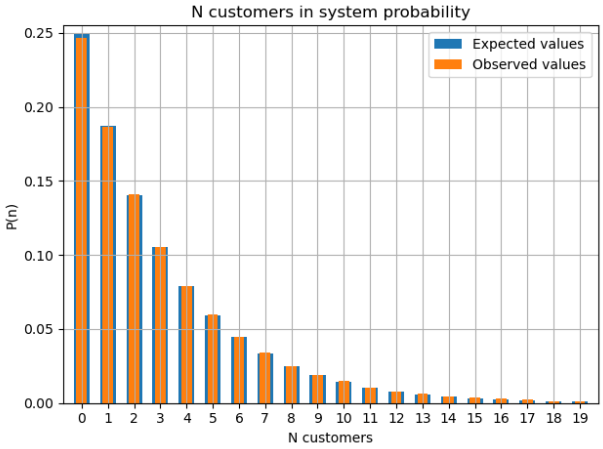

Figura 23: Prob N clientes en cola, caso 3

4.1.4. Cuarto caso

Arrival rate (λ) = 2.0 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 100\,\%$

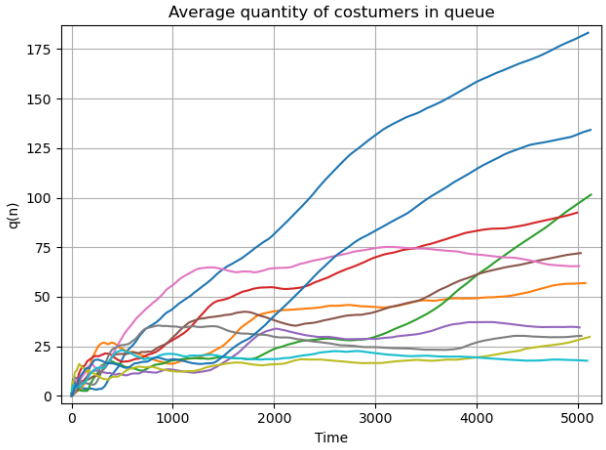

Figura 24: Nº prom clientes en cola, caso 4

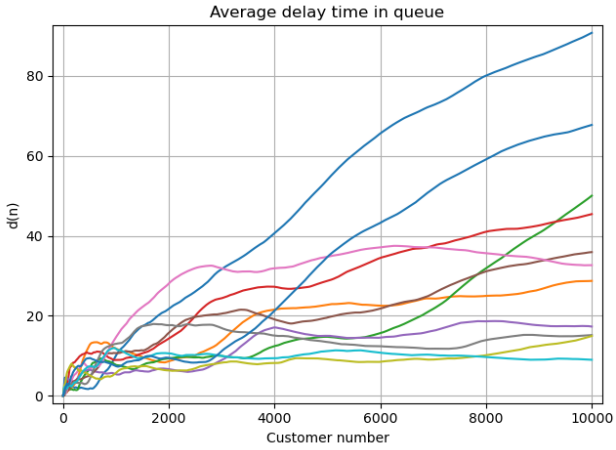

Figura 25: Tiempo promedio en cola, caso 4

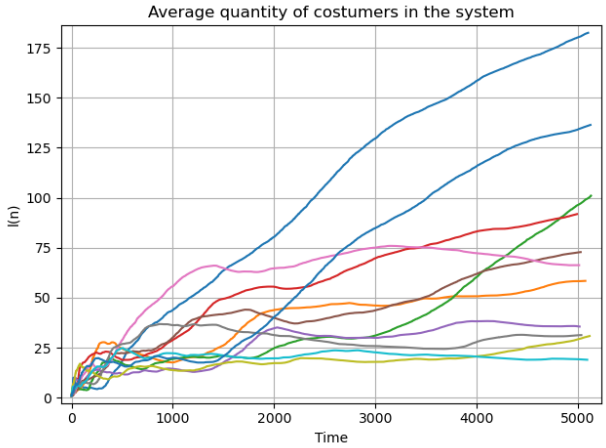

Figura 26: Núm. prom de clientes en sistema, caso 4

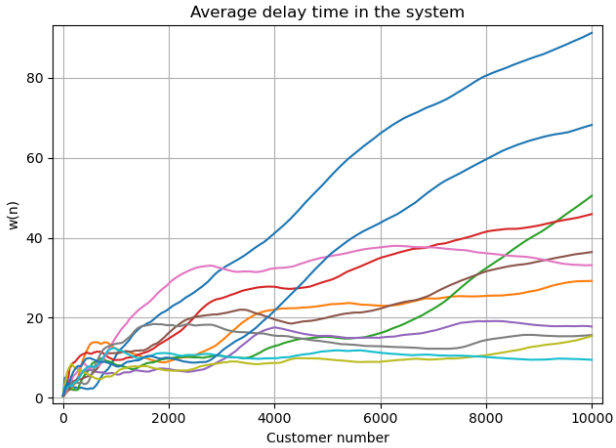

Figura 27: Tiempo promedio en sistema, caso 4

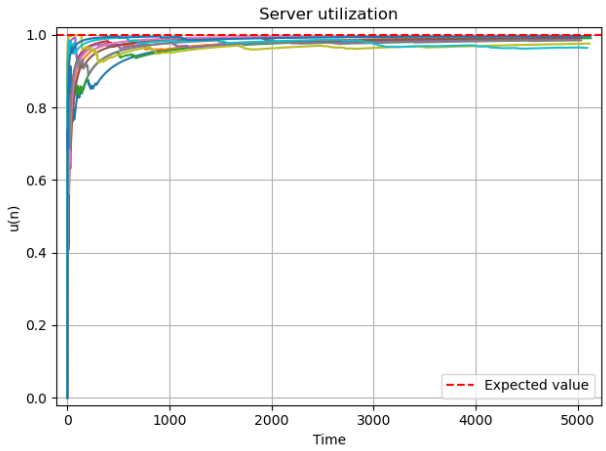

Figura 28: Utilización del servidor, caso 4

4.1.5. Quinto caso

Arrival rate (λ) = 2.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu}$ = 125 %

