TP 3 - ESTUDIO DE LA SIMULACION DE UN SISTEMA MM1

Nicolás Andrés Bologna
Legajo: 43937
Universidad Tecnológica Nacional
Zeballos 1341, S2000 Rosario
nicolasandresbologna@hotmail.com

Hernán Carlos Basso
Legajo: 42945
Universidad Tecnológica Nacional
Zeballos 1341, S2000 Rosario
hernan.cbasso@gmail.com

22 de octubre de 2020

ABSTRACT

En el siguiente trabajo se busca introducir teóricamente al sistema de colas (en concreto al M/M/1 y M/M/1/K), generar un simulador de este sistema y analizar sus medidas de rendimiento. Además se realizó un modelo en Anylogic para introducirse en el software y comparar los resultados.

Keywords Probabilidad · Simulación · Teoría de Colas · MM1 · MM1K

1. Introducción

En el día a día los ingenieros en sistemas se encuentran en una posición de toma de decisiones para dar servicios a los distintos requerimientos que una organización o empresa pueda tener. En función de la calidad de su desarrollo (y de los recursos disponibles) el tiempo de espera del cliente o aplicación para obtener un servicio será mayor o menor. Desde ese punto de vista se podría decir que una de las funciones de un ingeniero es decidir qué debe esperar a qué (por ejemplo, en un servidor de APIs, que endpoint tendrá prioridad).
Es común experimentar en alguna ocasión la sensación de estar perdiendo el tiempo al esperar en una cola. El fenómeno de las colas se ha vuelto algo natural: esperar en el supermercado o en un peaje, escuchar el contestador del servicio de internet diciendo que nos encontramos en x posición de la cola de espera y y miles de situaciones cotidianas donde los recursos son limitados y no se puede brindar un servicio al instante a todos los clientes.
Las esperas por lo general no son buenas, hacen visualizar al usuario la importancia del producto o servicio que va a adquirir y puede hacerlo repensar o reelegir que producto/servicio adquirir.
Lo ideal sería que se pueda brindar el servicio en el momento en el que el cliente lo necesite pero esto nunca va a poder hacerse si se cuentan con recursos limitados, llevando a preguntarse ¿Por qué hay que esperar? ¿Cuánto hay que esperar? La respuesta es casi siempre simple, en algún momento la capacidad de servicio ha sido (o es) menor que la capacidad demandada. Esta limitación se puede eliminar invirtiendo en elementos que aumenten la capacidad. En estos casos la pregunta es: ¿Compensa invertir en máquinas o servidores? En ese caso ¿cuánto hay que escalar? La teoría de colas intenta responder a estas preguntas utilizando métodos matemáticos analíticos.

2. Marco Teórico - Colas De Espera

2.1. Presentación

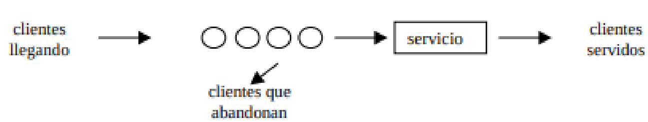
En esta sección del documento se presentará el marco teórico de la simulación de un sistema M/M/1 (perteneciente a la teoría de colas) modelado mediante mecanismos que asumen el próximo evento.

2.2. Teoría de colas

La teoría de colas 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. 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.
La teoría de colas permite modelar sistemas donde varios clientes que demandan cierto servicio o prestación, confluyen en un mismo servidor y, por lo tanto pueden registrarse esperas desde que un cliente llega al sistema y el servidor atiende sus demandas. En este sentido, la teoría es muy útil para modelar procesos tales como la llegada de datos a una cola en ciencias de la computación, la congestión de red de computadoras o de telecomunicación, o la implementación de una cadena productiva en la ingeniería industrial.
En el contexto de la informática y de las tecnologías de la información y la comunicación las situaciones de espera dentro de una red son más frecuentes.

2.3. Sistema de Colas

Un sistema de colas se puede describir como un conjunto de clientes que llega a un sistema buscando un servicio, esperan si este no es inmediato, y abandonan el sistema una vez han sido atendidos. En algunos casos se puede admitir que los clientes abandonen el sistema si se cansan de esperar. El término cliente se usa con un sentido general y no implica que sea un ser humano, puede significar piezas esperando su turno para ser procesadas o una lista de trabajo esperando para imprimir en una impresora.



(a) Sistema de cola básico

La teoría de colas fue originariamente un trabajo práctico. La primera aplicación de la que se tiene conocimiento es del matemático danés Erlang sobre conversaciones telefónicas en 1909, para el cálculo de tamaño de centralitas. Después se convirtió en un concepto teórico que consiguió un gran desarrollo, y desde hace unos años se vuelve a hablar de un concepto aplicado aunque exige un importante trabajo de análisis para convertir las fórmulas en realidades, o viceversa.

2.4. Características

Las características básicas que se deben utilizar para describir adecuadamente un sistema de colas son seis:

- Patrón de llegada de los clientes
- Patrón de servicio de los servidores
- Disciplina de cola
- Capacidad del sistema
- Número de canales de servicio
- Número de etapas de servicio

Algunos autores incluyen una séptima característica que es la población de posibles clientes.

2.4.1. Patrón de llegada de los clientes

En situaciones de cola habituales, la llegada es estocástica[1], es decir la llegada depende de una cierta variable aleatoria, en este caso es necesario conocer la distribución probabilística entre dos llegadas de cliente sucesivas. Además habría que tener en cuenta si los clientes llegan independiente o simultáneamente. En este segundo caso (es decir, si llegan lotes) habría que definir la distribución probabilística de éstos. También es posible que los clientes sean "impacientes", es decir, que lleguen a la cola y si es demasiado larga se vayan, o que tras esperar mucho tiempo en la cola decidan abandonar. Por último es posible que el patrón de llegada varíe con el tiempo. Si se mantiene constante le llamamos estacionario, si por ejemplo varía con las horas del día es no-estacionario.

Este lapso de tiempo entre llegadas es importante porque cuanto menor sea el intervalo de tiempo, con más frecuencia llegan los clientes, lo cual aumenta la demanda de servidores disponibles. Existen dos clases básicas de tiempos entre llegadas:

- Determinístico, en el cual clientes sucesivos llegan en un mismo intervalo de tiempo, fijo y conocido. Ejemplo: una línea de ensamblaje donde los artículos llegan a una estación en intervalos invariables de tiempo.
- Probabilístico, en el cual el tiempo entre llegadas sucesivas es incierto y variable. Los tiempos entre llegadas probabilísticos se describen mediante una distribución de probabilidad. La distribución exponencial ha probado ser confiable en muchos problemas prácticos.

