

AVANCES EN INVESTIGACION FORMATIVA

Memorias del I Encuentro de Investigación Formativa, 2010

Universidad Pontificia Bolivariana



Escuela de Ingenierías

Facultad de Ingeniería Industrial

Grupo de Investigación en Sistemas Aplicados en la Industria (GISAI)

2010

PRÓLOGO

Hablar de la investigación formativa en el entorno académico implica necesariamente hacer un ejercicio de reflexión e interiorización acerca de nuestro quehacer docente en aras de construir los pilares básicos del proceso investigativo desde el aula, es si se quiere, la posibilidad manifiesta del encuentro y desencuentro con el alumno y el docente en un permanente dialogo de saberes acerca de los múltiples objetos de estudio que tanto la realidad como la ciencia y la técnica nos convocan a problematizar desde nuestro claustro académico, es entonces, una imperiosa necesidad de abordar desde las pequeñas dudas hasta los complejos problemas la voluntad inquebrantable de la academia por formar en el hacer y en el pensar para servir a una sociedad ávida de soluciones que nos demanda día a día ingentes esfuerzos por vincularnos estrechamente a sus cotidianidades, es entonces hablar sobre el cómo volvernos y volver al otro y a lo otro con la clara vocación de seguirmos sorprendiendo, extrañando y curioseando en nuestra permanente búsqueda de la verdad histórica que nos convoca hoy y siempre.

Siendo así, la Dirección de la Facultad de Ingeniería industrial a través de su **Grupo de Investigación Sistemas Aplicados en la Industria (GISAI)** de la Universidad Pontificia Bolivariana considerando importante y necesario dar a conocer ante la comunidad académica de nuestra universidad los resultados parciales y finales de los proyectos de aula en el marco del desarrollo de nuestro proceso de investigación formativa que actualmente adelanta la Escuela de Ingenierías y en específico la Facultad de Ingeniería Industrial, han realizado este nuestro **I ENCUENTRO DE INVESTIGACION FORMATIVA EN INGENIERIA INDUSTRIAL**.

Evento que conto con la participación activa de docentes, investigadores, estudiantes, egresados y comunidad en general para generar un diálogo de saberes donde se permita visualizar el quehacer investigativo desde nuestra aulas, donde tuvo asidero el debate, la sana critica y la confrontación respetuosa y dignificante de las ideas propias del fundamento investigativo y del espíritu crítico y científico de nuestra Universidad.

Colocamos entonces hoy a consideración de los lectores el resultado del trabajo en equipo y las publicaciones derivadas en forma de ponencias que fueron enviadas y presentadas en este **I ENCUENTRO DE INVESTIGACION FORMATIVA EN INGENIERIA INDUSTRIAL**.

Msc. Javier Darío Fernández Ledesma

Director Grupo de Investigación GISAI

Universidad Pontificia Bolivariana, Facultad de Ingeniería Industrial

MODELO DE SIMULACION EN UN CALL CENTER

Luisa Fernanda Peláez

David Tabares Moreno

Adriana Padilla Espinosa

Docente: Javier Darío Fernández Ledesma

Área: Optimización

RESUMEN

En este trabajo simulamos por medio de promodel un problema de colas específicamente en un *call center* aplicado al outsourcing prestado por la compañía ALLUS GLOBAL BPO CENTER donde se atienden n número de clientes para ayudar a resolver sus inquietudes.

Se definieron las variables de entrada que son las llamadas de clientes, variables de estado la duración de la llamada y las variables de salida el número de clientes atendidos; gracias al programa se obtienen datos sobre las llamadas atendidas, el tiempo de espera de los clientes en la cola y el total de llamadas recibidas, al igual que las llamadas no atendidas. Sirviendo de esta manera el modelo como una buena base para la toma de decisiones dentro de la organización.