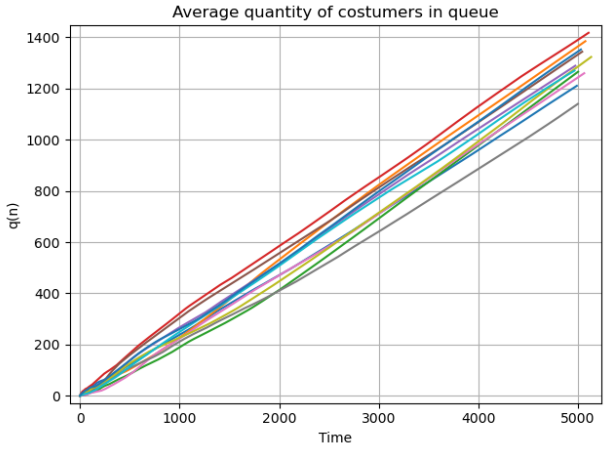

Figura 29: Nº prom clientes en cola, caso 5

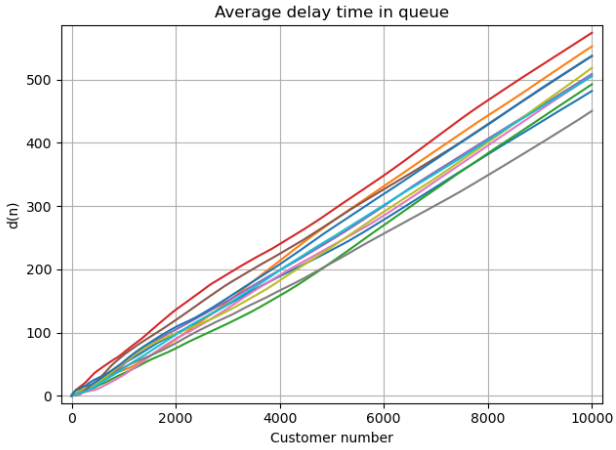

Figura 30: Tiempo promedio en cola, caso 5

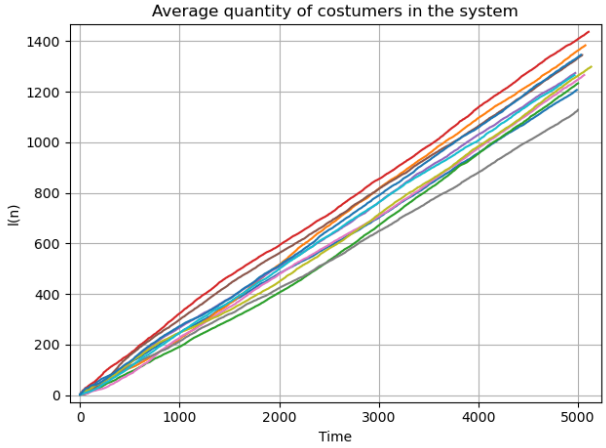

Figura 31: Núm. prom de clientes en sistema, caso 5

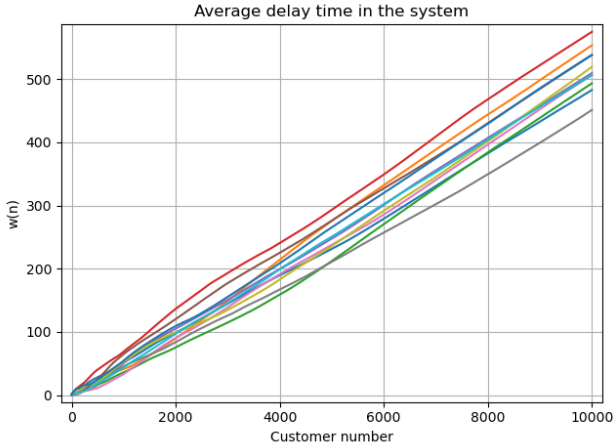

Figura 32: Tiempo promedio en sistema, caso 5

Figura 33: Utilización del servidor, caso 5

4.2. Gráficas Simulación AnyLogic