2.4.2. Patrón de servicio de los servidores

Este proceso define cómo son atendidos los clientes. En algunos casos puede existir más de una estación en el sistema en la cual se proporcione el servicio requerido. A esta estructura se la conoce como sistemas de colas de canal múltiple. En dichos sistemas, los servidores pueden ser idénticos, en el sentido de que proporcionan la misma clase de servicio con igual rapidez, o pueden ser no idénticos. Al contrario de un sistema de canal múltiple, existen los sistemas de colas de canal sencillo, como por ejemplo un proceso de producción con una estación de trabajo que proporciona el servicio requerido, y todos los productos deben pasar por esta. Otra característica de este proceso es el número de clientes atendidos al mismo tiempo en una estación. En los bancos y supermercados, solamente un cliente es atendido a la vez. Por el contrario, los pasajeros que esperan en una parada de colectivo son atendidos en grupo. Por último, otra característica del proceso de servicio es si se permite o no la prioridad, esto es ¿puede un servidor detener el proceso con el cliente que está atendiendo para dar lugar a un cliente que acaba de llegar? Por ejemplo, en una sala de urgencias, la prioridad se presenta cuando un médico, que está atendiendo un caso que no es crítico es llamado a atender un caso más crítico. Cualquiera sea el proceso de servicio, es necesario tener una idea de cuánto tiempo se requiere para llevar a cabo el servicio. Esta cantidad es importante debido a que cuanto más dure el servicio, más tendrán que esperar los clientes que llegan. Como en el caso del proceso de llegada, este tiempo puede ser determinístico o probabilístico. Con un tiempo de servicio determinístico, cada cliente requiere precisamente la misma cantidad conocida de tiempo

[1] Concepto matemático que sirve para usar magnitudes aleatorias que varían con el tiempo o para caracterizar una sucesión de variables aleatorias (estocásticas) que evolucionan en función de otra variable, generalmente el tiempo.

para ser atendido. Con un tiempo de servicio probabilístico, cada cliente requiere una cantidad distinta e incierta de tiempo de servicio. Los tiempos de servicio probabilísticos se describen matemáticamente mediante una distribución de probabilidad. La distribución exponencial ha sido confiable en muchas aplicaciones.

Como se puede ver, las variaciones de los elementos de un caso de colas dan lugar a diversos modelos de colas.

2.4.3. Disciplina de cola

La disciplina de cola es la manera en que los clientes se ordenan en el momento de ser servidos de entre los de la cola. Cuando se piensa en colas se admite que la disciplina de cola normal es FIFO (atender primero a quien llegó primero), esta disciplina es la que se utilizará en este trabajo. Sin embargo en muchas colas es habitual el uso de la disciplina LIFO (atender primero al último). También es posible encontrar reglas de secuencia con prioridades, como por ejemplo secuenciar primero las tareas con menor duración o según tipos de clientes. En cualquier caso dos son las situaciones generales en las que trabajar. En la primera, llamada en inglés "preemptive", si un cliente llega a la cola con una orden de prioridad superior al cliente que está siendo atendido, este se retira dando paso al más importante. Dos nuevos subcasos aparecen: el cliente retirado ha de volver a empezar, o el cliente retorna donde se había quedado. La segunda situación es la denominada "no-preemptive" donde el cliente con mayor prioridad espera a que acabe el que está siendo atendido. Este último tipo de cola se puede ver en servicios donde hay diferentes tipos de clientes, como por ejemplo en los servidores de videojuegos, donde un cliente "premium"tiene más prioridad a la hora de entrar a un servidor que un usuario normal.

2.4.4. Capacidad del sistema

En algunos sistemas existe una limitación respecto al número de clientes que pueden esperar en la cola. A estos casos se les denomina situaciones de cola finitas. Esta limitación puede ser considerada como una simplificación en la modelización de la impaciencia de los clientes.

2.4.5. Número de canales de servicio

Es evidente que es preferible utilizar sistemas multiservidor con una única línea de espera para todos los servidores a una cola por servidor. Por tanto, cuando se habla de canales de servicio paralelos, se habla generalmente de una cola que alimenta a varios servidores mientras que el caso de colas independientes se asemeja a múltiples sistemas con sólo un servidor.

2.4.6. Número de etapas de servicio

Un sistema de colas puede contar con una o varias etapas de servicio. En los sistemas con múltiples etapas el cliente puede pasar por un número de etapas mayor que uno. Una peluquería es un sistema con una sola etapa, salvo que haya diferentes servicios (manicura, maquillaje) y cada uno de estos servicios sea desarrollado por un servidor diferente. En algunos sistemas se puede admitir la vuelta atrás o "reciclado", esto es habitual en sistemas productivos como controles de calidad.

2.5. Notación de Kendall

La notación de Kendall es el sistema estándar utilizado para describir y clasificar un sistema de colas. David George Kendall propuso en 1953 describir modelos de colas utilizando tres factores y desde entonces se ha extendido a la siguiente notación:

$$A/B/C/D/E/F$$

Donde:

- A: Distribución de llegada.
- B: Distribución de servicio.

A y B pueden ser M, D o G:

1. M: Distribución de Markov. La tasa de arribos es una variable de Poisson, el tiempo entre arribos es Exponencial.
2. D: Distribución determinística (un valor fijo).
3. G: General, es decir cualquier distribución de probabilidad, menos Poisson o Exponencial.

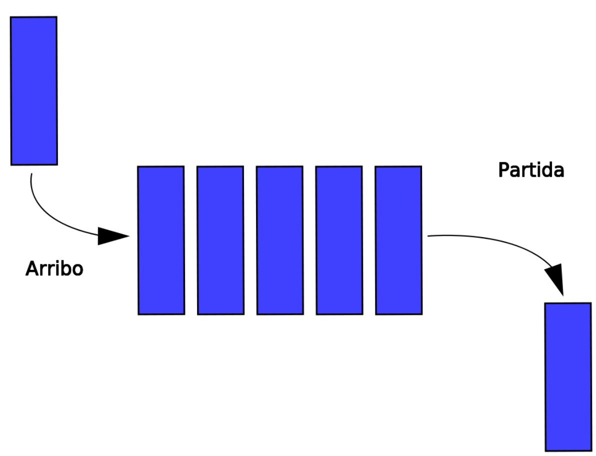
- C: Entero positivo que denota el número servidores en paralelo.
- D: Cantidad máxima de clientes permitidos en el sistema. Si esta capacidad es superada se rechaza el arribo de un nuevo cliente.
- E: Política de atención de la cola
  - FIFO (first in first out).
  - LIFO (last in first out).
  - SIRO (service in random order).
  - Prioridad.
- F: Tamaño de la población que ingresa al sistema. Infinita o un valor numérico.

3. Marco Teórico - Simulación

3.1. Simulación de eventos discretos

Este tipo de simulación se centra en modelar sistemas donde las variables cambian de estado instantáneamente con el paso del tiempo. En un instante de tiempo ocurre un evento donde se puede cambiar el estado del sistema (que ocurra un evento no implica que haya un cambio de estado).
Este tipo de sistemas pueden ser resueltos manualmente, pero debido a la gran cantidad de información que debe ser guardada y procesada en el mundo real, este cálculo se realiza con computadoras.