INTRODUCCION

Según [1] un Call Center es un centro de servicio telefónico que tiene la capacidad de atender altos volúmenes de llamadas, con diferentes objetivos. Su principal enfoque es el de la generación de llamadas de Salida (Llamadas de Outbound) y la recepción de llamadas (Llamadas de Inbound), cubriendo las expectativas de cada una de las campañas implementadas”

Para estas empresas en concreto es muy importante conocer datos de la calidad y la cantidad de llamadas efectuadas o atendidas, la razón es muy sencilla, el principal negocio de estas empresas se centra en la realización y recepción de llamadas con lo cual el control de la información que hace referencia a las llamadas es de vital importancia para valorar el negocio y beneficio de estas compañías.

En esta información se valoran datos de las llamadas y de los agentes, número de llamadas recibidas, número de llamadas realizadas, duración de las llamadas, tiempos medios, tiempos de respuesta, disponibilidad de agentes, etc. Esta información es la que ofrece nuestro programa de una forma sencilla y eficaz gracias a la simulación en promodel.

Con todos estos datos se puede conocer si se está realizando bien o no el trabajo y cuáles son los puntos críticos en el caso de que los haya para así tomar las medidas correctivas necesarias, para lograr que el servicio prestado sea óptimo y de la mejor calidad y así traiga como resultado obtener clientes satisfechos.

ALLUS GLOBAL BPO CENTER

Según [2] Allus Global es la compañía líder en América Latina en la provisión de soluciones BPO. Brinda servicios de valor y excelencia a una vasta gama de industrias convirtiéndose en una extensión de la empresa del cliente, atendiendo sus procesos a través de la innovación, la creatividad y el conocimiento.

Ventajas competitivas de alta diferenciación la posicionan como la compañía world class líder de la región. Una red global con presencia en Argentina, Colombia, Estados Unidos, España y Perú, desde donde atiende con costos competitivos y eficacia operacional a los mercados de Europa y América.

Esta compañía es experta en la provisión de servicios *outsourcing* alineados con los objetivos del cliente y flexibles a sus requerimientos con más de diez años de experiencia. Su foco en la atención de diferentes sectores industriales en varios mercados del mundo les permite consolidar un *expertise* propio con conocimiento profundo de cada industria inmersa tanto en el entorno local como en el escenario global, atendiendo las particularidades que cada contexto requiera.

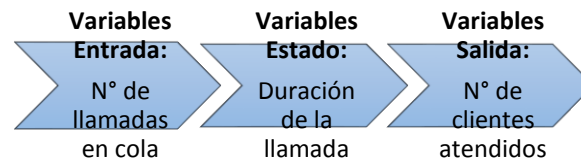
DEFINICION DEL PROBLEMA

En Colombia Allus ingresa al mercado al momento de la absorción de una empresa denominada Multitenlace, la cual se desempeñaba como outsourcing Call center para algunas compañías dentro del país.

De esta manera Allus trabaja como la nueva herramienta de outsourcing para diferentes compañías de atención al cliente como puede ser el caso de Almacenes Éxito, El Metro de Medellín, Bancolombia, entre otros. El proceso es muy simple, la compañía interesada acuerda con la prestadora del servicio el número de asesores que considere necesarios para desempeñarse durante sus horarios de atención al cliente, de esta manera, la empresa representada debe brindar una capacitación al personal de Allus, aclarando el comportamiento y el perfil de la empresa en cuanto a su línea de atención, para ello definen un guion el cual facilita la gestión y le da mayor seguridad a los asesores para responder una posible pregunta de los clientes.

Una vez estas capacitaciones son ejecutadas, se comienza con el proceso de recepción de llamadas, de esta manera se pueden presentar diferentes tipos de problemas, uno de ellos es la saturación de las líneas, lo cual suele suceder en casos de alta estacionalidad por ejemplo, para ello las compañías tienen establecidos algunos de estos estándares, sin embargo la predicción de estos consta de un margen de error estadístico lo cual hace que los métodos no sean cien por ciento seguros sino que tan solo den soporte a una reducción de la incertidumbre.

Por ello pretendemos simular el comportamiento de un call center para medir su nivel de eficacia identificando las siguientes variables:



De esta manera en nuestro trabajo comenzamos a ejecutar la simulación a través de Promodel comenzando por crear la interfaz o el layout de la simulación donde a través de Build-Background Graphics- Front of Grid, definimos el entorno donde se va a realizar la simulación, obteniendo como resultado una sala que va a ser las veces de la compañía donde se encuentren los operadores del call center.