Gráficas obtenidas en AnyLogic PLE considerando un tamaño de cola infinita. Las medidas de desempeño a nivel sistema (W y L) aparecen como Ws y Ls.

4.2.1. Modelo de AnyLogic General

Resultado final del modelo (sin ejecutar aún el mismo)

Figura 34: Modelo AnyLogic

4.2.2. Primer caso

Arrival rate (λ) = 0.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 25\,\%$

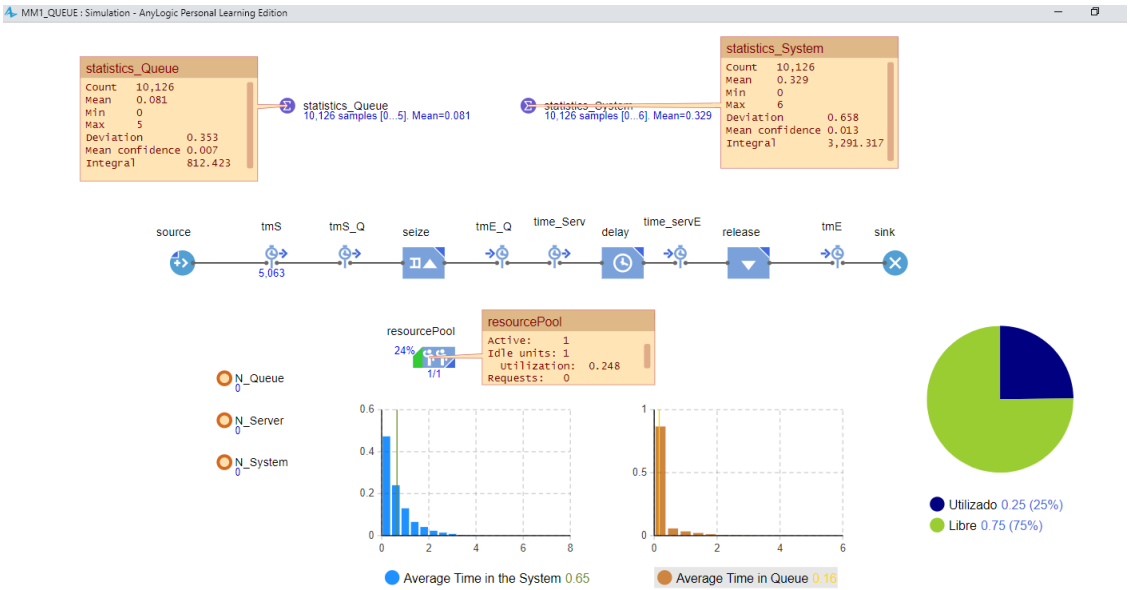

Figura 35: $\rho = 0{,}248$; $W_s = 0{,}65$; $W_q = 0{,}16$; $L_q = 0{,}081$; $L_s = 0{,}329$

