PROGRAMA PARA LA SIMULACIÓN DE UNA COLA M/M/1

Jorge Aching Samatelo,

jorge_aching@hotmail.com

Egresado de la Facultad de Ingeniería Electrónica de la UNMSM,
Lima, Perú

Resumen: Este trabajo se enmarca dentro de la Teoría de Colas y en particular trata del estudio del modelo M/M/1 y la simulación implementada mediante un programa que permita observar el comportamiento estocástico de este modelo para el tráfico de paquetes en una red, donde el comportamiento estadístico vendrá dado por los valores promedios de sus indicadores. Para la implementación del simulador se utilizó los conceptos relacionados con la programación orientada a objetos [Luis Joyanes, 1998], el cual permite implementar de forma natural sistemas con alta complejidad. Además, una parte importante del programa vendrá dado por la utilización de una estructura de datos tipo cola, el cual nos permite establecer la política de atención y recepción de los paquetes (el primero que llega, primero en ser atendido). El trabajo culmina con la exposición y explicación detallada de las variables que definen el comportamiento estadístico de la cola.

Abstract : This work has as a goal to carry out a study of a tail M/M/1 and the implementation of a program that allow the behavior simulation; since, this is an statistic system the study of its statistic behavior is done by the mean of its indicators. To implementation of simulator it use related concepts to programming orientated to objects, since as, this allow implement, naturally system with high complexity. Besides, an important point of programming is done using a structure of data column type, which allows us establish a politic of packet attention and receiving (First In First Out), We end with the exposition and explanation detailed of variables which explain the statistic of the tail.

Palabras Claves: Teoría de colas, estocástico, *array*, función de densidad de probabilidad, distribución de Poisson, variable aleatoria, simulación.

I. INTRODUCCIÓN

Una de las herramientas matemáticas más poderosas para realizar el análisis cuantitativos de las redes de computadoras es la Teoría de Colas. El creador de la Teoría de Colas fue el matemático danés A. K. Erlang por el año 1909. Ha tenido un fuerte auge por su utilidad en la construcción de modelos de comportamiento estocástico de gran número de fenómenos, tanto naturales como creados por el hombre. Esta técnica se desarrolló en un inicio para analizar el comportamiento estadístico de los sistemas de conmutación telefónica, sin

embargo, desde entonces, también ha sido aplicada para resolver muchos problemas de redes. [Mischa Schwartz, 1994]

II. SISTEMA DE COLAS

Se pueden utilizar sistemas de colas para modelar procesos en los cuales los clientes van llegando, esperan su turno para recibir el servicio, reciben el servicio y luego se marchan. Los sistemas de colas de espera pueden definirse mediante cinco componentes:

1) La función de densidad de probabilidad del tiempo entre llegadas.
2) La función de densidad probabilidad del tiempo de atención.
3) El número de servidores.
4) La política de atención en las colas.
5) El tamaño máximo de las colas.

La densidad de probabilidad del tiempo entre llegadas, describe el intervalo de tiempo entre llegadas consecutivas. Así por ejemplo, si se registrara la llegada de clientes al cajero de un supermercado. A cada llegada, se registrará el tiempo transcurrido desde que ocurrió la llegada anterior. Después de un tiempo suficientemente largo de estar registrando las muestras, la lista de números podría clasificarse y agruparse: es decir, tantos tiempos entre llegadas de 0.1 segundos, de 0.2 seg. , etc. Esta densidad de probabilidad caracteriza el proceso de llegadas.

Cada cliente requiere de cierta cantidad de tiempo proporcionado por el cajero. El tiempo de atención requerido varia entre un cliente y otro (por ejemplo, un cliente puede presentar un carro lleno de artículos, y el siguiente puede traer únicamente una caja de galletas). Para analizar un sistema de colas de espera, deben conocerse tanto la función de densidad de probabilidad del tiempo de atención, como la función de densidad del tiempo entre llegadas.

La cantidad de cajeros no necesita explicarse. Muchos bancos, por ejemplo, tienen una sola cola larga para varios cajeros y cada vez que se libera uno, el cliente que se encuentra en la cola se dirige a dicha caja; a este sistema se le denomina sistema de cola multiservidor. En otros bancos, cada cajero, tiene su propia cola particular, en este caso se tiene un conjunto de colas independientes de un sólo servidor, y no un sistema multiservidor.

La política de atención de una cola describe el orden según el cual los clientes van siendo tomados de la cola de espera. Los supermercados utilizan el método del primero en llegar es el primero en ser servido. En las salas de urgencia de los hospitales se utiliza, a menudo, el criterio de él que esté más grave, no el primero en llegar es el primero en ser atendido. En un entorno amistoso de oficina, ante la fotocopiadora, se atiende primero al que tenga menor volumen de documentos a copiar.

No todos los sistemas de colas de espera poseen una capacidad infinita de recepción de clientes. Así cuando hay un número grande de clientes, pero sólo existe un número finito de lugares en cola de espera, algunos de estos clientes se pierden o son rechazados.

La hipótesis de utilizar una probabilidad de tiempo entre llegadas exponencial es totalmente razonable para cualquier sistema que maneja una gran cantidad de clientes independientes. En semejantes condiciones, la probabilidad de que lleguen exactamente n clientes, durante un intervalo de longitud t, estará dada por la ley de Poisson:

$$p(t) = \frac{(\lambda)^k e^{-\lambda t}}{n!} = \frac{(\lambda t)\, e^{-\lambda t}}{n!} \qquad (2.1)$$

en la cual λ es la velocidad media de llegadas. Aunque la hipótesis de una densidad de probabilidad de tiempo entre llegadas de tipo exponencial es normalmente razonable, en términos generales es más difícil defender la hipótesis de que los tiempos de atención lo sean también. Sin embargo, para las situaciones en las cuales mientras más grande sea el tiempo de atención, menor será su probabilidad de ocurrir, el modelo M/M/1 puede ser una aproximación adecuada.

2.1 Proceso de Poisson

Considérese la secuencia de *m* pequeños intervalos, cada uno de longitud Δt, como muestra la figura 1.

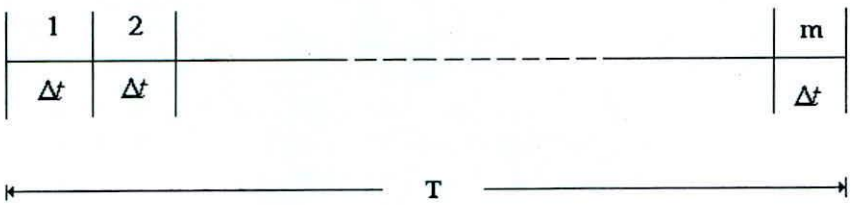


Figura 1. Derivación de la distribución de Poisson

Sea $p = \lambda\Delta t$ la probabilidad de una llegada en cualquier intervalo Δt y $q = 1 - \lambda\Delta t$ la probabilidad de cero llegadas en el intervalo Δt .si asumimos *a p y q* como eventos excluyentes La probabilidad de *k* llegadas en el intervalo $T = m\Delta t$ estará dada por la distribución binomial:

$$p(k) = \binom{m}{k} . p^k . q^{m-k} \tag{2.2}$$

Tomando Δt pequeño, y *T* constante, se tiene:

$$p(k) = \frac{(\lambda T)^k . e^{-\lambda T}}{k!} \qquad k = \{0,1,2,3,.....\} \tag{2.3}$$

