**Simulación de un conmutador telefónico mediante eventos discretos**

Cátedra: Modelos y Simulación

Carrera: Ingeniería en Sistemas. Dpto. de Cs. Exactas y Naturales

Alumno: Picón, Luis Darío

Docentes: Lasso, Marta, Vidal Pablo

Universidad Nacional de la Patagonia Austral, Unidad Académica Caleta Olivia (UNPA-UACO)

Ruta 3. Acceso Norte. Caleta Olivia. Santa Cruz. Argentina

Septiembre 2014

**Resumen**

En esta sección debe resumirse en qué consiste el problema planteado en el documento.

El objetivo de este trabajo es desarrollar un modelo de conmutador telefónico que permita evaluar bajo qué condiciones se producen variaciones en la proporción de llamadas que el sistema procesa con éxito.

No se trata concretamente de maximizar la proporción éstas, ya que naturalmente, es directamente proporcional a la cantidad de enlaces y líneas telefónicas. Por el contrario, en este trabajo se propone el desarrollo de un modelo que permita evaluar las múltiples combinaciones entre cantidad de enlaces y líneas telefónicas y analizar la proporción resultante, de manera que pueda servir de recomendación para interesados en búsqueda de configuraciones que hagan más eficiente al sistema.

**Tabla de contenidos**

[1. Presentación del problema 2](#_Toc399159120)

[2. Modelado del sistema 7](#_Toc399159121)

[3. Resultados 9](#_Toc399159122)

[4. Conclusión 14](#_Toc399159123)

[Glosario de Siglas y Acrónimos 14](#_Toc399159124)

[Referencias 15](#_Toc399159125)

**Índice de figuras**

[Figura 1. Algoritmo Ejercicio 1 3](#_Toc398010865)

[Figura 2. Algoritmo Ejercicio 2 4](#_Toc398010866)

**Índice de tablas**

[Tabla 1. Tabla de elementos del WSDL 10](#_Toc384896588)

[Tabla 2. Notación BPMN utilizada 19](#_Toc384896589)

**Introducción**

En esta sección se debe expresar la organización del documento que se presenta.

Un conmutador telefónico, en un sentido amplio, está pensado para establecer una comunicación entre dispositivos conectados a él.

A los efectos de construir una abstracción del sistema real, que simplificara el análisis, se decidió modelar un conmutador que solo administre las llamadas salientes y entrantes efectuadas entre teléfonos conectados a él por medio de líneas telefónicas. Es decir, se planteó como un sistema cerrado donde no se permitan llamadas externas.

El modelo desarrollado permite por un lado, analizar y por otro, precedir el comportamiento de un conmutador. El objetivo perseguido es conocer de qué manera varía la proporción de llamadas exitosas y los tiempos de espera por falta de enlace bajo determinados contextos

El documento se organiza de la siguiente manera: en la Sección 1, se realiza una descripción completa y detallada del problema, donde se definen los componentes principales del sistema. En la Sección 2, se presenta el modelado y diseño del sistema planteado. En la Sección 3, se presentan los resultados obtenidos al realizar experimentos con el modelo desarrollado y en la Sección 4 se describen apreciaciones personales y mejoras sugeridas en base a los resultados obtenidos.

## 1. Presentación del problema

Se debe realizar la descripción completa y detallada del problema (definir entidades, atributos, actividades, etc).

El conmutador a modelar tiene una cantidad limitada de líneas telefónicas (entre 5 y 100) conectadas a él y además tiene una cantidad de enlaces (entre 1 y 10) utilizados para conectar cualesquiera dos líneas, sujeto solo a la condición de que se puede hacer solamente una conexión a la vez por cada línea.

Los tiempos entre arribos de llamadas tienen una distribución de probabilidad exponencial con media de 3 segundos y los tiempos de servicio (o duración de las llamadas) tienen una distribución de probabilidad exponencial con media de 180 segundos.

Se supondrá que cada llamada tiene igual probabilidad de provenir de cualquier línea que no esté ocupada y que puede dirigirse a cualquier línea excepto a sí misma, sin importar si la línea está ocupada o no. Por conveniencia, la selección de destino y origen se hará en el momento que llegue la llamada, como también la duración de la llamada.