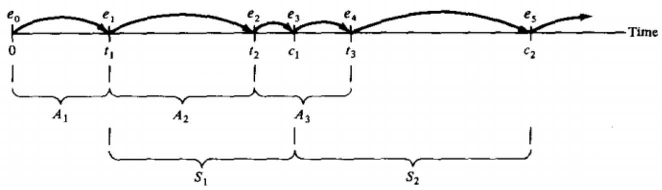
En el caso a desarrollar en este trabajo se encuentran dos tipos de eventos, la llegada de un cliente nuevo y el fin del servicio de uno, que deriva en la partida del mismo. Un arribo es un evento porque hace que la variable de estado del servidor cambie de desocupado a ocupado, o que aumente el tamaño de la cola de espera. Así mismo, una partida es un evento en sí porque cambia el estado del servidor o en su defecto, disminuye el tamaño de la cola de espera.



(b) Eventos en un sistema de colas

3.1.1. Mecanismos de avance del tiempo

Debido a la naturaleza de los modelos de simulación de eventos discretos, se debe hacer un seguimiento del valor actual del tiempo simulado mientras la simulación procede, también se necesita un mecanismo para avanzar el tiempo simulado de un valor a otro. Llamamos a la variable que da el valor actual del tiempo simulado, reloj de simulación.Se sugieren dos métodos para avanzar el reloj de simulación: avance al próximo evento y avance de tiempo a incremento fijo. Con el avance de tiempo al próximo evento, el reloj de simulación es inicializado en cero y los tiempos de ocurrencia de eventos futuro son determinados. El reloj de simulación avanza al tiempo de ocurrencia del evento futuro más inminente, el estado del sistema se actualiza por el hecho de que tal evento ha ocurrido, y nuestro conocimiento de los tiempos de ocurrencia de eventos futuros es actualizado. Luego, el reloj de simulación avanza al tiempo de evento más inminente, el estado del sistema es actualizado y determinamos el tiempo de eventos futuros.



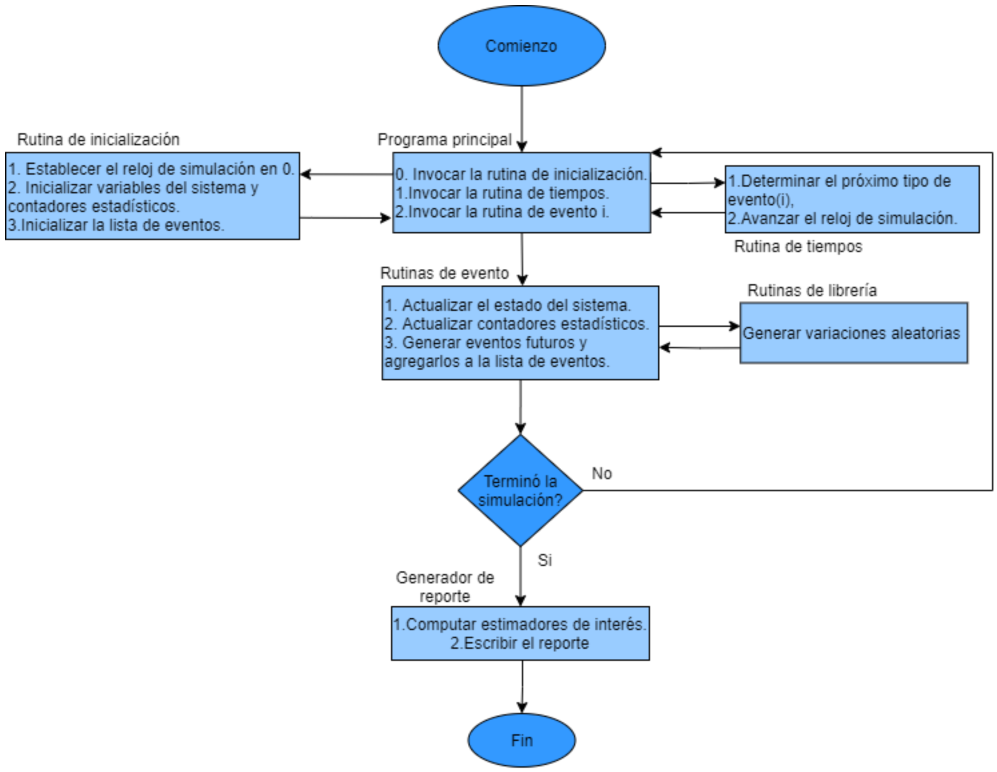
(c) Duración de las diferentes mediciones entre eventos

3.1.2. Componentes y organización de un modelo de simulación de eventos discretos

- Estado del sistema: Es la colección de variables de estado necesarias para describir al sistema en un tiempo simulado
- Reloj de simulación: Variable que da el valor actual del tiempo simulado.
- Lista de eventos: Lista que contiene el próximo tiempo en que cada tipo de evento ocurrirá.
- Contadores estadísticos: Variables utilizadas para almacenar información estadística sobre el desempeño del sistema.
- Rutina de inicialización: Es un subprograma que inicializa el modelo de simulación al tiempo cero.
- Rutina de tiempo: Subprograma que determina el próximo evento de la lista de eventos y luego se actualiza el reloj de simulación al tiempo donde este evento ocurrirá.
- Rutina de eventos: Subprograma que actualiza el estado del sistema cuando un tipo de evento particular ocurre.
- Rutina de librería: Conjunto de subprogramas utilizados para generar observaciones aleatorias de distribuciones de probabilidad que fueron determinadas como parte del modelo de simulación.
- Generador de reportes: Es un subprograma que computa estimadores de las medidas de desempeño deseadas y produce un reporte cuando la simulación termina.
- Programa principal: Subprograma que invoca la rutina de tiempo para determinar el próximo evento y luego transfiere el control a la rutina de evento correspondiente para actualizar apropiadamente el estado del sistema. El programa principal podría también comprobar la terminación e invocar al generador de reportes cuando la simulación finalice.

3.1.3. Diagrama de flujo del mecanismo de avance al próximo evento

A fin de abstraer el algoritmo de cualquier lenguaje de programación se presenta en el formato pseudo-código una alternativa de flujo de procesos para lograr obtener resultados para su futuro análisis.