4.2.3. Segundo caso

Arrival rate (λ) = 1.0 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 50\,\%$

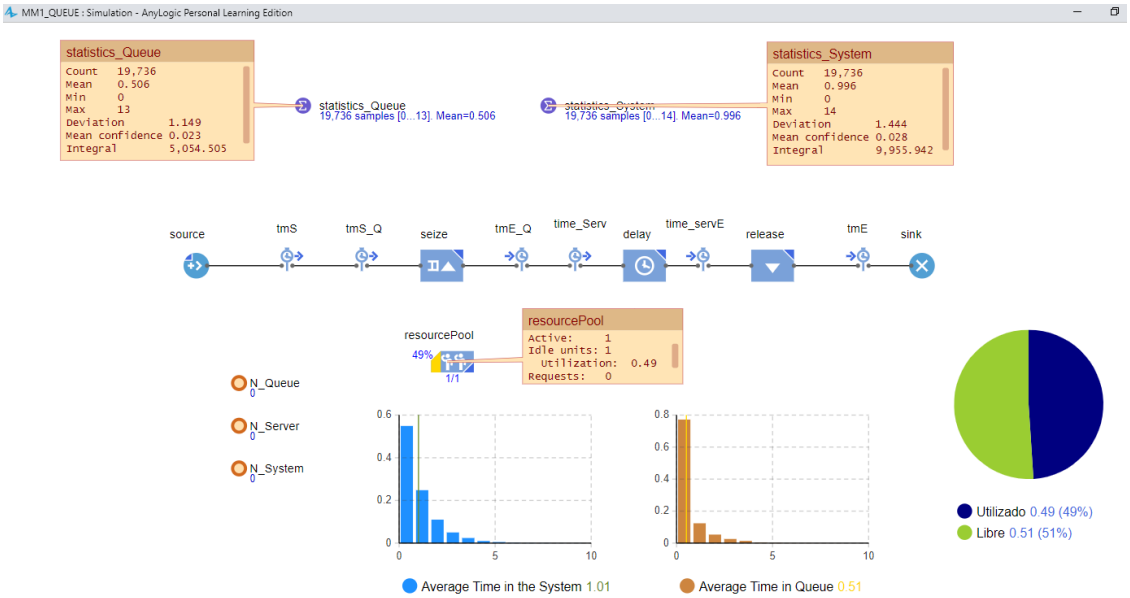

Figura 36: $\rho = 0{,}49$; $W_s = 1{,}01$; $W_q = 0{,}51$; $L_q = 0{,}506$; $L_s = 0{,}996$

4.2.4. Tercer caso

Arrival rate (λ) = 1.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 75\,\%$