Ahora, consideremos un intervalo grande de tiempo y señalemos todos los puntos donde ocurre una llegada; representemos el tiempo entre dos llegadas consecutivas como τ (figura 2), obviamente τ es una variable aleatoria. Para un proceso de Poisson se supondrá que τ tiene una distribución exponencial, es decir:

$$f_\tau(\tau) = \lambda . e^{-\lambda\tau} \qquad \tau \geq 0 \tag{2.4}$$

cuyo valor medio es:

$$E(\tau) = \frac{1}{\lambda} \tag{2.5}$$

mientras que su desviación estándar está dada por:

$$\sigma_\tau^2 = \frac{1}{\lambda^2} \tag{2.6}$$

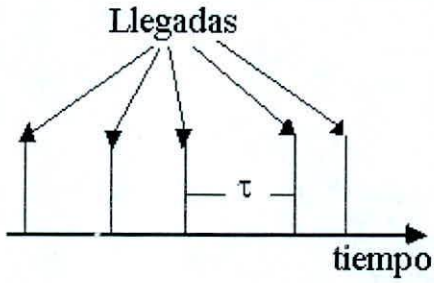


Figura. 2 Llegadas de Poisson

El proceso de atención es similar al proceso de llegada, por lo que la probabilidad de un cumplimiento de atención en el intervalo $(t, t+\Delta t)$ es $\mu\Delta t$ mientras que la probabilidad de no cumplimiento en $(t, t+\Delta t)$ es $(1-\mu\Delta t)$, siguiendo un procedimiento similar al caso de llegadas y considerando que las probabilidades de cumplimiento y no cumplimiento son independientes, se obtendrá que la función de densidad de probabilidad para la variable *r* (figura 3) que define el tiempo entre dos atenciones consecutivas, será:

$$f_r(r) = \mu . e^{-\mu r} \qquad r \geq 0 \tag{2.7}$$

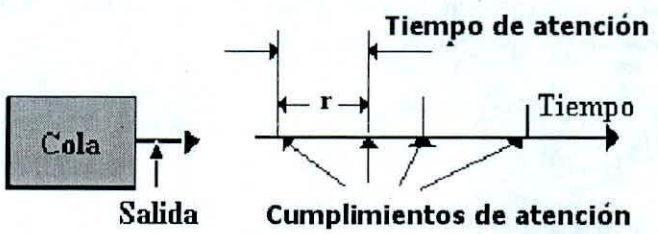


Figura 3. Atenciones de servicio a la salida de una cola

2.2 *Notación Kendall*

Para especificar un tipo de cola se escribe:

Distribución de llegada / Distribución de atención / nº de servidores / capacidad / disciplina /...

En el proceso de llegada puede aparecer:

M: los tiempos entre llegadas siguen una distribución exponencial.
G: los tiempos entre llegadas son variables aleatorias independientes.
D: corresponde a un tiempo entre llegadas determinístico.

De forma análoga se identifican los procesos de atención con *M, G y D.*

Ejemplo: Si se escribe

$$M/D/2/\infty$$

significa que el tiempo entre llegadas es exponencial, el tiempo de atención es determinístico (normalmente vendrá dado por una lista o vector), el número de servidores es 2, la capacidad es infinita y la disciplina es FIFO.

2.3 La cola M/M/1

Es el tipo de cola más sencilla para analizar, se trata de una cola del tipo de servidor único con procesos de llegada de Poisson, estadística de tiempo de atención de distribución exponencial y con política de servicio FIFO (First In First Out) . Las propiedades estadísticas de la cola M/M/1, la ocupación promedio de la cola, la probabilidad de bloqueo para una cola finita, el rendimiento promedio, etc. se determinara con facilidad una vez que se encuentre la probabilidad de estado p_n [1] en la cola. El sistema está operando en estado estacionario de manera que esta probabilidad no varia con el tiempo [Thomas L. Saaty, 1967].

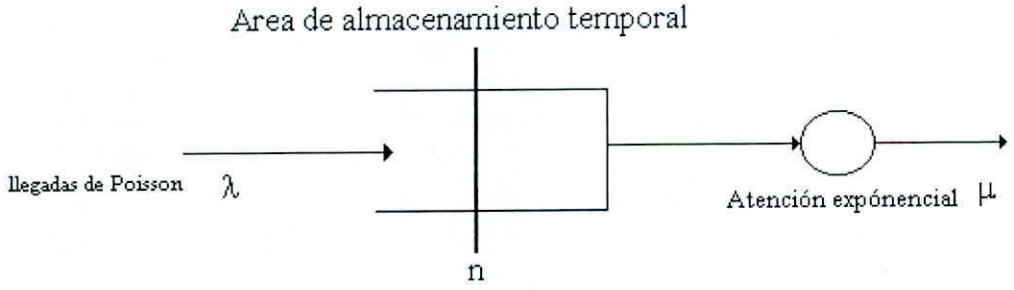


Figura 4 Cola M/M/1

la cola empieza a formarse conforme la tasa de llegadas de paquetes λ se aproxima a la capacidad de transmisión de paquetes μ. Para una área de almacenamiento temporal finita, la cola llegaría a un estado de saturación conforme λ exceda a μ. Cuando el área de almacenamiento temporal se satura, se bloquea la llegada de todos los paquetes (usuarios) siguientes. Si se supone un área de almacenamiento infinita, la cola se vuelve inestable a medida que λ tienda a μ. A través de la simulación se mostrará que para un servidor único $\lambda < \mu$ asegura la estabilidad. En particular, se vera que $\rho = \lambda/\mu$ es un parámetro critico en el análisis de la teoría de formación de colas. Este parámetro suele denominarse utilización o intensidad de tráfico en el enlace.

III. ESTRUCTURA DE LOS DATOS TIPO COLA

Una estructura tipo cola se sustenta en el concepto de FIFO (First In First Out), siendo esta una estructura en la cual el primero que entra es el primero que sale, ejemplo: las colas que se forman en el supermercado al momento de pagar, siendo el primero que está en la cola el primero en ser atendido. Tomando este concepto una

[1] p_n es la probabilidad de que haya *n* usuarios (paquetes o llamadas) en la cola, incluyendo el que esta en servicio.

cola será: un conjunto dinámico que obedece a la propiedad FIFO. La mecánica de una cola viene descrita en la figura 5.

Los elementos ingresan por la parte inferior y salen por la parte superior originando que el primero en entrar sea el primero en salir (observe en la figura 5 la numeración dada a los círculos).

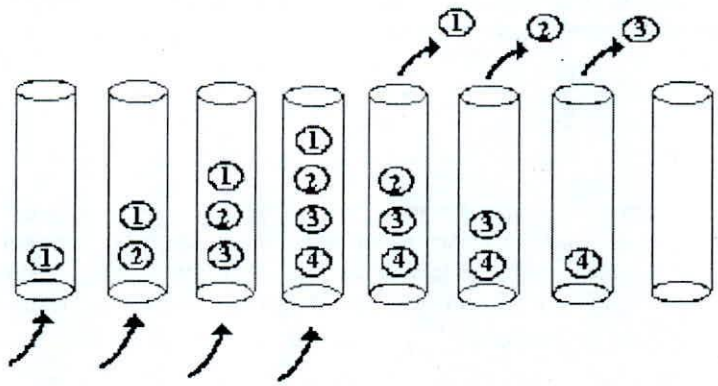
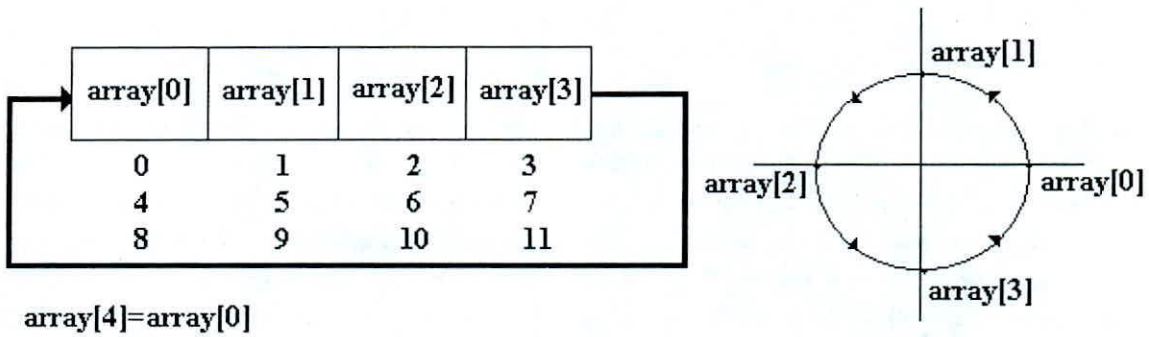


Figura 5. Comportamiento de una cola

3.1 Implementación de una cola

Para poder diseñar un simulador de colas, primero se debe implementar una cola a través de *arrays*[2] y no de nodos (opción clásica), debido a que, para este caso se obtiene, una mayor eficiencia del simulador. Se utiliza un *array* de tipo circular siendo este aquel en que la última sigue a la primera componente. Veamos el caso de un *array* de 4 elementos mostrado en la figura 6:



array[4]=array[0]
array[5]=array[1]
. .
array[n]=array[n%4] n>=4

Figura 6 Relación de los índices en una cola circular.