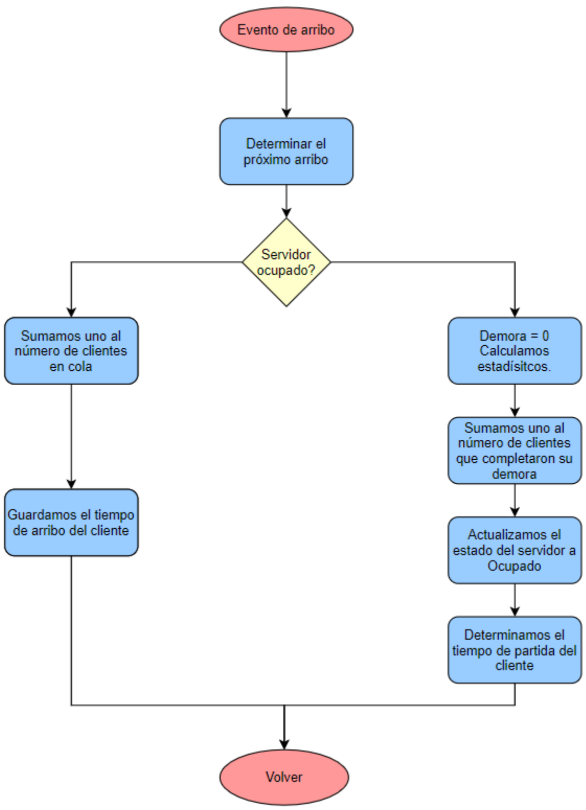


(d) Diagrama de flujo de arribo

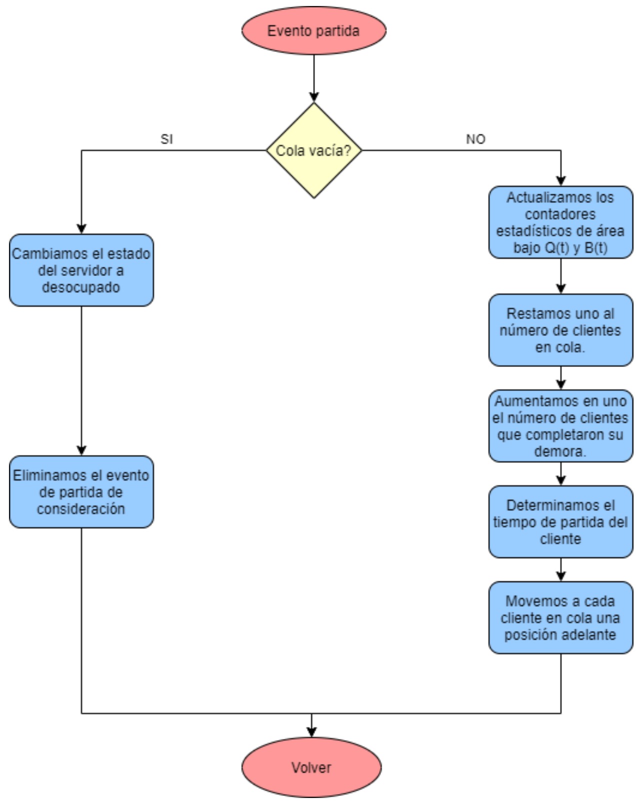
Desde el programa principal se invocan los i eventos que en el caso de los sistemas de colas son los presentados en la siguiente sub-sección.

3.1.4. Diagramas de flujo de eventos de arribo y partida

Los eventos de arribos y las partidas son parte fundamental de los sistemas de cola, por lo que las rutinas que modelan a los mismos deben ser diseñadas cuidadosamente para no omitir ningún detalle y la simulación pueda ser realizada satisfactoriamente y finalmente poder obtener resultados en los cuáles poder basarse a la hora de tomar una decisión informada. En los siguientes gráficos se observará la secuencia de operaciones que se realizarán en cada uno de estas rutinas.



(e) Diagrama de flujo de arribo



(f) Diagrama de flujo de partida

4. Marco Teórico - Simulación Aplicada a Colas de Espera

4.1. Sistema M/M/1

Uno de los sistemas de espera mas sencillos es el presentado en este apartado. Consiste en un proceso estocástico cuyo espacio de estado (del sistema) es el conjunto 0,1,2,3, ... donde el valor corresponde al número de clientes en el sistema, incluidos los que están actualmente en servicio. Cuenta con un único servidor que sirve a los clientes uno a la vez desde el frente de la cola. Cuando se completa el servicio, el cliente deja la cola y el número de clientes en el sistema se reduce en uno. El espacio de espera es de tamaño infinito, por lo que no hay límite en la cantidad de clientes que puede contener. La población de clientes es el conjunto de todos los posibles clientes del sistema y para problemas como este, donde el número de clientes potenciales es bastante grande, el tamaño de la población se considera como si fuera infinito. El proceso de espera del estudio es mediante una sola línea ya que los clientes esperan ser atendidos por un único servidor.

Este sistema, junto con una variación del mismo será el objetivo a tratar en este trabajo de manera práctica.

4.2. Simulación de un sistema de colas con un único servidor

4.2.1. Medidas de rendimiento

Para medir el rendimiento del sistema en cuestión se observarán las estimaciones de tres cantidades:

- $\hat{d}$(n) Promedio esperado de demora en cola de los n clientes que completaron su demora.

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

  Dónde:
  D_i es la demora del cliente i.

- $\hat{q}$(n): Promedio esperado de clientes en cola (basados en los observados en la simulación).

$$\hat{u}(n) = \frac{\int_0^{T(n)} Q(t) * dt}{T(n)} \tag{2}$$

  Dónde:
  T(n) es el tiempo requerido para observar nuestras n demoras en cola.
  Q(t) la función que determina la longitud de la cola en un momento t.

- $\hat{u}(n)$: Porcentaje de utilización del servidor:

$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t) * dt}{T(n)} \tag{3}$$

  Dónde:
  T(n) es el tiempo requerido para observar nuestras n demoras en cola.
  B(t) es igual a 1 si el servidor esta ocupado en el momento t e igual a cero si el mismo está desocupado en ese momento.

Estos estimadores se utilizarán a lo largo de la simulación para calcular las medidas de rendimiento antes mencionada. Sin embargo, si se quiere obtener un resultado exacto de estos valores se debe realizar un cálculo analítico.

Esta forma de resolver situaciones analíticamente no siempre es viable pero en el caso presente se puede realizar a partir de las siguientes fórmulas:

- Promedio de clientes en el sistema.

$$L = \lambda * W \tag{4}$$

- Promedio de clientes en cola.

$$L_q = \frac{p^2}{1-p} \tag{5}$$

- Tiempo promedio en sistema.

$$W = W_q + \frac{1}{\mu} \tag{6}$$

- Tiempo promedio en cola.

$$W_q = \frac{L_q}{\lambda} \tag{7}$$

- Utilización del servidor.

$$U = p = \frac{\lambda}{\mu} \tag{8}$$

- Probabilidad de n clientes en cola (Probabilidad de n+1 clientes en el sistema).

$$P_{n+1} = p^{n+1} * P_0 \tag{9}$$

  Donde $P_0 = 1 - p$

- Probabilidad de denegación de servicio con n clientes máximos por cola.

$$P_d = 1 - (P_{n+1} + ... + P_0) \tag{10}$$

5. Desarrollo práctico

Aquí se analizará el código a reproducir de la bibliografía en Python 3.X y los distintos elementos introducidos que harán posible la simulación del sistema de colas antes descripto. Se busca mostrar:

- La validez de la simulación a eventos discretos, mediante la simulación del sistema que ha sido estudiado (MM1) y para el cuál es posible obtener las medidas de rendimiento de forma analítica.
- Las medidas de rendimiento obtenidas al cumplir la condición de finalización.
- La evolución de las medidas de rendimiento en el sistema simulado a través del tiempo.