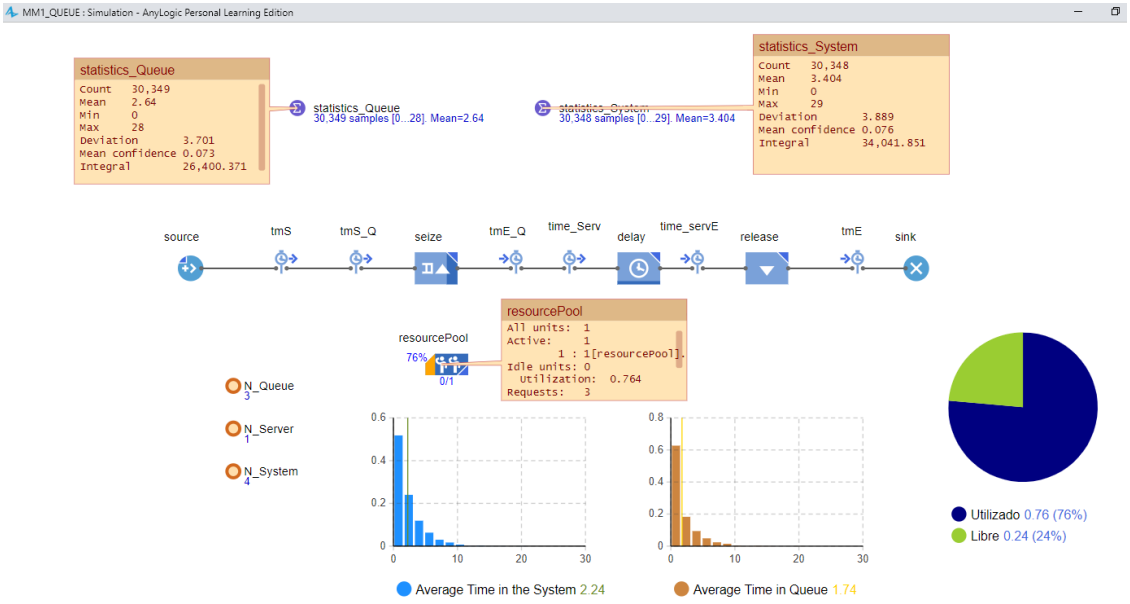

Figura 37: $\rho = 0{,}764$; $W_s = 2{,}24$; $W_q = 1{,}74$; $L_q = 2{,}64$; $L_s = 3{,}404$

4.2.5. Cuarto caso

Arrival rate (λ) = 2.0 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 100\,\%$

Figura 38: $\rho = 0{,}991$; $W_s = 21{,}92$; $W_q = 21{,}43$; $L_q = 42{,}889$; $L_s = 43{,}88$

4.2.6. Quinto caso

Arrival rate (λ) = 2.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 125\,\%$

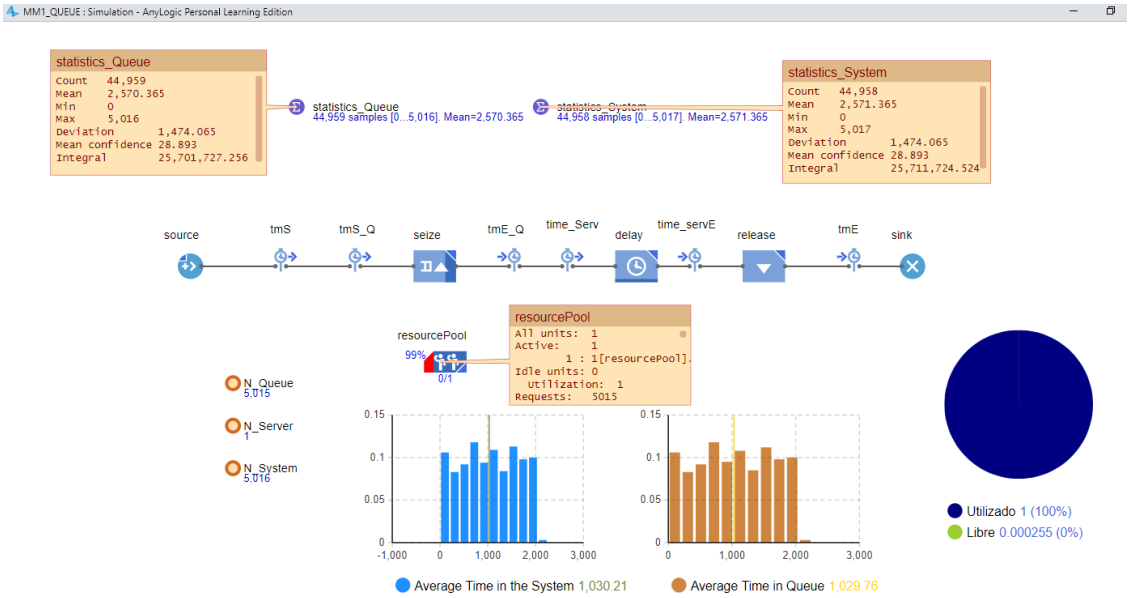

Figura 39: $\rho = 1$; $W_s = 1030{,}21$; $W_q = 1029{,}76$; $L_q = 2570{,}37$; $L_s = 2571{,}37$

4.3. Tablas de Resultados

4.4. Medidas de rendimiento principales

Para cada caso: Resultados del modelo analítico contra la simulación en Python (junto a una proporción de exactitud). Exactamente lo mismo con el modelo analítico contra la simulación en AnyLogic. Un resultado favorable en las proporciones sería $1 \pm 0{,}15$.

Nota: Aparte de la utilización del servidor, no se comparan otros parámetros de rendimiento cuando $\lambda \geq \mu$.