Una vez definido el entorno debemos crear las locaciones es decir los puntos por donde se van a desenvolver los procesos indicados una vez se corra la simulación, para nuestro caso como vamos a simular un asesor que atiende varias llamadas durante el día definimos una cola de entrada, un asesor (al que definimos en el programa como Cajero), y una cola de salida. La cola de entrada la definimos como Queue pues con esto le indicamos que es por donde ingresan las entidades, de igual manera le damos que aparezca como una línea y que sea invisible durante la simulación. Así mismo definimos la cola de salida como una línea, invisible durante la simulación y de tipo converso, es decir que es por donde salen las entidades.

De esta misma manera, a través de Build-Locations agregamos un escritorio para darle forma a nuestro entorno y ponemos una marca de X para indicarle al programa que ahí deben llegar las entidades una vez se corra la simulación.

Ahora bien, ya tenemos un layout es decir una “organización”, tenemos los lugares donde se ejecutan los procesos (Locations), ahora definimos las Entidades que son las que van a recibir el servicio dentro de cada proceso, es decir son los clientes que realicen las llamadas dentro del sistema. En nuestro caso simbolizaremos una llamada con un Cliente y un grafico de una persona, para definir las entidades entramos por Build-Entities y definimos un cliente seleccionando su grafico correspondiente, de igual forma definimos un segundo grafico donde visualicemos una mujer para lograr así ver una aleatoriedad de género en cuanto al orden de las llamadas. Esto lo hacemos solo para efectos visuales.

Una vez definidas las entidades que van a recibir el servicio, es decir los clientes, debemos ejecutar la forma en que lleguen estas entidades al servicio es decir a las locations, por lo tanto ingresamos por Build-Arrivals y designamos a Cliente como entidad que llega a la Location Cola_cajero; para definir la cantidad de personas o el flujo de personas que llegan lo hacemos mediante un ciclo, ya que nuestra demanda es variable en el tiempo según el horario de atención, pues como decíamos

previamente la demanda es estacionaria. Para ello ingresamos a través de Build-More elements-Arrival Cycles e identificamos el Ciclo 1 en cantidades y le definimos la tabla de demanda donde indicamos la hora y la cantidad de Entidades que llegarán al servicio. Comenzamos por el numero 1 hasta el 24 indicando las 24 horas del día, y tendremos en cuenta las 6am como hora inicial, es decir que 1 representa la hora entre las 6 y las 7.

La demanda la definimos según los parámetros establecidos por nuestro texto guía pues eran consecuentes con la información suministrada por nuestra fuente dentro del call center. Una vez definimos este ciclo de llegada según la demanda, volvemos a las llegadas del sistema en Arrivals y terminamos por agregar la cantidad cada Ciclo 1 pues es el ciclo que ya definimos. Ahora bien para determinar la aleatoriedad visual de la que hablamos previamente entre hombre y mujer, le definimos una lógica programada de la siguiente manera:

```
WRITELINE demanda, CALHOUR()
IF id=1 THEN
{ begin graphic 2 id=0 end }
ELSE
{ begin graphic 1 id=1 end }
```

Con lo que estamos programando la simulación para alternar los gráficos que definimos en las entidades de hombre y mujer, reconociendo que la variable “id” (que definiremos posteriormente) pertenece a un comando binario donde 1 representa hombre y 0 representa una mujer.

Así pues terminamos de definir las llegadas de las entidades es decir los clientes o las llamadas recibidas dentro del sistema a cada una de las locaciones que prestaran el servicio. Por lo tanto debemos definir cuál es el proceso que se realizara dentro de cada locación; para ello ingresamos a través de Build-Processing, donde encontramos básicamente tres procesos que debemos definir, el momento cuando ingresa la llamada, el momento durante la atención, y el momento en que sale la llamada del cliente satisfecho.

Aquí determinamos tres renglones donde cada uno representará la entidad que se trasladará durante el proceso, el lugar donde lo ejecute, la operación que desarrolle, qué sale después del proceso (output), hacia donde sale (Que proceso continua) y definimos la velocidad por defecto para promodel con Move for 0.5.