El propósito de la simulación será procesar un número dado de llamadas y determinar la proporción de exitosas y de no exitosas. Toda llamada que no se pueda establecer (por destino ocupado o por falta de enlace) se considera no exitosa e inmediatamente abandona el sistema.

Una variante será permitir que las llamadas que encuentren todos los enlaces ocupados, esperen a que se libere uno en base a primera llegada, primera servida (FIFO). Para éstas se deberá calcular el tiempo medio de espera que tuvieron por falta de enlace. En el caso de esta variante se considerarán no exitosas solo las llamadas cuyo destino este ocupado.

La **tabla xx** describe los componentes principales del sistema.

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| Estado | Las variables de estado del sistema nos permiten indicar en un determinado instante de tiempo t entre otras:   * Cantidad de llamadas en el sistema * Cantidad de llamadas que arribaron al sistema * Cantidad de llamadas que abandonaron el sistema * Cantidad de servidores ocupados |
| Entidades | * Llamadas * Enlaces |
| Atributos | Ninguna de las entidades tiene atributos distintivos que los caractericen, como podrían ser prioridad alta, media o baja para los enlaces, por ejemplo. |
| Eventos | * Llegada de una llamada * Salida de una llamada |
| Actividades | * Tiempo entre llegadas de llamadas * Duración de las llamadas |
| Conjuntos | * La cola FIFO de espera de las llamadas |

Se observa que los enlaces (que son representados en el modelo como servidores de línea de espera) no tienen asignada prioridad, es decir, cuando llega una llamada y existen al menos dos servidores libres ésta es atendida de manera indistinta por cualquiera de los dos.

Para el caso de las llamadas, tampoco existe algún tipo de prioridad al arribar. De esta manera todas llamadas y todos los enlaces tienen exactamente el mismo tratamiento.

**Ejercicio 1:**

La lógica del modelo para la primera variante del sistema se visualiza en el diagrama de flujo de la **Figura XX**.

En primera instancia se fijan la cantidad de teléfonos, la cantidad de enlaces y el tiempo de simulación. Una vez ingresados estos datos ésta comienza.

Las llamadas arriban al sistema de acuerdo a la distribución de probabilidad establecida. Al llegar cada llamada le es asignado el teléfono de origen, el teléfono de destino y la duración de la llamada. Bajo estas condiciones puede ocurrir que el teléfono de destino este ocupado, que todos los enlaces estén ocupados, o ambos. En cualquiera de estos tres casos la llamada no se puede establecer y se considera no exitosa; inmediatamente abandona el sistema.

Si el teléfono destino está libre y existe al menos un enlace disponible la llamada se considera exitosa. Permanece en el sistema durante el tiempo que dure la llamada, y al culminar éste abandona el sistema.

Al alcanzar el tiempo de simulación fijado, ésta termina. Se entrega como resultado la proporción de llamadas exitosas y de no exitosas. Para simplificar el diagrama de flujo no se incluyo la validación cíclica que constata no excederse del tiempo de simulación fijado.