Utilización del Servidor (ρ)

Caso	Relación $\frac{\lambda}{\mu}$	Analítica	Python	Proporción	AnyLogic	Proporción
1	25 %	0.250	0.247426	1.010405	0.248	1.008064
2	50 %	0.500	0.498473	1.003062	0.490	1.020408
3	75 %	0.750	0.753450	0.995421	0.764	0.981675
4	100 %	1.000	1.000000	1.000000	0.991	1.009081
5	125 %	1.000	1.000000	1.000000	1.000	1.000000

Número promedio de clientes en cola (Lq)

Caso	Relación $\frac{\lambda}{\mu}$	Analítica	Python	Proporción	AnyLogic	Proporción
1	25 %	0.083	0.082254	1,009069	0,081	1,024691
2	50 %	0.500	0.478122	1,045758	0,506	0,988142
3	75 %	2.250	2.286164	0,984181	2.64	0,852272

Tiempo promedio en cola (Wq)

Caso	Relación $\frac{\lambda}{\mu}$	Analítica	Python	Proporción	AnyLogic	Proporción
1	25 %	0.166	0.165313	1.004155	0.16	1,037500
2	50 %	0.500	0.479012	1,043815	0,51	0,980392
3	75 %	1.500	1.519805	0,986968	1,74	0,862068

Número promedio de clientes en el sistema (L)

Caso	Relación $\frac{\lambda}{\mu}$	Analítica	Python	Proporción	AnyLogic	Proporción
1	25 %	0.333	0.332657	1.002034	0.329	1.012158
2	50 %	1.000	0.979012	1.021438	0.996	1.004016
3	75 %	3.000	0.990194	0.990194	3.404	0.881316

Tiempo promedio en el sistema (W)

Caso	Relación $\frac{\lambda}{\mu}$	Analítica	Python	Proporción	AnyLogic	Proporción
1	25 %	0.666	0.665313	1.002034	0.650	1.001103
2	50 %	1.000	0.979012	1.021438	1.010	1.001032
3	75 %	2.000	3.029708	0.990194	2.240	0.892857

4.5. Otras medidas de rendimiento

Resultados del modelo analítico contra la simulación en Python (junto a una proporción de exactitud). Un resultado favorable en las proporciones sería $1 \pm 0{,}15$.

Probabilidad de n clientes en sistema Se toma como ejemplo para la comparación el modelo analítico contra la simulación de Python con la configuración 2 (Es decir, el tercer caso: Arrival rate (λ) = 1.5 - Service rate (μ) = 2.0 - Ratio $\frac{\lambda}{\mu} = 75\,\%$).

Se utilizó modelo de cola infinita para obtener estos resultados; se exhiben las probabilidades para $0 \leq n < 20$.

Tamaño de n	Solución Analítica	Simulación Python	Proporción
0	0.250000	0.246550	1,013993
1	0.187500	0.186409	1,005852
2	0.140625	0.140960	0,997623
3	0.105469	0.105510	0,999611
4	0.079102	0.079201	0,998750
5	0.059326	0.059886	0,990648
6	0.044495	0.044604	0,997556
7	0.033371	0.034057	0,979857
8	0.025028	0.025113	0,996615
9	0.018771	0.019113	0,982106
10	0.014078	0.014781	0,952438
11	0.010559	0.010635	0,992853
12	0.007919	0.007978	0,992604
13	0.005939	0.006090	0,975205
14	0.004454	0.004708	0,946049
15	0.003341	0.003632	0,919878
16	0.002506	0.002988	0,838688
17	0.001879	0.002143	0,876808
18	0.001409	0.001450	0,971724
19	0.001057	0.001193	0,886001

Probabilidad de denegación de servicio Se utiliza para la comparación el modelo analítico contra la simulación de Python con la configuración 2 (Es decir, el tercer caso: ($\lambda = 1{,}5$, $\mu = 2{,}0$, Ratio $\frac{\lambda}{\mu} = 75\,\%$)

Tamaño de Cola	Solución Analítica	Python	Proporción
0	0.750000	0.751173	0,998438
2	0.421875	0.181609	2,322985
5	0.177979	0.053900	3,302022
10	0.042235	0.011664	3,620970
50	0.000001	0.000001	1.000000

5. Discusión

5.1. Interpretación de los resultados

Puede apreciarse que, como el modelo analítico muestra, en las simulaciones para valores de $\lambda \geq \mu$ el **comportamiento del sistema se torna *inestable***, pues la cola comienza a crecer infinitamente, al ser la cantidad de clientes que llegan mayor a la cantidad que el servidor puede procesar. Esta imposibilidad de pasar del estado transiente al estado estable, se evidencia tanto en el modelo analítico de colas (3.3) como en la simulación ejecutada en los casos 4 y 5. En estos gráficos, las tablas se tornan a un crecimiento lineal en tanto las métricas de espera promedio en cola y en sistema como en las de cantidad promedio de clientes en cola y en sistema.