[2] La traducción sería *''arreglo''* entendido conforme a la idea de tabla, aunque también se puede emplear el termino *''matriz''*. Se ha decidido mantener el nombre original, por que en la mayoría de literaturas especializada en programación utiliza la palabra *''array''*.

Se podrá ver que para cualquier valor del índice del *array* siempre habrá un elemento correspondiente en el *array* circular, por ejemplo: Si el índice es igual a *n=389* y el *array* circular tiene una longitud de 16, el elemento que le corresponde a ese índice será el elemento dado por el *array* [5]; se hace uso del operador módulo " % " [3] para determinar la posición correspondiente, esto es:

$$Array\,[389]=array[389\%16]=array[5] \tag{2.8}$$

3.2 Representación de las colas a través de un array circular

Se denota por:

h : Índice del *array* donde está almacenado, el primer elemento de la cola, es decir, el que puede salir.
t : Índice del *array* donde está almacenado, el último elemento de cola, es decir, el último que fue ingresado a la cola.
N: Longitud total del *array*.

La operación de **eliminar** implicará actualizar el valor de *h*, es decir, cada vez que se extraiga un elemento de la cola el índice *h* debe tomar un nuevo valor, existen 2 posibilidades la primera es que el valor actual del índice sea menor que la longitud del *array* por lo que la actualización de *h* se dará incrementando su valor en uno; la segunda opción se presentará cuando el valor de *h* sea mayor que la longitud del array actualizándose el valor de *h* a través del operador módulo (Ver figura 7).

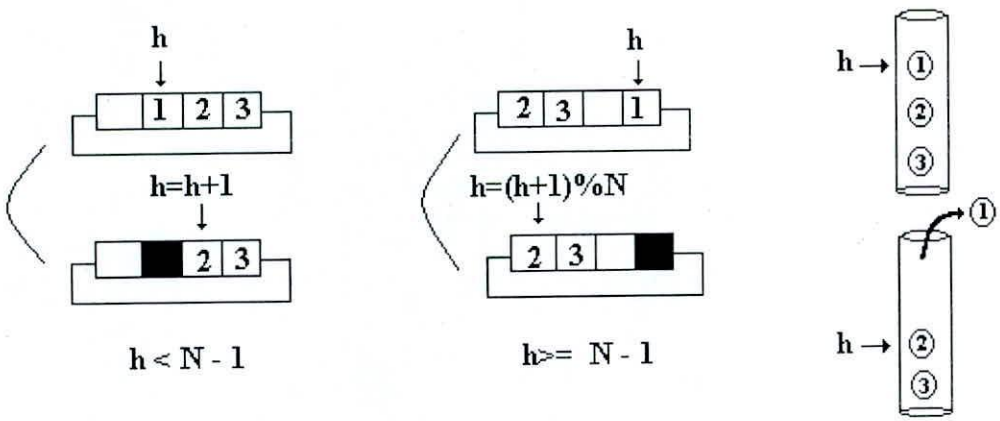


Figura 7. Operación de eliminar

La operación de **añadir** implicara actualizar el valor de *t*, esta operación también presenta dos opciones similares al caso anterior, donde la actualización de *t* se dará cada vez que ingrese un nuevo elemento a la cola (observar figura 8).

[3] El operador modulo permite calcular el residuo que se obtiene al dividir sus 2 operadores, siendo el dividendo el operador de la izquierda y el divisor el operador de la derecha..

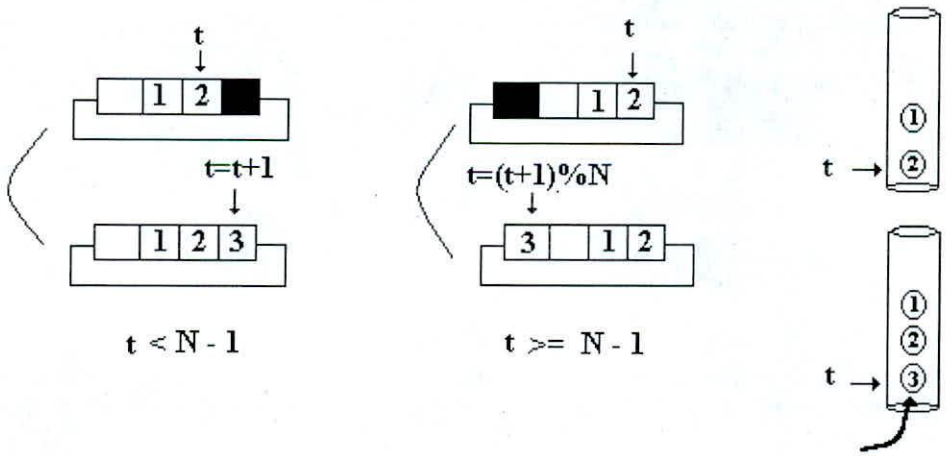


Figura 8. Operación de añadir

IV. SISTEMA DE SIMULACIÓN DE COLAS

El objetivo de un programa de simulación es modelar algún sistema físico de tal forma que se pueda analizar y predecir su comportamiento. Se utiliza una colección de variables de estado para representar el estado del sistema físico. A menudo las variables de estado de un sistema se definen para representar los estados de las entidades involucradas. Por ejemplo, en la simulación del tráfico aéreo, un avión sería una entidad del sistema, y su estado podría estar definido como en el aire, en tierra, aterrizando o despegando.

Habitualmente, se desea modelar un sistema estocástico; en este caso, a una o más variables de estado se les asigna valores de acuerdo con una distribución de probabilidad. Además, los cambios en el estado del sistema son normalmente medidos en función del tiempo. En una simulación de un sistema continuo el estado del sistema está evolucionando con el tiempo. Por el contrario, en una simulación de sistema discreto a las variables de estado sólo se les permite cambiar en instantes de tiempo discreto.

En su forma más simple un sistema de colas consiste en un único servidor que atiende a una única cola. Esta situación se ilustra en la figura 9.

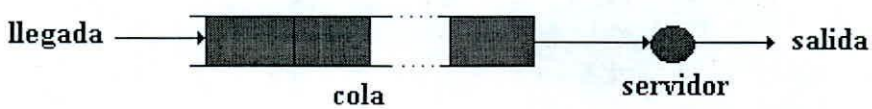


Figura 9. Modelo de un sistema de colas

Con el fin de simular este sistema se debe especificar la siguiente información:

1. ***La distribución del tiempo de llegada de clientes***. Se asumen que los clientes llegan de acuerdo a una distribución de probabilidad conocida *A(x)*. concretamente. *A(x)=Pr*{tiempo entre llegadas consecutivas $<x$}.