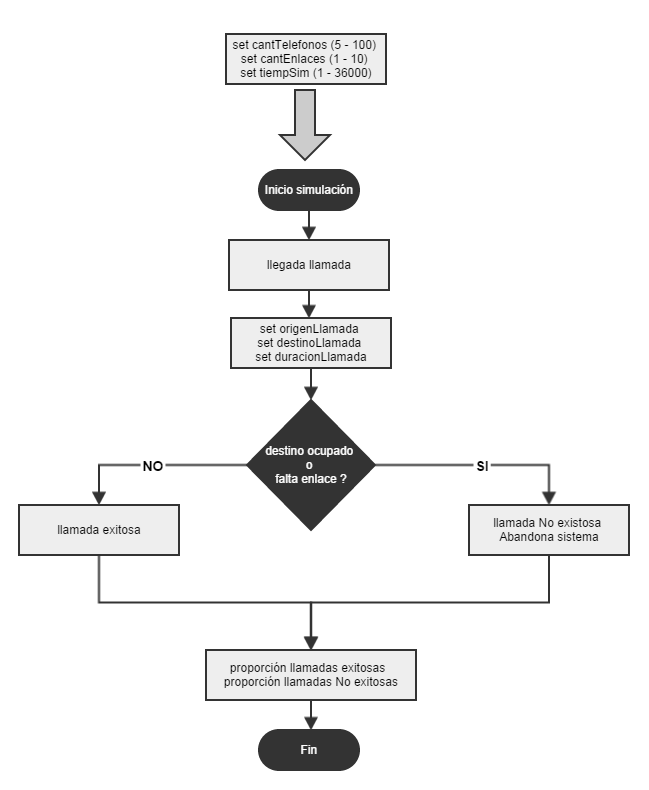


Figura 1. Algoritmo Ejercicio 1

**Ejercicio 2:**

La lógica del modelo para la segunda variante del sistema se visualiza en el diagrama de flujo de la **Figura XX**.

Las condiciones iniciales son idénticas a la primera variante, pero el tratamiento que se le da a las llamadas que encuentran todos los enlaces ocupados es distinto.

Al llegar una llamada puede ocurrir que esté ocupada, sin enlace, o ambas.

En caso de estar el destino ocupado, independientemente de si hay enlaces libres, se considera no exitosa y abandona el sistema.

Si el destino está disponible pero no hay enlaces disponibles para establecer la llamada, esta se coloca en una cola FIFO a la espera de que se desocupe un enlace.

Es importante remarcar que una llamada con destino ocupado y sin enlaces disponibles será considerada ocupada

En caso más deseado de encontrar el destino libre y enlaces disponibles, la llamada es atendida inmediatamente y se considera exitosa.

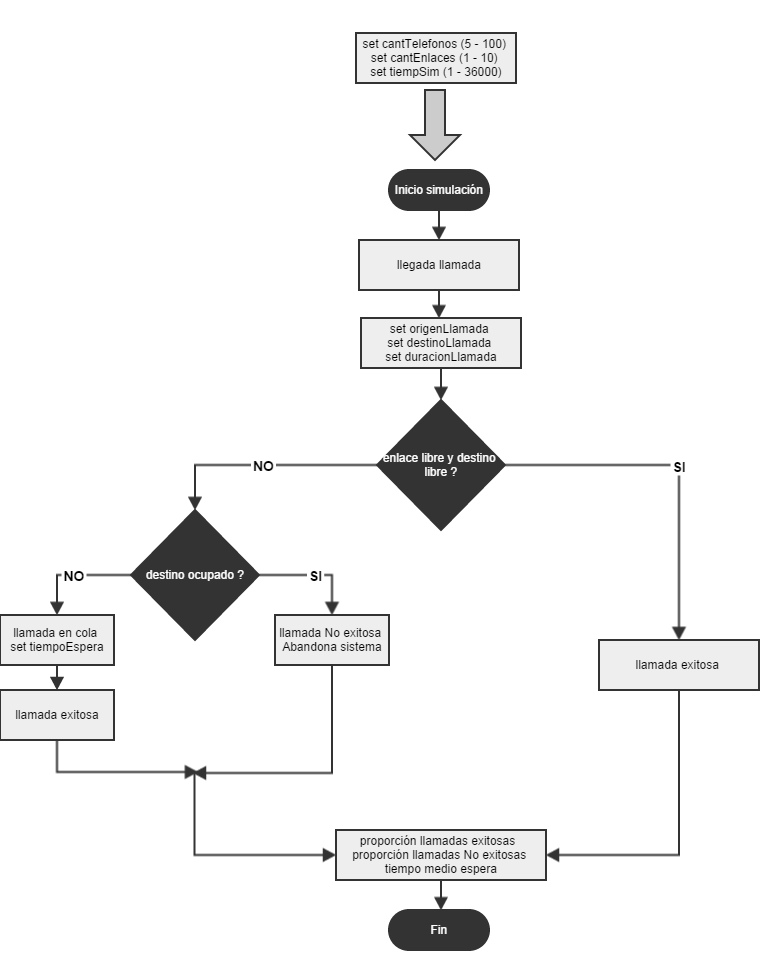
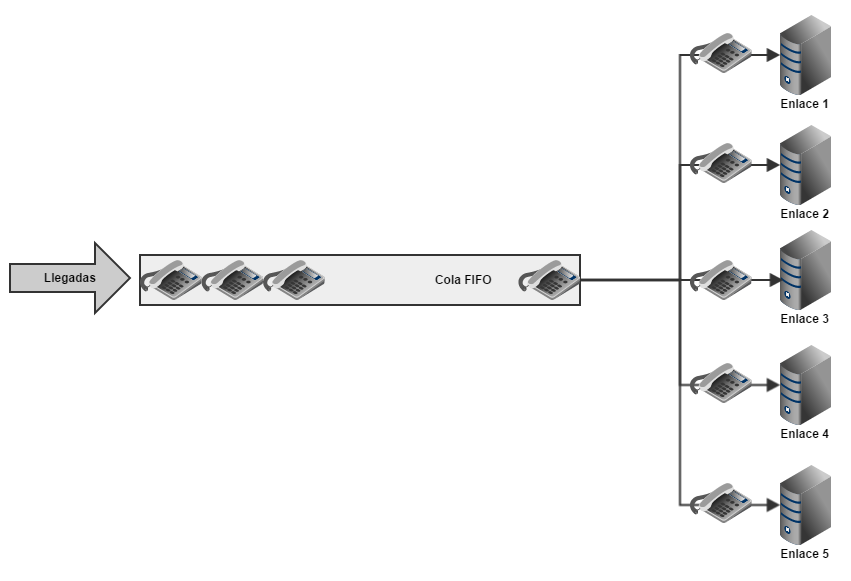