El **caso 3** ($\lambda = 1{,}5$, $\mu = 2{,}0$, Ratio $\frac{\lambda}{\mu} = 75\,\%$) - **Secciones 4.1.3 y 4.2.4** fue el más eficiente de los presentados, con un rendimiento del servidor de aproximadamente 75 % (es decir, casi igual al valor analítico). De las configuraciones que analizamos, es la mejor candidata para ser utilizada en una implementación real de sistema de colas. Aunque la implantación de mas servidores y/o múltiples colas no seria algo vital en este caso, esto brindaría un nivel de seguridad adicional ante el caso de que la tasa de arribos aumente y se acerque peligrosamente a la cantidad de clientes que el sistema puede atender por unidad de tiempo.

El **caso 1** ($\lambda = 0{,}5$, $\mu = 2{,}0$, Ratio $\frac{\lambda}{\mu} = 25\,\%$) - **Secciones 4.1.1 y 4.2.2** , representa el escenario mas anti-económico para su implementación, dada la muy baja utilización del servidor. Por lo que si se tiene un sistema de estas características no seria algo verdaderamente útil el aumentar el numero de servidores de colas dado esa tasa de arribos y de servicio, resultando en un costo innecesario.

El **caso 2** ($\lambda = 1$, $\mu = 2{,}0$, Ratio $\frac{\lambda}{\mu} = 50\,\%$) - **Secciones 4.1.2 y 4.2.3** no fue tan bueno como el caso 3, pero tuvo una mejoría con respecto al caso 1. Las probabilidades de que haya mas de 3 o 4 clientes en cola en un momento dado son bastante bajas, y esto queda plasmado en que la mitad del tiempo el servidor está en desuso, traduciéndose en la práctica a un desperdicio de los recursos utilizados para brindar el servicio.

En los **casos restantes**, sucede lo que arriba en esta sección se describe en donde la cantidad de clientes que puede servir el sistema por unidad de tiempo no da a basto para que la cantidad de clientes en cola retroceda; para solucionar un sistema con estas características en la práctica, se debe considerar seriamente aumentar la cantidad de servidores y/o la capacidad de los mismos.

Se observa que si bien un **valor de ρ alto** (pero menor que 1) es bueno, también aumenta la probabilidad de que n clientes estén en cola para valores de n cada vez más altos, y disminuye para valores de n pequeños como 0 ó 1, lo que se observa en forma descendente desde el caso 1 hasta llegar al caso 3. Otro escenario a destacar, es que si no se posee una cola lo suficientemente grande, puede ocurrir que haya *denegación de servicio*, esto es, cuando la cola llega al número máximo de usuarios que puede soportar, no podrá atender a un nuevo usuario que requiera el servicio, denegándoselo.

Cabe resaltar que tanto el tener colas largas, es decir, con mayor probabilidad de n clientes en cola con un n grande, como la posibilidad de que el servicio sea denegado son **factores sumamente negativos en la práctica** y lo mejor es minimizarlos.

En las tablas (sección 4.4), analizando los valores que poseen las columnas de proporciones vemos que tanto los resultados en Python como los resultados de AnyLogic toman valores aproximadamente iguales a las soluciones que ofrecen las ecuaciones del modelo analítico.

Con respecto a la última tabla (sección 4.5) sobre **denegación de servicio**, esta nos da valores bastante alejados con respecto a la solución analítica. Consideramos que, o bien la metodología aplicada en el mismo no fue totalmente exacta, o bien la ecuación que nos brinda el modelo teórico no es totalmente falible para este tipo de simulación. Sin embargo, hace falta mas pruebas para determinar realmente que esto sea así, por lo que momentáneamente estos valores no son del todo exactos.

Queremos connotar que en las **gráficas que presenta AnyLogic** también se muestra información estadística muy interesante que se puede **ver en tiempo real**: cómo el **tamaño de la cola**, **los usuarios en servidor**, **el factor de utilización**, así como las gráficas referentes a las **distribuciones de los tiempos promedios** y la **utilización del servidor** que se muestra en forma de gráfico de torta. Pudiendo ser todos estas características de gran utilidad y con un bajo esfuerzo para un estudio que requiera un análisis más profundo, como por ejemplo que la **desviación de clientes en cola con respecto a la media** sea importante tenerla entre ciertos márgenes de calidad.