Por ello definimos una operación principal que es el momento de la prestación del servicio, es decir en la entidad Cajero programamos la siguiente operación:

```
INC Llamadas_atendidas
INT min_ent
INT min_sal
```

```

min_ent=CALMIN()
WRITE Cliente_atendidos, CALHOUR()
WRITE Cliente_atendidos, min_ent
WAIT 3
min_sal=CALMIN()
WRITE Cliente_atendidos, min_sal
WRITELINE Cliente_atendidos, min_sal-min_ent
min_sal=CALMIN()

```

Con la cual estamos definiendo que cada vez que llegue una entidad a la locación Cajero, incremento la variable Llamadas atendidas, le decimos que nos escriba en un archivo de texto llamado clientes atendidos la hora, el minuto de entrada de la entidad, que espere dos minutos y vuelva a escribir el minuto, definiéndolo como minuto de salida. De igual forma le indicamos que calcule la diferencia entre el minuto de entrada y el minuto de salida y que la escriba en este mismo archivo.

El comando de esperar tres minutos para que vuelva a tomar el tiempo definiéndolo como minuto de salida, es porque estamos estipulando una duración de servicio de tres minutos.

También para controlar las llamadas en espera definimos la siguiente operación indicando los comandos:

```
INC Total_llamadas
```

```
Llamadas_espera=Total_llamadas-Llamadas_atendidas
```

```
IF Llamadas_espera=16 THEN
{begin INC No_atendidos end}
```

```
WRITE Llamadas, Total_llamadas
WRITE Llamadas, Llamadas_atendidas
WRITE Llamadas, Llamadas_espera
WRITELINE Llamadas, No_atendidos
```

Así, cada vez que llegue una llamada a la cola, se incrementara en 1 la variable Total llamadas, y le definimos las llamadas en espera como la diferencia entre las llamadas atendidas y el total de las llamadas que entren al sistema.

Ahora bien, para determinar cuántas llamadas no logran entrar al sistema, pues la cola solo tiene capacidad para 15 llamadas, entonces definimos que si contaba 16 llamadas en espera entonces incrementara la variable de los no atendidos en 1.

Una vez programamos estas operaciones, debemos definir las variables a través de Build-Variables donde incorporamos al sistema las variables que ya hemos nombrado como son Llamadas atendidas, id, Total llamadas, Llamadas en espera y No atendidos. A cada una de estas variables les activamos un contador para que vaya aumentando acorde a la programación establecida dentro del proceso.

Para que nos aparezcan los títulos de las variables podemos ingresarlas dentro del layout a través de Build-Background Graphics-Front of Grid e incorporamos un cuadro de texto donde marquemos cada contador de las variables.

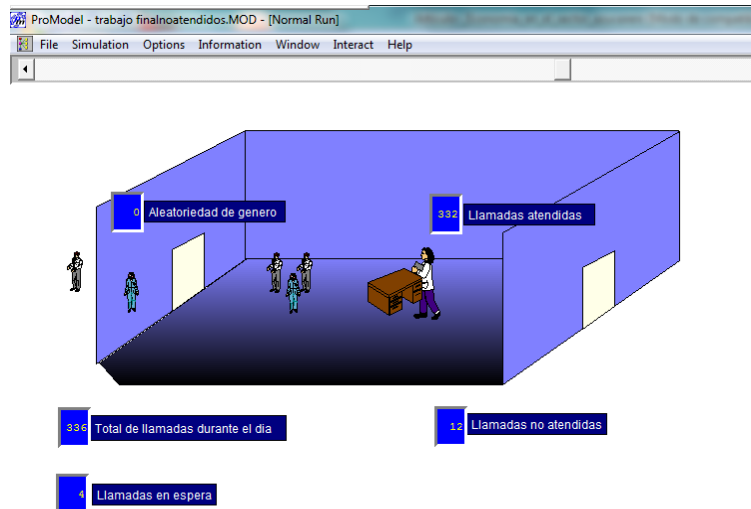
También debemos importar los archivos de texto donde vayan a quedar grabados los resultados de la simulación, es decir tenemos que darle una ruta a los comandos que procesamos dentro de las operaciones al momento de definir los Write. Por lo tanto creamos tres archivos de texto: Clientes atendidos, Demanda y Llamadas, a demás de esto ingresamos a través de Build-More elements-External Files y definimos el ID de cada uno de los archivos lo identificamos de tipo General Write (Escritura) y le damos la ruta donde se encuentra el archivo.