Figura 2. Algoritmo Ejercicio 2

La **Figura XX** representa gráficamente la manera en que las llamadas que ingresan esperan en una cola FIFO al encontrar todos los enlaces ocupados. Se puede apreciar como cada enlace es modelado como un servidor.



## 2. Modelado del sistema

Presentar el modelado y diseño del sistema que se plantea, explicar el lenguaje en el que se ha desarrollado y la justificación de su elección.

El núcleo del modelo se basa en un sistema de línea de espera con n servidores en paralelo, donde n en este caso en concreto representa los enlaces del conmutador telefónico y varía entre 1 y 10. Se trata de una simulación de eventos discretos, es decir, el avance del tiempo se produce en intervalos variables cuando ocurre un evento, y solo ante tales ocurrencias es posible actualizar las variables de estado del sistema.

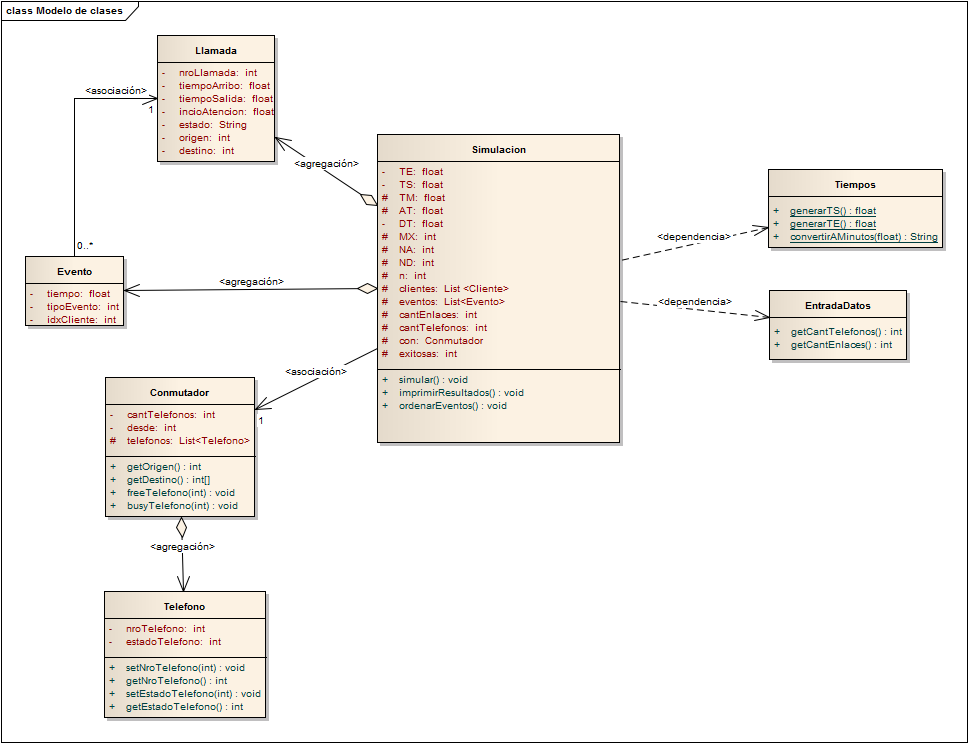
Al núcleo, además, se le han realizado algunas alteraciones y adicionamientos para ajustarse al funcionamiento del conmutador.