5.2. Limitaciones

Analizar el estado transiente o transitorio esta fuera del alcance de este trabajo, dado que a pesar de ser posible, es mucho mas complejo, tanto en términos de la simulación como tanto mas de forma analítica y a la vez de menor utilidad que analizar el estado estable.

Para la simulación en AnyLogic, estamos utilizando la versión PLE (Personal Learning Edition), pues legalmente nos corresponde esa y no sería correcto utilizar una versión mayor sin haberla pagado primero. Si bien esto nos da un marco lícito para presentar nuestros trabajos de forma pública, tiene limitaciones en el modelado y ejecución del sistema, como por ejemplo que no deja realizar múltiples ejecuciones del mismo modelo de forma automática (entre otras cosas).

5.3. Recomendaciones futuras

Queda como una posible extensión de este trabajo el utilizar mas de un servidor, es decir mantener la forma M/M/c pero cambiar el c a un numero mayor a 1 y/o colas múltiples con/sin presencia de alguna con prioridades en lugar de utilizar la disciplina FIFO.

Otras modificaciones que serian posibles, constituyen el utilizar una población finita dado que en el trabajo de supone infinita tanto en las variantes de cola infinita como finita. Es decir, cambiar el L del modelo completo M/M/c/K/L.

Por ultimo, también seria oportuno cambiar la distribución del proceso de arribos y/o el de llegadas, pasando de una modelo M/M/c a uno M/G/c por ejemplo.

6. Conclusiones

Los valores obtenidos en la simulación de Python explicitados en tablas y gráficas, se acercan mucho a los valores calculados analíticamente, por lo que podemos decir que es un método confiable para modelar un sistema de colas de la vida real, sucediendo lo mismo con las gráficas de la simulación de AnyLogic.

En este modelo, resultó simple encontrar soluciones analíticas, por lo que es bastante conveniente optar por esta opción si se piensa que el modelo no va a escalar a uno más complejo que M/M/1. Para un modelo mucho más complicado consideramos oportuno elegir desarrollar una simulación en Python o AnyLogic, y analizar los resultados.

Considerando que el modelo realizado en AnyLogic se realizó de manera más rápida, concluimos que dicho Software es confiable, eficiente y versátil, pudiendo agregar o quitar variables de manera simple y lograr simular tanto modelos simples como algunos más complejos. Sin embargo, el codificar el modelo en un lenguaje de alto nivel tiene un mayor valor didáctico, al brindar un entendimiento superior en cuanto a los detalles y alternativas de implementación. A la vez, la flexibilidad de agregar características a los modelos que puede que no sea posible dentro del software es algo de suma utilidad, aunque con el software Anylogic esta limitación es mucho menor dado que traslada los modelos a una interfaz muy cómoda para el usuario con un sistema de drag-and-drop totalmente configurable y funcionalidad en lenguaje de programacion Java.

Referencias

[1] Simulation Modeling and Analysis.
A. Law, W. D. Kelton - Mcgraw-hill Series - 2014 (5th ED.) - ISBN 978-0073401324

[2] Wikipedia ES. Teoría de Colas.
https://es.wikipedia.org/wiki/Teoría_de_colas

[3] Universidad de Montevideo. Teoría de Colas.
Apunte "De Colas y Esperas"

[4] Wikipedia ES. Modelo Físico.
https://es.wikipedia.org/wiki/Modelo_físico

[5] Wikipedia ES. Modelo Matemático.
https://es.wikipedia.org/wiki/Modelo_matemático

[6] Wikipedia ES. Simulación.
https://es.wikipedia.org/wiki/Simulación

[7] Python Doc. Python 3.8.2 | Documentación Oficial.
https://docs.python.org/3/

[8] Microsoft (VSC). Visual Studio Code Official Webpage
https://code.visualstudio.com/

[9] AnyLogic. AnyLogic | Documentación Oficial.
https://help.anylogic.com/index.jsp

[10] Numpy Doc. Numpy 1.18 | Documentación Oficial.
https://numpy.org/doc/1.18/

[11] Pyplot Doc. Pyplot 3.1.1 | Documentación Oficial.
https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.html