Llegando al final de la programación, a través de Build-General Information damos clic en Initialization Logic y determinamos el siguiente comando:

```
RESET demanda
reset Cliente_atendidos
WRITE CLIENTE_ATENDIDOS, "Horas"
WRITE CLIENTE_ATENDIDOS, "Min_entrada"
WRITE CLIENTE_ATENDIDOS, "Min_salida"
WRITELINE CLIENTE_ATENDIDOS, "Tiempo Ocupado"
reset Llamadas
WRITE LLAMADAS, "Total llamadas"
WRITE LLAMADAS, "Llamadas atendidas"
WRITE LLAMADAS, "Llamadas espera"
WRITELINE LLAMADAS, "No atendidos"
```

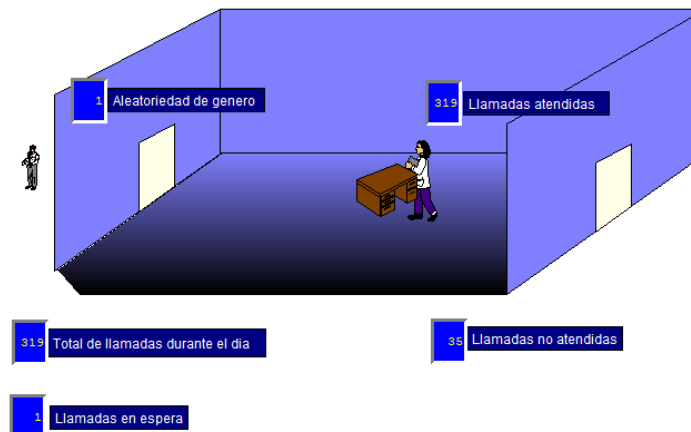
De esta manera estamos diciéndole al programa que cada vez que se corra al simulación borre los archivos de texto para dejarlos limpios nuevamente para ingresar los nuevos datos, además que escriba los títulos para cada columna dentro de cada archivo, es decir Hora, Minuto de entrada, Minuto de Salida, Tiempo Ocupado, etc.

Por último ingresamos a través de Simulation-Options y definimos Calendar date para correr la simulación durante 24 horas. Así pues, terminamos por dar clic sobre RUN y observamos como corre nuestra simulación dentro del call center.



RESULTADOS

Con toda la información suministrada obtenemos los siguientes resultados que fueron consignados en los archivos *.txt* que creamos y ligamos dentro de la simulación:



Cientes atendidos.txt

horas	min_entrada	min_salida	tiempo ocupado
0	1	4	3
0	5	8	3
0	8	11	3
0	12	15	3
0	15	18	3
0	19	22	3
0	22	25	3
0	26	29	3
0	29	32	3
0	33	36	3
0	36	39	3
0	40	43	3
0	43	46	3
0	47	50	3
0	50	53	3
0	54	57	3
0	57	0	-57
1	1	4	3
1	4	7	3
1	8	11	3
1	11	14	3
1	15	18	3

Llamadas.txt

Total llamadas	llamadas atendidas	Llamadas espera	no atendidos
1	0	1	0
2	0	2	0
3	1	2	0
4	1	3	0
5	2	3	0
6	3	3	0
7	3	4	0
8	3	5	0
9	3	6	0
10	5	5	0
11	5	6	0
12	5	7	0
13	6	7	0
14	6	8	0
15	6	9	0
16	7	9	0
17	7	10	0
18	7	11	0
19	7	12	0
20	7	13	0
21	7	14	0
22	8	14	0
23	8	15	0
24	8	16	1
25	10	15	1

Así podemos ver que contando con un tiempo de servicio de tres minutos cuando estamos en la hora 19 es decir entre las 12am y la 1am se han recibido un total de 319 llamadas durante el día para el minuto 32 en los cuales podemos identificar 318 llamadas que ya han sido atendidas, contando con

solo 1 llamada en espera, sin embargo encontramos que para este momento 35 llamadas no fueron atendidas durante el día, es decir a estas personas se les dejo de prestar el servicio, lo que indica perdida de dinero para la compañía, sin tener en cuenta la insatisfacción de los clientes y su desfavorecimiento en su good will.