El modelo se implemento con el lenguaje de programación de propósito general Java, utilizando el paradigma orientado a objetos (OOP). No se utilizo ninguna librería existente en Java para simulación de sistemas de eventos discretos.

Se desestimo la utilización de la herramienta de simulación ARENA.

El sistema cuenta con una clase principal que ejecuta a través de su método simular() la funcionalidad principal. Es un algoritmo completamente lineal y estructurado, sin embargo se sirve de otras clases auxiliares o de soporte para lograr su objetivo.

Esta estrategia permite reducir la complejidad y encapsular en cada clase auxiliar un comportamiento concreto, lo que las hace a su vez completamente reutilizables.



El comportamiento de cada clase y sus respectivos métodos y atributos se explicara a continuación:

|  |  |  |
| --- | --- | --- |
| **Simulacion** | | |
| Clase principal | | |
| **Atributos** | | |
| TE | Tiempos entre llegadas | |
| TS | Tiempo de servicio | |
| TM | Tiempo de reloj de la simulación (tiempo actual) | |
| AT | Tiempo de la siguiente llegada | |
| DT | Tiempo de la siguiente salida | |
| MX | Tiempo máximo de simulación. Establece el tiempo limite a partir del cual dejan de ingresar llamadas al sistema | |
| NA | Número de llegadas hasta el instante TM | |
| ND | Número de salidas hasta el instante TM | |
| n | Número de clientes en el sistema en el instante TM | |
| clientes | Un ArrayList de clientes | |
| Eventos | Un ArrayList de eventos. Representa un Array dinámico de elementos de tipo “evento” en el cual se van añadiendo los eventos de llegada o de salida que va generando el sistema | |
| cantEnlaces | Guarda la cantidad de enlaces ingresada por el usuario en tiempo de ejecución para una determinada corrida | |
| cantTelefonos | Guarda la cantidad de teléfonos ingresada por el usuario en tiempo de ejecución para una determinada corrida | |
| con | Un objeto de tipo Conmutador | |
| exitosas | Atributo que cuenta durante una corrida la cantidad de llamadas exitosas | |
| **Métodos** | | |
| float generarTS() | | Genera los tiempos de servicio |
| float generarTE() | | Genera los tiempos entre arribos |
| simular() | | Metodo principal |
| imprimirResultados() | | Genera una salida por pantallas del resultado de una corrida |
| ordenarEventos() | | Asegura que el ArrayList de eventos se mantenga ordenado de manera ascendente. |

Los tiempos durante la simulación se operan en segundos, y para las salidas se formatean en minutos y segundos.

Durante la simulación se producen dos tipos de eventos: el arribo de una llamada al sistema y la salida de una llamada del sistema. Cada evento que se genera es colocado en una Lista de eventos que constantemente se mantiene ordenada de menor a mayor para asegurar que el próximo evento atendido por el sistema sea efectivamente el más reciente. Además el reloj que controla el tiempo del sistema actualiza su tiempo en base a los eventos

## 3. Resultados

En esta sección de deberán mostrar los resultados de la simulación.

Cada experimento o corrida toma como datos de entrada la Cantidad de teléfonos, la Cantidad de enlaces y el Tiempo de simulación. Los tiempos entre llegadas y de servicio no se encuentran parametrizados por ser considerados datos constantes.

Los datos de entrada son fijados por el usuario al inicio de la simulación por medio de una interfaz grafica de usuario (GUI). Al culminar cada experimento los resultados son visualizados a través de un panel.