2. ***La distribución del tiempo de atención***. Se asume que el tiempo de atención de un cliente individual varía de acuerdo a una distribución de probabilidad conocida *S(x)*. Concretamente, *S(x)=Pr*{tiempo de atención al cliente<*x*}.

La construcción del simulador supondrá la creación de tres estructuras[Gregory L.Heilman, 1998]:

Estructura Clientes: Modela la lista de espera.(simplemente la cola)
Estructura Servidor: Modela la atención a los clientes.
Estructura organizador: Planifica los sucesos que ocurren durante la simulación.

4.1 Estructura Clientes

Está estructura es responsable de mantener la estadística relacionada con la lista de espera. Se asume que los clientes entran en la cola de acuerdo a una distribución de probabilidad conocida. Los miembros de datos de está estructura son:

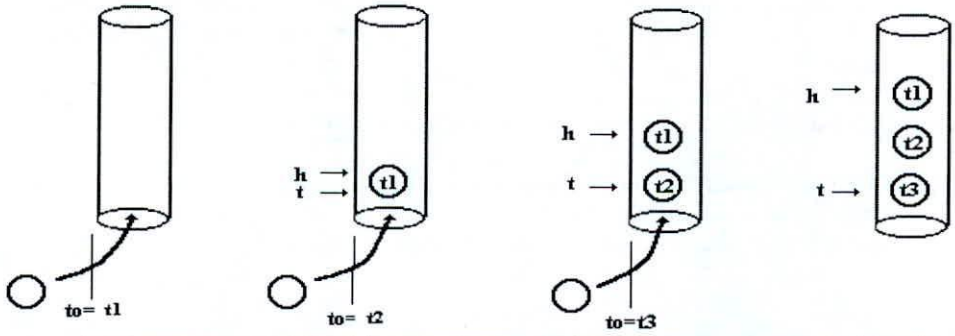
- ***longitud***: longitud de la cola
- ***tam_max_cola***: se utiliza para mantener el tamaño máximo que alcanza la cola, este indicador es útil si se considera una cola infinita.
- ***total_clientes***: número total de clientes atendidos en la simulación.
- ***clientes _perdidos***: clientes que no entran o que no quieren entrar en la cola.
- ***tiempo_ult_suceso***: almacena el momento en el que se realizó la última operación de añadir o avanzar sobre una cola.
- ***tiempo_total_espera***: almacena el tiempo de espera acumulado por todos los clientes.
- ***tiempo_max_espera***: almacena la espera más larga en la cola que cualquier cliente haya sufrido.
- ***tam_total_cola***: se utiliza para calcular el tamaño medio de la cola, almacena la suma acumulada de los productos de los tamaños de la cola por la cantidad de tiempo que la cola tuvo un tamaño particular. De este modo, el tamaño medio de la cola se puede determinar dividiendo el valor almacenado en tam_total_cola, entre el tiempo total de simulación.

Las funciones miembro que modifican cualquiera de los miembros de datos de la estructura *cola _ clientes* son las funciones *entrar_cola_cliente* y *salir_cola_cliente*, cuyas acciones a realizar se muestran a través de las figuras 10 y 11 respectivamente. Con el fin de calcular las diversas estadísticas asociadas a la cola, sólo es necesario almacenar en la cola los tiempos en que los clientes entran. También se han incluido las funciones *estado_cola_cliente* y *rechazado_cola_cliente* cuyos propósitos son de determinar el estado de la cola y llevar la cuenta de los paquetes pérdidos ya sea por el fenómeno de rehuse[4] o por que la cola se encontraba llena(se considera está segunda posibilidad porque se ha tomado una cola de tamaño finito)

4.2 Estructura Servidor

Es responsable de recoger las estadísticas relacionadas con el tiempo de atención. Se asume que el tiempo de atención de un cliente individual es determinado por una función de probabilidad conocida. Los miembros de datos de esta estructura son:

[4] El fenómeno de Vehuse se presenta cuando los paquetes comienzan a ser rechazados por la cola, ya sea por que la cola a tomado un tamaño muy grande o por tener un tiempo de espera y de atención muy largo. Este fenómeno es representado a través de una variable aleatoria que determina si un paquete es aceptado o no por la cola.

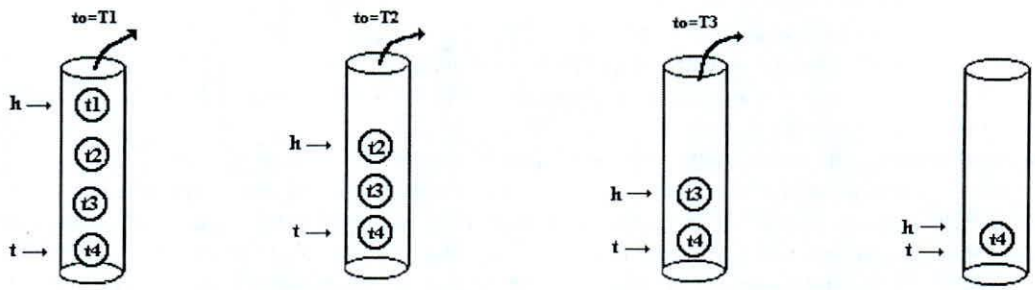


longitud :	1
tiempo_ult_suceso:	t1
total_clientes:	1
clientes_perdidos:	0
tam_max_cola:	1

longitud :	2
tiempo_ult_suceso:	t2
total_clientes:	2
clientes_perdidos:	0
tam_max_cola:	2

longitud :	3
tiempo_ult_suceso:	t3
total_clientes:	3
clientes_perdidos:	0
tam_max_cola:	3

Figura 10. Operaciones a realizar por la función miembro Entrar



longitud	3
tiempo_ult_suceso	T1
espera	T1 - t1
si tiempo_max_espera<espera entonces tiempo_max_espera=espera	
tiempo_total_espera	to+espera

longitud	2
tiempo_ult_suceso	T2
espera	T2 - t2
si tiempo_max_espera<espera entonces tiempo_max_espera=espera	
tiempo_total_espera	to+espera

longitud	1
tiempo_ult_suceso	T3
espera	T3 - t3
si tiempo_max_espera<espera entonces tiempo_max_espera=espera	
tiempo_total_espera	to+espera

Figura 11. Operaciones a realizar por la función miembro *Salir*

- ***clientes_atendidos***: Numero total de clientes atendidos por el servidor durante la simulación.
- ***tiempo_total_atencion***: cantidad de tiempo necesario para realizar ese servicio.
- ***ocupado_hasta***: se utiliza para almacenar el momento en el que el servidor cambiará de estado ocupado a estado libre. Un objeto servidor está en estado ocupado siempre que esté atendiendo a un cliente y está en estado libre siempre que esté disponible para atender a un cliente.

Las funciones Miembro que determinan el comportamiento del servidor son:

Disponible_En_servidor; esta función es responsable de calcular el instante en el que el servidor estará disponible para atender.

Ocupado_Servidor; entrega un valor booleano indicando si el servidor cambio de estado o si continua ocupado.

Atender_cliente_servidor el cual actualiza la estadística del servidor y devuelve un valor booleano indicando que el servidor ya no atiende paquetes.

El retardo es el tiempo empleado en entregar un paquete al servidor y este procesarlo, dicho valor es establecido en la simulación a través de una variable aleatoria.

De la figura 12, se observa el tiempo que le toma al servidor estar libre para atender otro paquete viene dado por el miembro de dato *ocupado_hasta*, veamos un ejemplo que engloba el manejo de tiempos por parte del servidor; en el tiempo de simulación to=2seg. se decide atender un paquete (*to*>tiempo_atencion), el cual es pasado al servidor y procesado generando dicha operación un retardo de 5 seg., entonces, el servidor estará libre de recibir otro paquete en el 2+5=7 seg., siendo este valor almacenado en la variable *ocupado_hasta*.