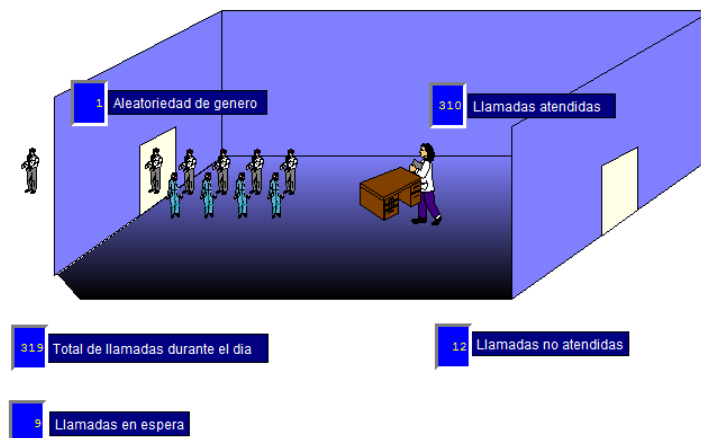
Cabe resaltar también, que en las horas de alta demanda se llega a ocupar toda la cola, es decir el sistema no es capaz de recibir una sola llamada mas, por lo que se denotan hasta 15 llamadas en espera por ser atendidas por el asesor. Al momento de contar 16 llamadas incrementa en uno el valor de las llamadas no atendidas.

También podemos ver que a medida que aumenta la frecuencia de llegada de las llamadas según la tasa de demanda, es posible visualizar como va aumentando el número de llamadas en espera así el tiempo de servicio sea el mismo, es decir que a medida que aumentan las llamadas en espera, las llamadas atendidas acumuladas son las mismas, debido al tiempo de llegada de la alta demanda.

MEJORA

Ahora bien, en la búsqueda de mejorar el servicio y disminuir el número de llamadas no atendidas, pensamos que teniendo en cuenta que nuestra simulación es para un sistema de colas M/M/1, no aplica una solución en cuanto agregar más asesores, pues se estaría cambiando a un problema de tipo M/M/n. Por lo tanto una solución válida seria la disminución en el tiempo de servicio, teniendo en cuenta que a los asesores se les hace la salvedad que deben sostener una llamada mínimamente por 2 minutos.

De esta manera disminuimos el tiempo de servicio a 2 minutos por llamada y obtuvimos los siguientes resultados:



Clientes atendidos.txt

horas	min_entrada	min_salida	tiempo ocupado
0	1	3	2
0	4	6	2
0	6	8	2
0	9	11	2
0	11	13	2
0	14	16	2
0	16	18	2
0	19	21	2
0	21	23	2
0	24	26	2
0	26	28	2
0	29	31	2
0	31	33	2
0	34	36	2
0	36	38	2
0	39	41	2
0	41	43	2
0	44	46	2
0	46	48	2
0	49	51	2
0	51	53	2
0	54	56	2
0	56	58	2
0	59	1	-58
1	1	3	2

Llamadas.txt

Total llamadas	llamadas atendidas	Llamadas espera	no atendidos
1	0	1	0
2	0	2	0
3	1	2	0
4	1	3	0
5	3	2	0
6	3	3	0
7	4	3	0
8	4	4	0
9	5	4	0
10	6	4	0
11	6	5	0
12	7	5	0
13	8	5	0
14	8	6	0
15	9	6	0
16	9	7	0
17	9	8	0
18	10	8	0
19	10	9	0
20	10	10	0
21	10	11	0
22	10	12	0

De esta manera podemos observar que disminuyendo el tiempo de servicio se presentan muchos beneficios, empezando por que para obtener un nivel de 319 llamadas atendidas durante el día, basta

con llegar a la hora 13 en el minuto 28, mientras que con un minuto de más (Tiempo de servicio de tres minutos), habría que esperar 6 horas más.