Las medidas de rendimiento de la simulación se comparan con los resultados obtenidos utilizando el método analítico para comprobar la validez de los mismos.

5.1. Modificaciones en el código base

En el proceso de reescribir el código de un lenguaje a otro no se pudo evitar tener que modificar algunos métodos y operadores lógicos para que el proceso siga la misma lógica que el código original a pesar de no contar con la misma estructura debido a las limitaciones de un lenguaje. Para poder distinguir que variables, constantes o métodos fueron agregados para cumplir las consignas de este trabajo los elementos de la bibliografía fueron escritos en inglés con los mismos nombres del código Pascal, mientras que los elementos agregados fueron escritos en español. El nuevo código consta de los siguientes componentes que a continuación se desarrollan.

5.1.1. Archivo con métodos útiles

En este archivo se encuentra el método $funExpon()$ que lo que hará será retornar un número aleatorio en base a la media enviada como parámetro siguiendo una distribución exponencial. Esta forma de generar números con distribución exponencial fue desarrollada en el trabajo 2.2.

5.1.2. Ejecución principal

Esta parte del código fue agregada ya que reemplaza a la ejecución principal de la bibliografía que fue colocada dentro de un método. Esta modificación se realizó para poder realizar la simulación una n cantidad de veces (este valor es una variable dentro del código). Esta repetición del experimento permitirá luego obtener las medidas de rendimiento de cada simulación para juntarlas y graficar o analizar como se comportan los estadísticos cuando la cantidad de repeticiones es muy grande. Se busca que los valores generados tiendan a los valores teóricos que fueron calculados previamente en el mismo código y también con un simulador en la web.

5.1.3. Método ExecuteSimulation()

Este nuevo método reemplazaría al programa principal del código de la bibliografía. Básicamente llama a los distintos métodos necesarios para completar un evento del sistema hasta devolver los datos de esa simulación en particular.

5.1.4. Método UpdateTimeAvgStats()

No fue modificado, modifica las variables globales que calculan el $Q(t)$ y $B(t)$

5.1.5. Método Depart()

Fue trasladado tal cual el código original salvo modificaciones propias de las diferencias entre lenguajes.

5.1.6. Método Arrive()

Se quitó la limitación del largo de la cola ya que no se quiere cortar la ejecución del programa. El resto del método respeta el original.

5.1.7. Método Timing()

Se encarga de setear que tipo de evento será el próximo, si un arribo o una partida. Se modificó una condición y el mensaje de error para que sea mas prolijo, pero esto no modifica la lógica.

5.1.8. Método Initialize()

En la bibliografía se utiliza para setear algunas variables en cero, sin embargo, debido a que ahora se realizan varias simulaciones se agregaron las inicializaciones de variables que se encontraban en la definición de las mismas con el objetivo de que se reinicien al comienzo de cada simulación y no solo al comienzo del programa principal.

5.1.9. Método Report()

Los valores de rendimientos generados por este método no fueron modificados, solo que en lugar de mostrarse en consola, son devueltos en un arreglo para sumarse a los arreglos acumulativos de estos valores para luego realizar los cálculos y las gráficas.

5.2. Resultados del simulador

Una vez explicado el flujo del algoritmo y en que consta el programa realizado se procede a presentar los resultados tanto analíticos como gráficos. Se utilizó un n de 10 repeticiones y un total de clientes servidos de 10000.

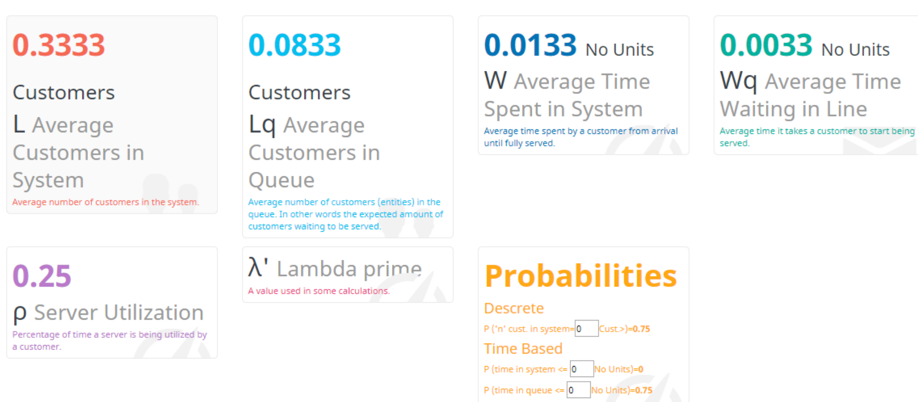
5.2.1. Tasas de arribo: 25 % con respecto a la tasa de servicio.

Para esa simulación se introdujeron como parámetros un $\lambda = 25$ y un $\mu = 100$.

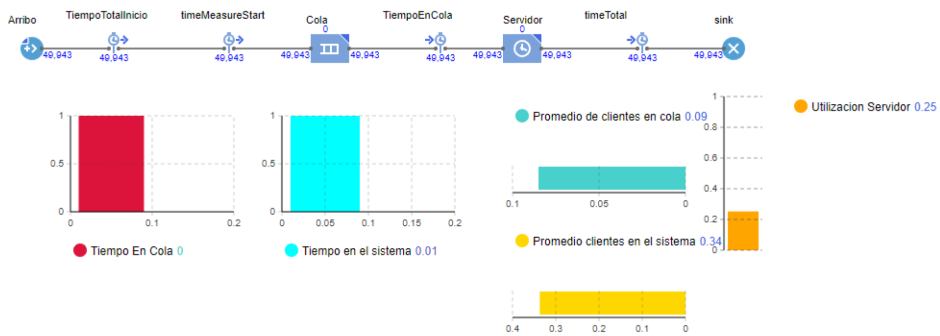
los valores teóricos calculados en la ejecución principal del programa y luego graficados con lineas punteadas resultaron los siguientes:

- Promedio de clientes en la cola de espera = 0.083
- Promedio de tiempo de demora en la cola = 0.003
- Promedio de tiempo de demora en sistema = 0.013
- Promedio de utilización del servidor = 0.25
- Promedio de clientes en el sistema = 0.333

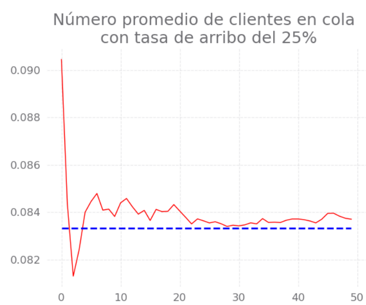
También se presentan estos mismos cálculos realizados con la calculadora de teoría de colas de internet.