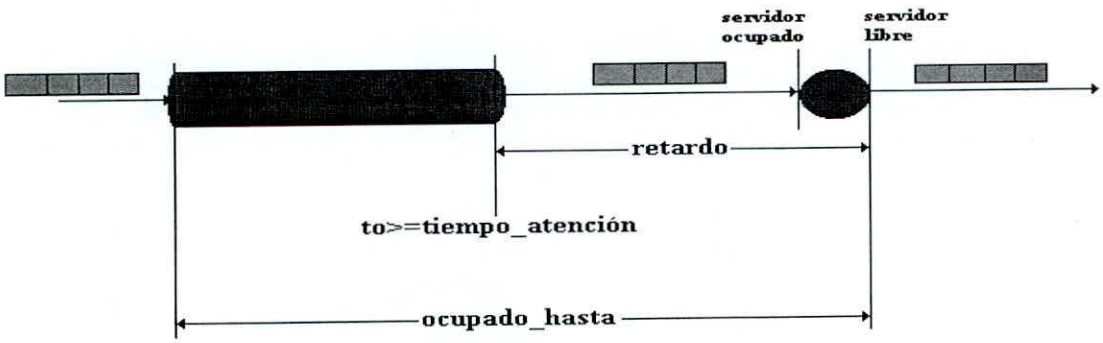


Figura 12. Tiempos de retardo y atención vinculados al servidor

4.3 Estructura Organizador

Controla el tiempo de simulación y es responsable de añadir y hacer avanzar clientes en las colas. Una función invocada por la estructura lleva a cabo la simulación. El siguiente seudocódigo ofrece una visión general de cómo el organizador funcionará en está simulación. Este, simula un sistema de colas con un único servidor inicializando el reloj de simulación t, e incrementándolo en la cantidad fija Δt en cada iteración.

Simular por tiempo (tiempo t_{inicio},tiempo t_{fin})

1 inicializar el estado del sistema y el reloj de simulación.
2 **mientras**($t < t_{fin}$) **hacer.**
3 $i \leftarrow i+1;\ t \leftarrow i * \Delta t$
4 **si** (t □ $t_{llegada}$), **entonces.**
5 determinar el estado de la cola.
6 **si** (la cola no está llena) **entonces.**
7 añadir un cliente a la cola y actualizar las estadísticas.
8 se calcula el tiempo de llegada del próximo paquete.
$t_{llegada} \leftarrow t + t_A \Leftrightarrow t_A$ se ha generado deacuerdo a $f_\tau(\tau)$
9 **caso contrario.**
10 se ha perdido un cliente.
11 **si** (t □ $t_{atencion}$) **entonces.**
12 estado_servidor ← inactivo

13 **si** (estado_servidor = inactivo y hay clientes en la cola) **entonces**
14 quitar un cliente de la cola y actualzar estadísticas
15 estado_servidor ← ocupado
16 se calcula el tiempo de atención del próximo paquete.
17 $t_{atención} \leftarrow t + t_s \Leftrightarrow t_s$ se ha generado deacuerdo a $f_r(r)$

En cada iteración del bucle de las líneas 2-16, se utilizan varias sentencias condicionales para comprobar si algún suceso se ha producido durante el intervalo de tiempo anterior. Por ejemplo, si llegó un cliente durante el intervalo de tiempo anterior y la cola no está llena, las sentencias de la línea 4 y 6 se evaluará el acierto provocando que el cliente sea añadido a la cola y que el tiempo de llegada del siguiente paquete sea generado.

4.3.1 Intervalo de Muestreo Optimo

¿Cómo asegurar que cada evento en la cola (llegada o atención) caerá en un intervalo de análisis pudiendo así, el evento ser registrado? . Para lograrlo se deberá establecer una relación entre el intervalo de simulación, el tiempo entre llegadas ($1/\lambda$) y el tiempo de atención de un solo paquete ($1/u$). Comenzaremos nuestra deducción con la llegada de un paquete en el tiempo *to*, el cual, se deberá encontrar en el intervalo de análisis, lo que implicara:

$$0 \leq t_o \leq \Delta t \tag{4.1}$$

permitiéndonos establecer la igualdad:

$$t_0 = k\Delta t \quad ; \; k \in \leq 0,1 \geq \tag{4.2}$$

Ahora supongamos que en el siguiente evento (atender un paquete) cae en el segundo intervalo de análisis (Figura 13), implicando que:

$$\Delta t \leq t_1 \leq 2\Delta t \tag{4.3}$$

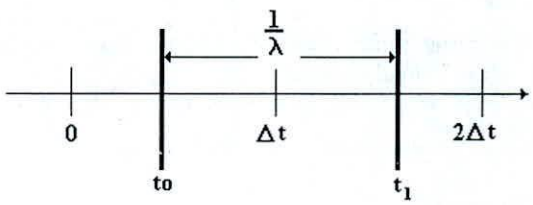


Figura 13. Intervalos de simulación para dos eventos consecutivos.

En promedio se podrá establecer la relación:

$$t_1 = t_o + \frac{1}{\lambda} \tag{4.4}$$

Remplazando (4.4) en (4.3) se obtiene:

$$\Delta t \leq t_o + \frac{1}{\lambda} \leq 2\Delta t \tag{4.5}$$

Remplazando (4.2) en (4.5) y desarrollando la desigualdad obtenemos las cotas para el intervalo de muestreo,

$$\frac{1}{(2-k)\lambda} \leq \Delta t \leq \frac{1}{(1-k)\lambda} \tag{4.6}$$

Por lo tanto, el mínimo intervalo de muestreo vendrá dado por:

$$\Delta t_{\min} = \frac{1}{(2-k)\lambda} \tag{4.7}$$

Para que Δt sea mínimo *(2-k)* debe ser máximo, por lo que *k* debe tomar su mínimo valor, es decir, *k=0* (ver 4.2). Tomando (4.7) la forma:

$$\Delta t_{\min} = \frac{1}{2\lambda} \tag{4.8}$$

Analizando en forma similar para los eventos de atención de paquetes se obtiene:

$$\Delta t_{\min} = \frac{1}{2\mu} \tag{4.9}$$

por lo tanto, el Δt optimo será:

$$\Delta t = \min\left\{\frac{1}{2\lambda}, \frac{1}{2\mu}\right\} \tag{4.10}$$

V. COMPORTAMIENTO ESTADISTICO DE LA COLA M/M/1

La probabilidad que se tenga *n* usuarios en la cola en el tiempo *t*, es equivalente a la probabilidad que la cola tenga una longitud *n* en el tiempo *t* (figura 14). Para entender mejor el concepto de p_n, supongamos que tenemos la cola con *N* estados(donde *N* es la longitud máxima de la cola).

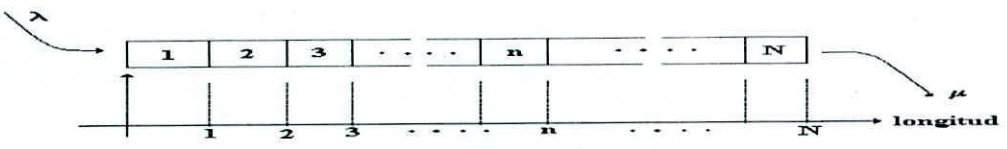


Figura 14. Vínculo entre la longitud con el estado de la cola.

Analicemos el caso de una transición de estados.