De igual forma podemos observar que mientras que anteriormente contábamos con 35 llamadas que no fueron atendidas, con estas condiciones solo tenemos 12 llamadas, contando con 310 llamadas atendidas y 9 llamadas en espera debido a la tasa de demanda pues el horario es diferente al de la primera simulación.

Esta es la mejor solución encontrada para la situación del call center establecido como un problema de colas de M/M/1 donde el tiempo de servicio fijado en 2 minutos representa la tasa de servicio optima para disminuir las llamadas que no pueden entrar al sistema debido a la congestión de llamadas.

CONCLUSIONES

- La implementación de Promodel como software para la simulación se hace muy útil al momento de pensar no solo en casos, como situaciones de logística, o producción, sino también líneas de atención, inversión o transporte, convirtiéndose en una herramienta muy útil dentro de aplicaciones para la ingeniería.
- Promodel se convierte en un paquete fácil y agradable para manejar a través de su entorno para la ejecución de diferentes simulaciones, pues permite mediante su configuración, un lenguaje de programación simple y entendible el cual se hace accesible para las personas que pretendan correr cualquier tipo de simulación.
- Al hablar específicamente del caso aplicado para nuestra simulación, la duración del servicio se disminuye a dos minutos por cliente, pues al realizar comparaciones en cuanto a la simulación con un tiempo de servicio de tres minutos, obtenemos una mayor tasa de clientes atendidos dentro de un menor lapso de tiempo.
- De igual manera con un tiempo promedio de servicio de dos minutos, se logra reducir en 23 llamadas el número de clientes no atendidos una vez se alcance el mismo nivel de llamadas (319). Por lo que se hace claro que al llegar al mismo nivel horario el nivel de llamadas no atendidas sería muchísimo menor que en el primer caso, además de que el número de llamadas atendidas sobrepasaría ridículamente el número de llamadas.
- Con la mejora establecida, se logra obtener un nivel de llamadas de 319 personas tan solo en la hora 13, mientras que en la situación actual para llegar a este mismo nivel de atención se debe esperar hasta casi el final del día en la hora 19.

- También es importante resaltar que el nivel de servicio mejora notoriamente pues aunque es un problema M/M/1 el número de llamadas atendidas aumenta notoriamente haciendo así que mediante el nivel de demanda establecida solo para la hora 13 donde se alcanza el punto de corte fijado para la comparación, la demanda es alta, haciendo que se esperen aun mas llamadas gracias al sistema planteado.
- Como dijimos con anterioridad en la búsqueda de mejorar el servicio y disminuir el número de llamadas no atendidas, pensamos que teniendo en cuenta que nuestra simulación es para un sistema de colas M/M/1, no aplica una solución en cuanto agregar más asesores, pues se estaría cambiando a un problema de tipo M/M/n.
- Al disminuir notoriamente el número de llamadas no atendidas se promueve el good will de la compañía en cuanto a su nivel de servicio, ya que es menor el numero personas a las que se les dejo de prestar el servicio, lo que indica un aumento de utilidad para la compañía, sin dejar de lado la satisfacción de los clientes.

BIBLIOGRAFÍA

Cia. Andicel de Colombia. Obtenido el 20 de Mayo de 2010 desde <http://www.gestiopolis.com/canales/demarketing/articulos/61/callcenter.htm>

Allus Global BPO Center. Innovación, creatividad y conocimiento para la nueva economía del mundo Fedesarrollo. Extraído el 20 de Mayo de 2010 desde <http://www.allus.com/compania.php>

Blanco Rivero, Luis Ernesto. Fajardo Piedrahita Iván Darío; Marzo de 2003. Simulación con Promodel Casos de Producción y Logística, Colección desarrollo empresarial, Editorial Escuela Colombiana de Ingeniería

Notas de Clase y Asesorías por Javier Darío Fernández. Simulación 2010-01, Ingeniería Industrial. Universidad Pontificia Bolivariana.