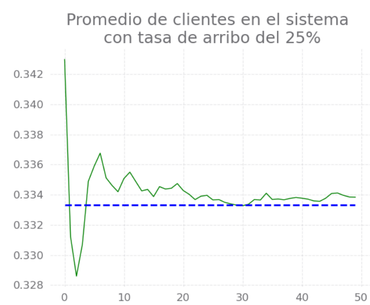


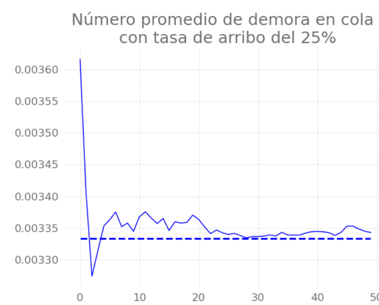
El resultado de la simulación en Anylogic fue el siguiente

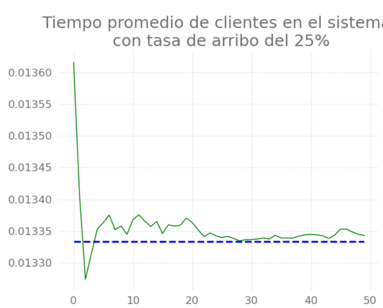


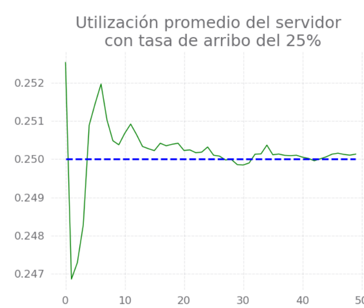
Finalmente, las gráficas generadas con python:



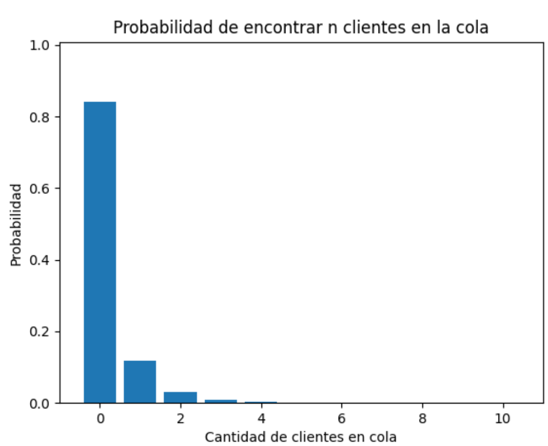








Además se agrega la gráfica de las probabilidades de encontrar una cierta cantidad n de clientes en la cola:



Como se aprecia, a partir de $n = 5$ la probabilidad de encontrar personas en cola ya es prácticamente nula. Esto se debe a la baja cantidad de arribos comparada con los cortos tiempos de servicio.

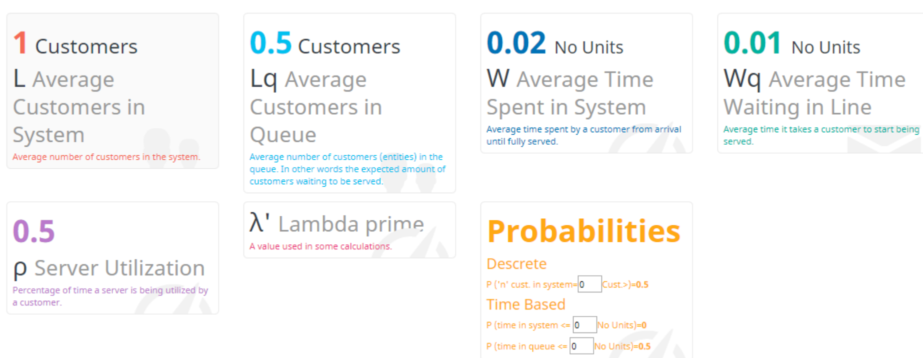
5.2.2. Tasas de arribo: 50 % con respecto a la tasa de servicio.

Para esa simulación se introdujeron como parámetros un $\lambda = 50$ y un $\mu = 100$.

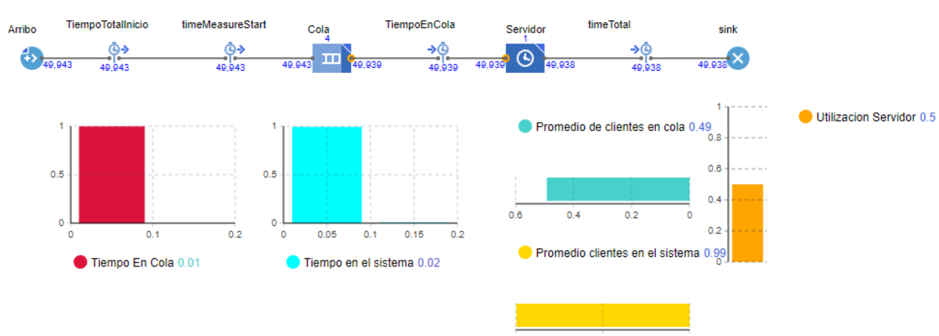
los valores teóricos calculados en la ejecución principal del programa y luego graficados con lineas punteadas resultaron los siguientes:

- Promedio de clientes en la cola de espera = 0.5
- Promedio de tiempo de demora en la cola = 0.01
- Promedio de tiempo de demora en sistema = 0.02
- Promedio de utilización del servidor = 0.5
- Promedio de clientes en el sistema = 1

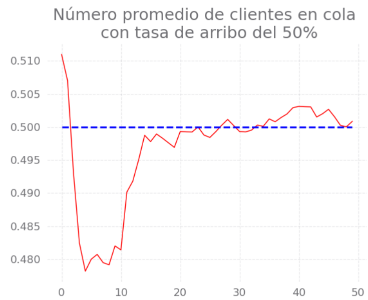
También se presentan estos mismos cálculos realizados con la calculadora de teoría de colas de internet.

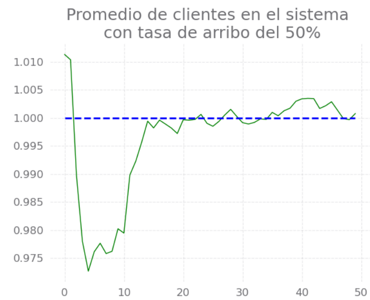


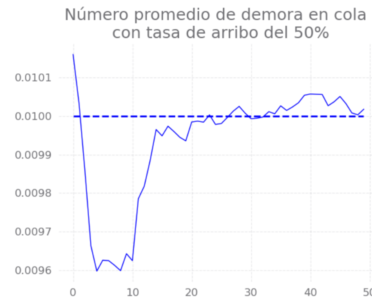
El resultado de la simulación en Anylogic fue el siguiente

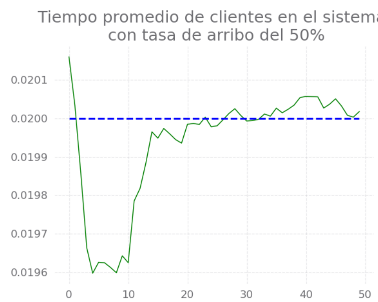


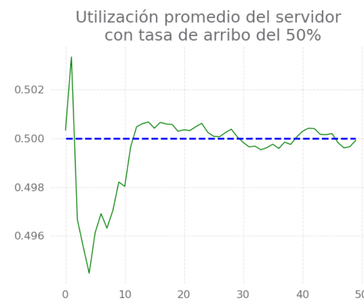
Finalmente, las gráficas generadas con python:



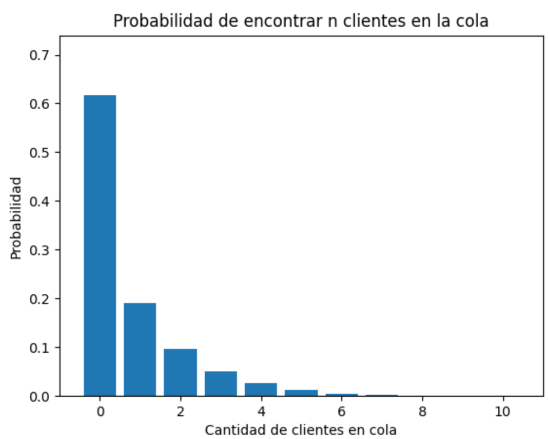








Al igual que en el caso anterior se agrega la gráfica de las probabilidades de encontrar una cierta cantidad n de clientes en la cola:



En este caso también se observa una probabilidad de 0 a partir de un valor, en este caso $n = 8$. Esto incrementa ya que al utilizarse mas el servidor las colas comienzan a hacerse mas largas.

5.2.3. Tasas de arribo: 75 % con respecto a la tasa de servicio.

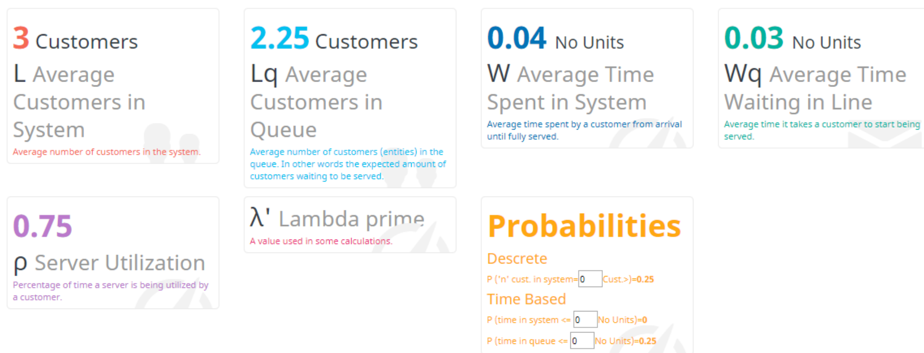
Para esa simulación se introdujeron como parámetros un $\lambda = 75$ y un $\mu = 100$.

los valores teóricos calculados en la ejecución principal del programa y luego graficados con lineas punteadas resultaron los siguientes:

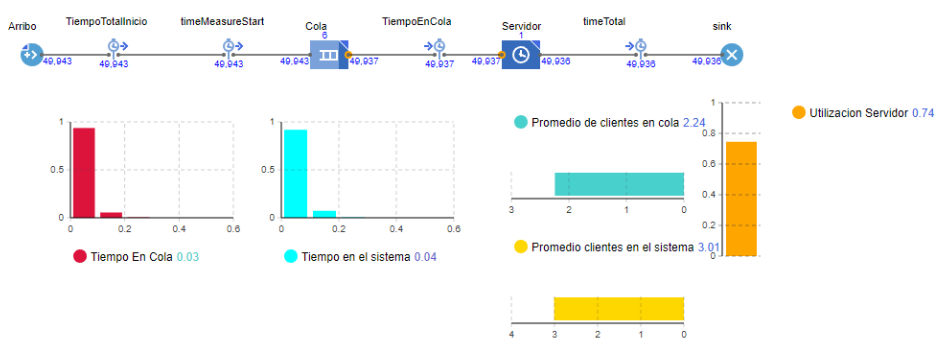
- Promedio de clientes en la cola de espera = 2.25
- Promedio de tiempo de demora en la cola = 0.03
- Promedio de tiempo de demora en sistema = 0.04

- Promedio de utilización del servidor = 0.75
- Promedio de clientes en el sistema = 3

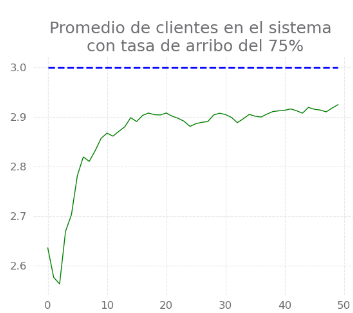
También se presentan estos mismos cálculos realizados con la calculadora de teoría de colas de internet.

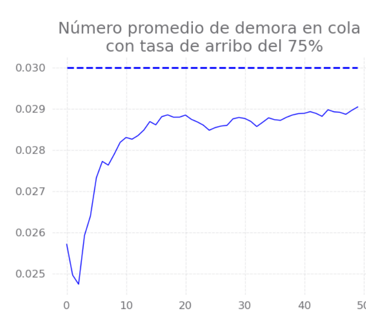


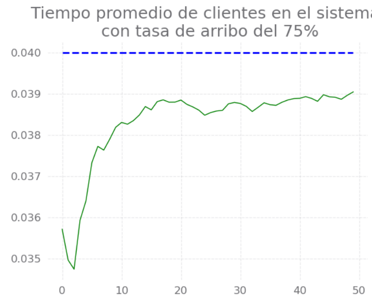
El resultado de la simulación en Anylogic fue el siguiente

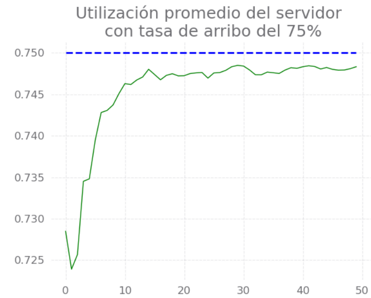


Finalmente, las gráficas generadas con python:

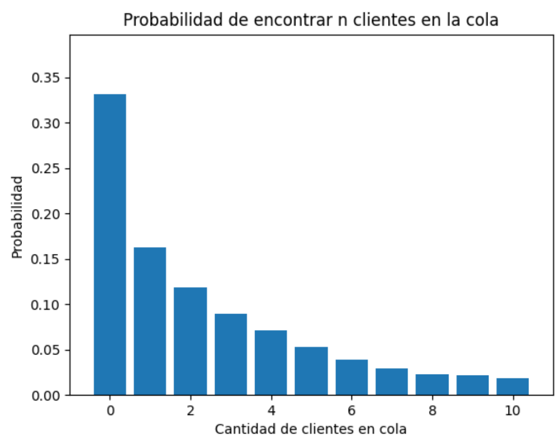








La gráfica de las probabilidades de encontrar una cierta cantidad n de clientes en la cola siguiendo los mismos límites que en los casos anteriores:



Con esta relación entre arribos y servicios el n necesario para que la probabilidad sea nula excede los calculados. En un caso práctico, si se cuenta con poco espacio de espera, por ejemplo 8 sillas, es muy probable que los clientes tengan que esperar parados.