En el momento t_n nos encontramos en el estado *n*, a partir de aquí se tienen 2 opciones que indicarán una variación de estado, que llegue un paquete más, aumentando así la longitud de la cola en 1 o que sea atendido

un paquete, disminuyendo la longitud de la cola en 1 (Figura 15). Supongamos que alguna de estas dos opciones se da en el tiempo t_{n+1}, entonces, se podrá afirmar que $t_{n+1}-t_n$ fue el tiempo que la cola permaneció en el

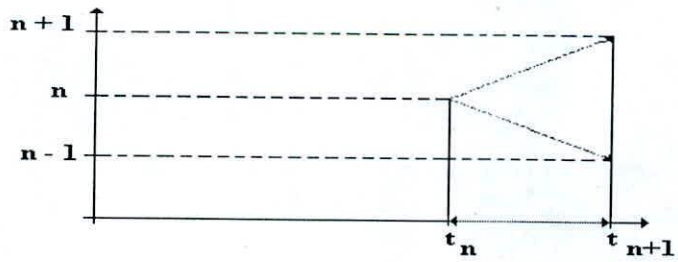


Figura 15. Paso del estado actual al siguiente

estado n (figura 15). Lo expuesto nos permite definir la probabilidad que se tenga n usuarios en el tiempo como la suma de todos aquellos intervalos en los cuales la cola tenga una longitud de n (figura 16), dividido por el tiempo total de simulación. Esto es:

$$P_n = \frac{1}{T}\sum_{i=1}^{m_n}\left(t_{n+1}^{i} - t_n^{i}\right) \tag{5.1}$$

donde: m_n es el número de intervalos del tiempo de simulación en los cuales la cola a tenido una longitud de n.

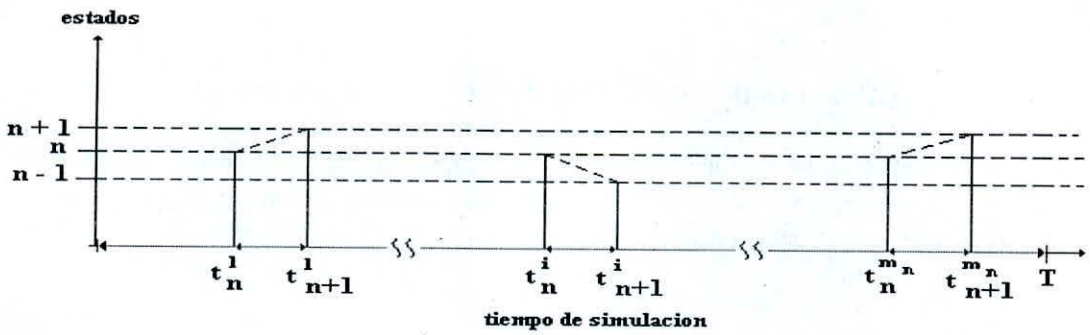


Figura 16. Intervalos de tiempo que la cola estuvo en el estado n.

Podremos definir el valor medio de p_n, es decir, los paquetes promedios en la cola como:

$$\langle P_n \rangle = E(n) = \sum_{n=1}^{N} nP_n = \frac{1}{T}\sum_{n=1}^{N}\sum_{i=1}^{m_n} n.\left(t_{n+1}^{i} - t_n^{i}\right) \tag{5.2}$$

La variable que contiene el valor de la doble sumatoria viene dado por *tam_total_cola*, por lo que, será necesario dividirla por el tiempo total de simulación para obtener *E(n)*.

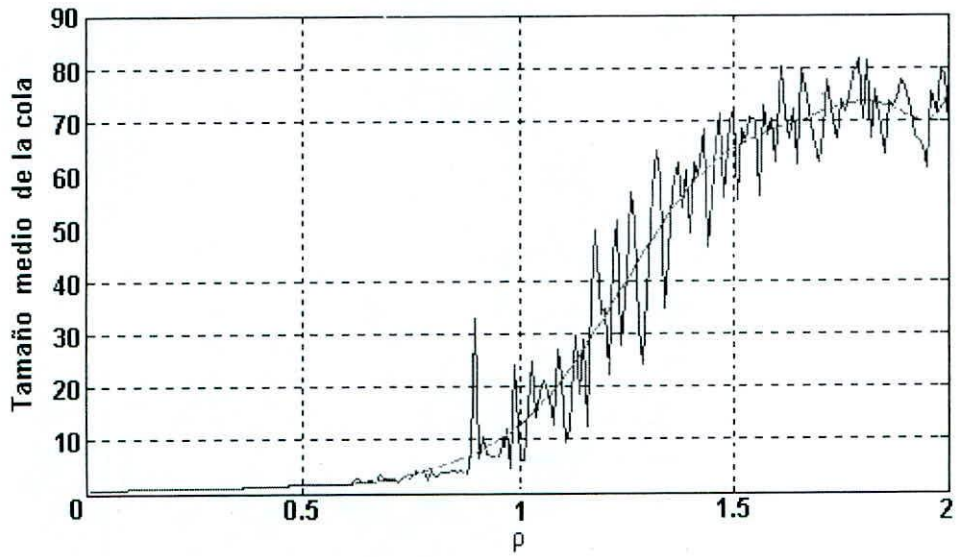


Figura 17. Tamaño medio de la cola versus ρ

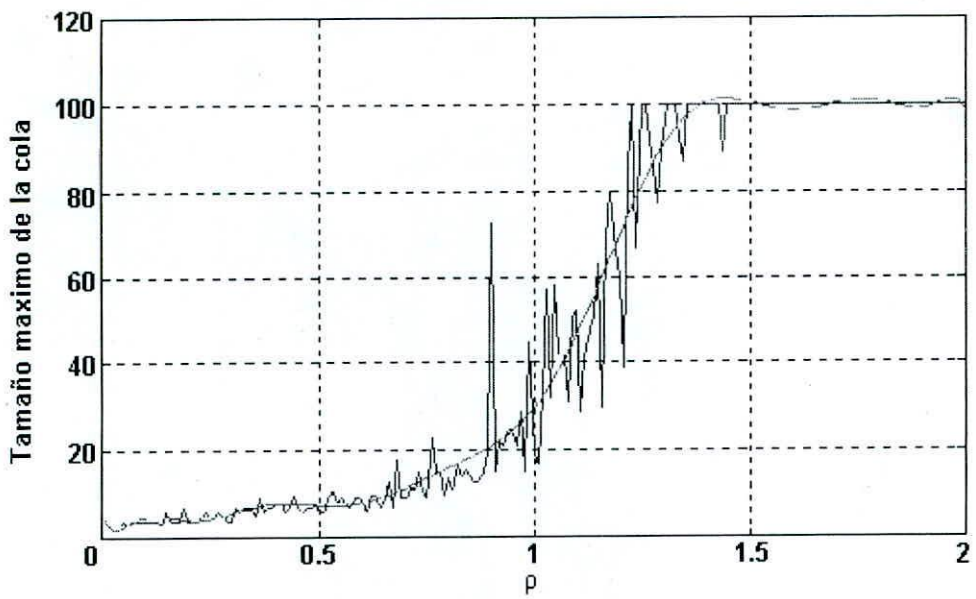


Figura 18. Tamaño máximo de la cola versus ρ

A partir de la ecuación (5.1) podemos determinar la probabilidad de rechazo como la sumatoria de todos aquellos intervalos donde la cola alcanzó su tamaño máximo (ver figura 19), dividido por el tiempo total de simulación. Para este caso no se considera la opción de Vehuse, es decir, unicamente los paquetes serán rechazados de la cola si es que esta se encuentra llena, cuya ecuación es:

$$P_B = \frac{1}{T}\sum_{i=1}^{m_N}\left(t^i_{N+1} - t^i_N\right) \tag{5.3}$$

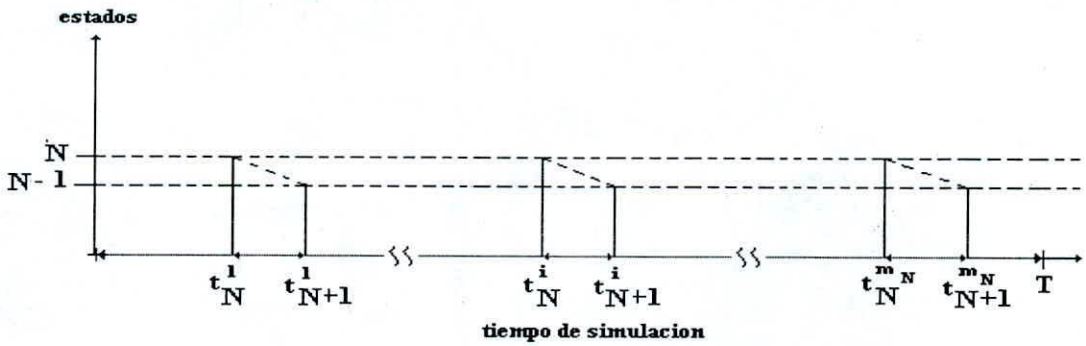


Figura 19. Intervalos de tiempo que la cola estuvo llena.