5.3. Resultados de un M/M/1/K

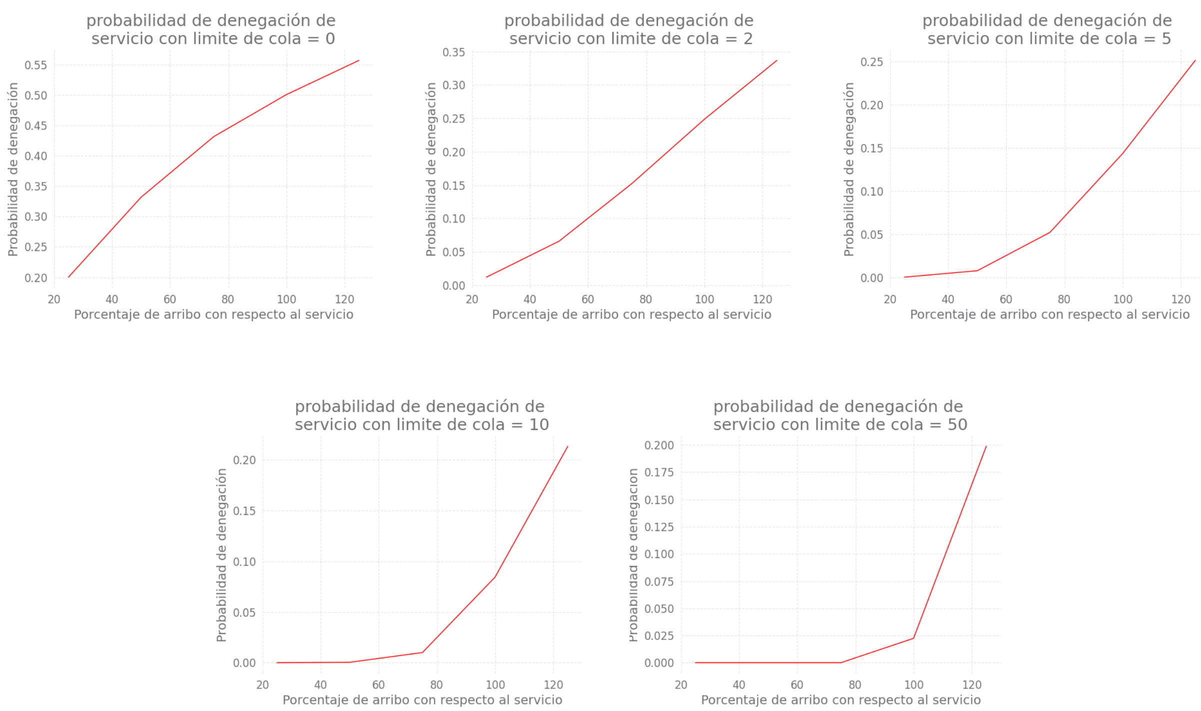
Este sistema es igual al M/M/1 con la diferencia de que la cola es finita. resumiendo sus características se pueden mencionar:

- Se tiene un sistema de llegadas que se producen según un proceso de Poisson de razón λ, donde los tiempos entre llegadas estarán distribuidos exponencialmente.
- Los tiempos de servicios están distribuidos de manera exponencial.
- Se posee un único servidor en el sistema.
- La capacidad del sistema es finita, ésta se expresa por la constante K.
- La disciplina del sistema será FIFO
- Se cuenta con una sola cola.

Se analizó un sistema con cola finita de tamaño: 0, 2, 5, 10, 50 y tasas de arribo: 25 %, 50 %, 75 %, 100 %, 125 % con respecto a la tasa de servicio.

Se realizaron 10 corridas y se calculó la media de los resultados de las mismas.

Tasa de arribo respecto al servicio	Tamaño máximo de cola	Probabilidad de denegación de servicio
25.0 %	0	0.1991
25.0 %	2	0.0125
25.0 %	5	0.0002
25.0 %	10	0.0
25.0 %	50	0.0
50 %	0	0.3348
50 %	2	0.0682
50 %	5	0.0083
50 %	10	0.0002
50 %	50	0.0
75.0 %	0	0.4286
75.0 %	2	0.1575
75.0 %	5	0.053
75.0 %	10	0.0118
75.0 %	50	0.0
100 %	0	0.5006
100 %	2	0.2498
100 %	5	0.1427
100 %	10	0.0846
100 %	50	0.0189
125.0 %	0	0.5542
125.0 %	2	0.3385
125.0 %	5	0.2504
125.0 %	10	0.2141
125.0 %	50	0.1936



En la tabla y gráficos anteriores se ve como, a medida que se "satura.el servidor aumentando la tasa de arribo con respecto a la tasa de servicio la probabilidad de que el servicio sea denegado aumenta. Un dato relevante es que cuando el límite de la cola es 0 la gráfica es cóncava hacia abajo y luego se va invirtiendo hasta llegar a tamaño igual a 5 que pasa a ser cóncava hacia arriba. Claramente el mayor valor de probabilidad de denegación se va a dar cuando la tasa de arribo sea mas grande que la tasa de servicio y no haya cola. Esta probabilidad nunca va a ser igual a 1 ya que siempre se va a atender por lo menos a un cliente de todos los que llegan.

6. Conclusiones

Al realizar la experiencia reiteradas veces nos percatamos que cuenta que la aleatoriedad juega un papel conflictivo para los resultados, ya que , a pesar de que los estadísticos simulados en cada experiencia resultan ser próximos a los resultados obtenidos por estadísticos analíticos, es necesario tener un numero de usuarios muy elevado para que estos se aproximen con mayor precisión.

Se puede observar las variación dichas gráficas, que con el paso del tiempo y los usuarios que utilizan el sistema, busca aproximarse al valor que se obtiene por métodos analíticos. Cabe resaltar que cada muestra difiera un poco de los resultados reales, es ahí donde juega la aleatoriedad y no es posible evitarlo, sólo podemos observarlo y estudiarlo.

Este trabajo fue de gran provecho, no solo se pudo profundizar en el estudio de la teoría de colas, sino también en la utilización de la herramienta Anylogic, demostrando su facilidad de uso y las ventajas tanto analíticas como gráficas a la hora de simular este tipo de eventos. Las comparaciones de los estadísticos de las diferentes fuentes (simulador online, cálculos teóricos, Anylogic) fueron positivas, ya que los resultados fueron correspondidos por los valores generados por el simulador basado en el código presentado por la bibliografía.

Referencias

[1] Song Chew Continuous-Service M/M/1 Queuing Systems *Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL*, 2019.

[2] José Pedro García Sabater Aplicando teoría de colas en Dirección de operaciones *Universidad Politécnica de Valencia*, 2016.

[3] https://www.supositorio.com/ Calculadora de teoría de colas.