Los tiempos de retardo en la cola estarán conformados por el tiempo medio de espera en la cola y el tiempo medio de servicio. El tiempo medio de espera en la cola lo determinamos sumando los tiempos de espera de cada uno de los paquetes a ser atendidos por el servidor, divididos por el número de paquetes que serán atendidos.

$$\frac{1}{\#\text{ paquetes atendidos}}\sum_i \text{tiempo de espera}_i = \frac{\text{tiempo\_total\_espera}}{\text{clientes\_atendidos}} \tag{5.4}$$

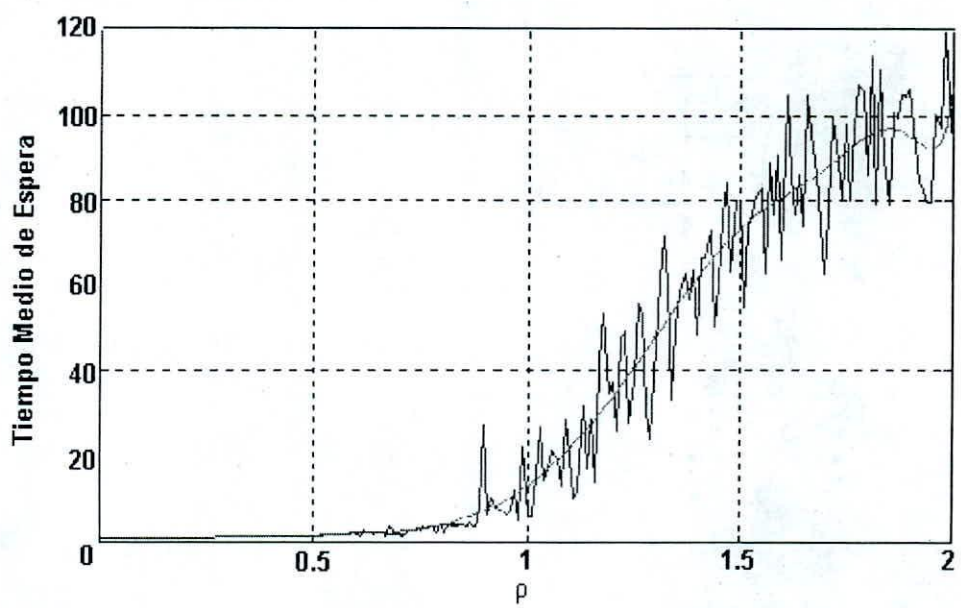


Figura 20. Tiempo medio de espera versus ρ

El tiempo medio de servicio se determina sumando los tiempos que a tomado atender a cada uno de los paquetes, dividido por el número de paquetes atendidos.

$$\frac{1}{\#\ \text{paquetes}\quad \text{atendidos}} \sum_{i} \text{tiempo}\quad \text{de atención}\ _{i} = \frac{\text{tiempo}\ \_\ \text{total}\ \_\ \text{atencion}}{\text{clientes}\ \_\ \text{atendidos}} \tag{5.5}$$

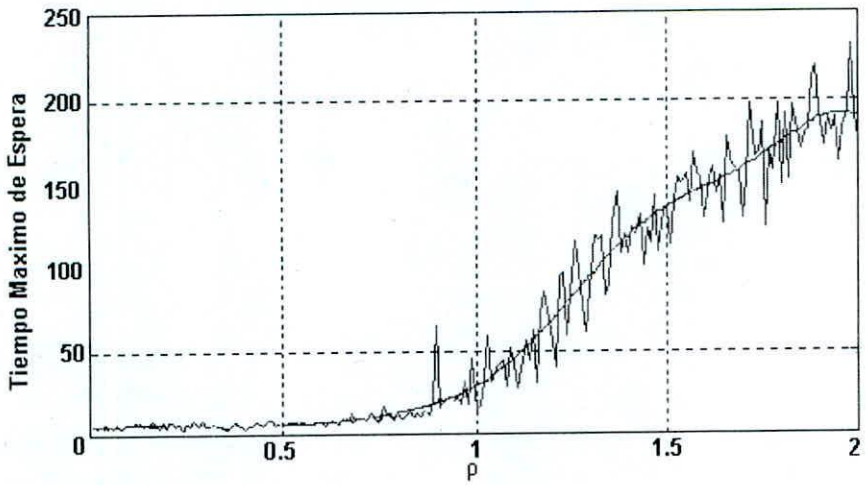
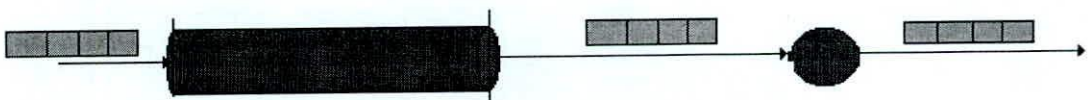


Figura 21. Tiempo máximo de espera versus ρ



tiempo_total_espera:
tiempo total que se ha esperado para atender a los clientes.
tiempo_max_espera:
La espera más larga que haya sufrido un paquete en ser atendido.

clientes_atendidos:
Numero total de clientes atendidos por el servidor.
tiempo_total_atencion:
Tiempo total que toma atenderlos.

Figura 22. Variables relacionadas con los tiempos de espera tanto de la cola como del servidor.

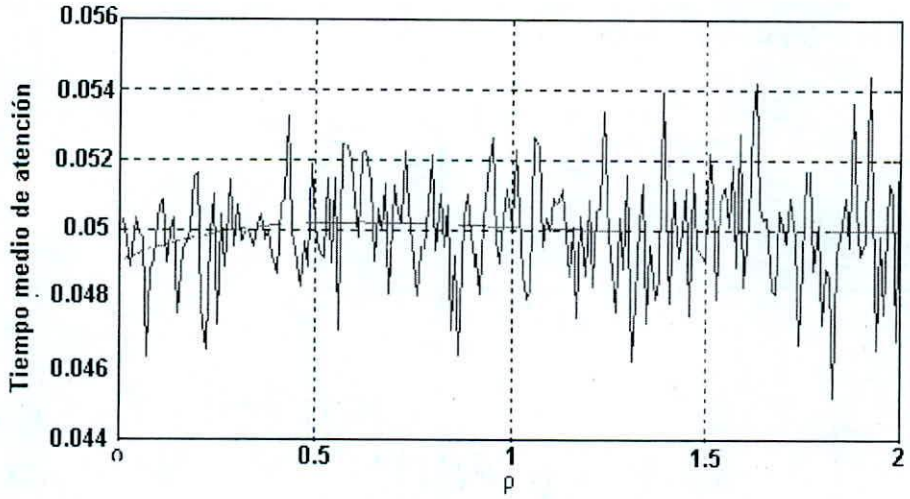


Figura 23. Tiempo medio de atención versus ρ

El número de paquetes rechazados y atendidos por la cola para un ρ determinado se almacenan en las variables *clientes_atendidos* y *clientes_rechazados*.

Para la obtención de las gráficas se ha tomado una longitud máxima de la cola de 100 elementos, se puede observar que el número de clientes atendidos cae drásticamente y la pérdida de clientes se hace evidente cuando la relación $\rho = \lambda/\mu$ es mayor que uno, dado que, llegan más paquetes de los que se pueden atender.

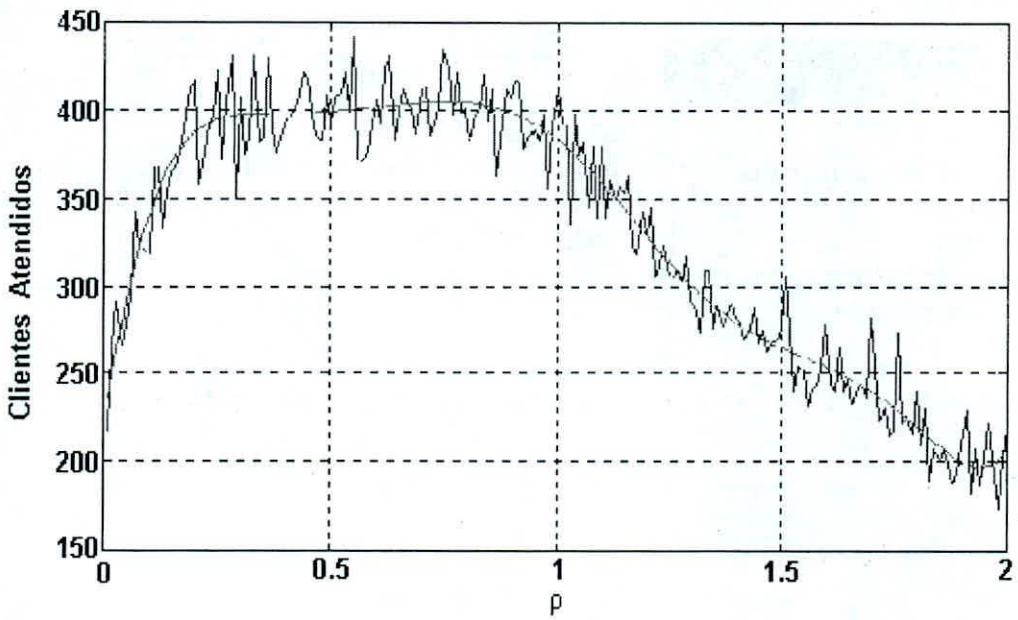


Figura 24. Clientes atendidos versus ρ

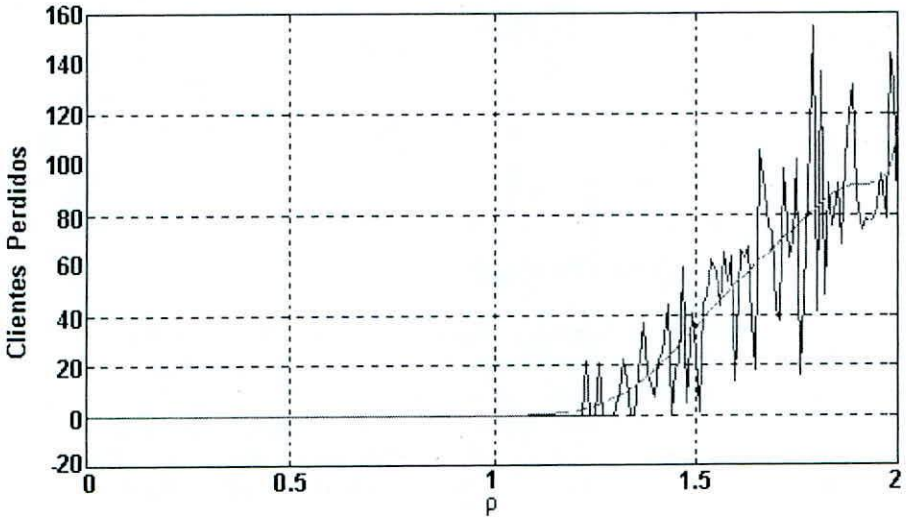


Figura 25. Clientes perdidos versus ρ

estos resultados fueron obtenidos a través del siguiente programa, ejecutado en MATLAB, donde se vé que el tiempo de simulación es de 400 seg., la máxima longitud de la cola es de 100 elementos.

```
N=100;
TiempoFinal=400;
dt=min([1/landa_medio 1/mu_medio])/2;
 t=0;tllegada=0;tatencion=0;t_ultima_llegada=0;t_ultima_atencion=0;
 while(t<TiempoFinal)
 t=t+dt;
 %*****************************COLA*********************************
 if(t>=tllegada)
    estado_cola=estado_cola_cliente(cola_cliente,N);
    if((estado_cola~=2))
    %*********se añade un elemento a la cola**********************
    perdidos=cola_cliente.clientes_perdidos;
    cola_cliente=entrar_cola_cliente(cola_cliente,tllegada);
    else
    cola_cliente.clientes_perdidos=cola_cliente.clientes_perdidos+1;
    end
    %*********se calcula el tiempo de llegada del proximo paquete**
    landa=VarExponencial(landa_medio,1,0);
    tllegada=t_ultima_llegada+landa;
    t_ultima_llegada=tllegada;
  end
  %*****************************SERVIDOR*****************************
  if(t>tatencion)
    estado_servidor=1;
    if((estado_servidor==1)&((estado_cola==1)|(estado_cola==2))&(cola_cliente.longitud>0))
    %quitar un elemento de la cola
    cola_cliente = salir_cola_cliente(cola_cliente,t);
    %actualizar las estadisticas
    servidor=AtenderCliente_servidor(servidor,t);
    estado_servidor=0;
    end
```

```
    %*****se calcula el tiempo de atencion del proximo paquete***
     mu=VarExponencial(mu_medio,1,0);
     tatencion=t_ultima_atencion+mu;
     t_ultima_atencion=tatencion;
  end
end
```

V. CONCLUCIONES

De las gráficas obtenidas para el tiempo medio de la cola, el tiempo medio de espera, clientes atendidos y clientes perdidos, se puede observar que la simulación presenta el comportamiento esperado de una cola M/M/1, siendo una relación de importancia $\rho = 1$, dado que, para valores mayores a 1 la cola pierde estabilidad. El tiempo medio de atención presenta un grado de distorsión respecto a la curva esperada, por que la variable *tiempo_total_atención* no solo utiliza la variable aleatoria que define el tiempo entre paquetes atendido, sino que también incluye una variable aleatoria con un comportamiento lineal que simula el efecto del retardo propio de la atención del servidor (ver figura 12).

Un inconveniente al realizar la simulación con Matlab es el tiempo que demora todo un proceso de simulación, para nuestro caso una Pentium 1, de 133Mhz, el tiempo de simulación fue de 3 horas 28 minutos. Una alternativa a este inconveniente es llevar el código ya optimizado en Matlab, a un lenguaje de bajo nivel como C++. Para mas detalles ver [Gregory,1998].

El vincular el tiempo de muestreo de la simulación con los tiempos promedios de llegada y atención de un paquete asegura la obtención de una data estadística confiable, puesto que, todos los eventos generados aleatoriamente han sido considerados, reflejándose en las carencias de valores negativos en las variables encargadas de recoger la estadística de la cola.

VI. BIBLIOGRAFÍA

- Luis Joyanes Aguilar, *Programación Orientada a Objetos*, Osborne Mc Graw-Hill,1998.
- Gregory L.Heilman, *Estructura de datos, algoritmos y programación orientada a objetos*, Mc Graw-Hill,1998.
- Matlab edición de estudiante, Prentice Hall, 1996.
- Thomas L. Saaty, *Elementos de la Teoría de Colas*, Mc Graw-Hill,1967.
- Mischa Schwartz, *Redes de telecomunicaciones*, Mc Graw-Hill,1994.