El reporte de la ejecución muestra los siguientes campos:

**Nro llamada**: Número natural que identifica a cada llamada y es asignado de acuerdo al orden de llegada al sistema.

**Tiempo arribo**: Instante de tiempo en que la llamada arriba al sistema.

**Inicio atención**: Instante en que el sistema asigna un enlace a la llamada.

**Tiempo salida**: Instante en que la llamada abandona el sistema.

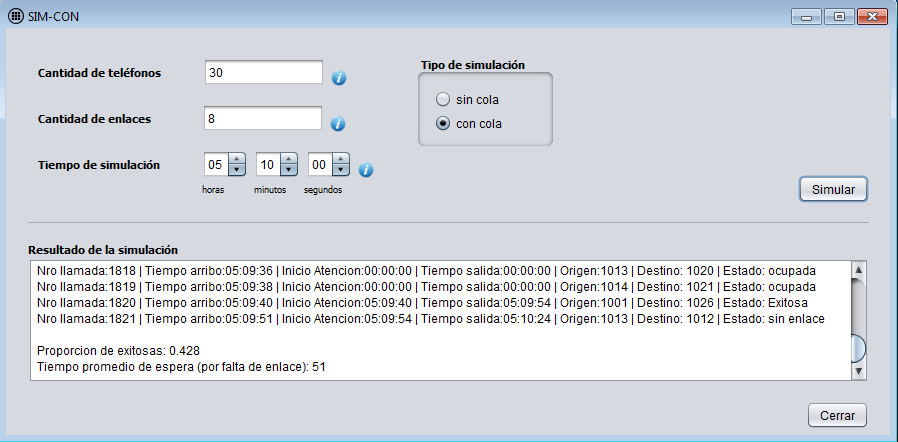
**Origen**: Número de teléfono que inicio la llamada.

**Destino**: Número de teléfono destino de la llamada.

**Estado**: Indica como culmino la llamada [ocupada/ sin enlace / Exitosa].

**Proporción de exitosas**: Indicador que muestra sobre el total de llamadas procesadas durante el experimento, que proporción de ellas se completó exitosamente.

La **Figura xx** expone la interfaz grafica del sistema.



Como se mencionó al presentar el problema, el objetivo del experimento es determinar la proporción de llamadas exitosas y de no exitosas. Para ello se decidió modelar dos variantes del sistema. En la primera, las llamadas que encuentran el destino ocupado o el conmutador sin enlaces libres abandonan inmediatamente el sistema.

En la segunda, las llamadas que encuentran el destino libre pero el conmutador sin enlaces disponibles, se colocan en una cola FIFO a la espera de que se libere un enlace para ser atendidas.

**Resultados ejercicio 1:**

Considerando que la cantidad de teléfonos del conmutador puede variar entre [5 , 100] y la cantidad de enlaces entre [1 , 10] se decidió analizar la proporción resultante para cada combinación entre ellas, fijando un tiempo de simulación de 10 horas para las pruebas. En una sección posterior se analizará en profundidad la razón de esta decisión y que variabilidad ofrece al modelo.

Para cada una de estas combinaciones se ejecutaron tres corridas, de cada una de ellas se obtuvo la proporción y finalmente el valor de la proporción que se fijo para dicha combinación fue el promedio de las tres. La razón de ejecutar varias corridas y promediarlas es aplacar el comportamiento estocástico del sistema debido a que existen componentes aleatorios como los tiempos entre arribos, los tiempos de servicio y la selección del origen y destino de cada llamada.

Las proporciones de exitosas para la primera variante del modelo se aprecian en la **Figura xx**

Como es natural esperar, la proporción de llamadas exitosas crece a medida que se incrementa la cantidad de enlaces, y a medida que se incrementa la cantidad de teléfonos. A pesar de ello, se observan variaciones en las velocidades de crecimiento que merecen un análisis.

La velocidad de crecimiento de la proporción de exitosas por variación de la cantidad de enlaces es baja para cantidades de teléfonos en el intervalo [5 , 15] aproximadamente. Sin embargo, a partir de una cantidad de 15 teléfonos la velocidad de crecimiento de la proporción por variación de la cantidad de enlaces es superior.

La velocidad de crecimiento de la proporción de exitosas por variación de la cantidad de teléfonos es baja para cantidades de enlaces en el intervalo [1 , 3] aproximadamente y a partir de los 4 enlaces la velocidad de crecimiento de la proporción aumenta considerablemente. Sin embargo a partir de cantidades de teléfonos de 25 o superiores se visualiza como la proporción tiene a mantenerse constante.

En cuanto a los valores de la proporción, el máximo valor se obtiene fijando la máxima cantidad de teléfonos y enlaces, es decir 100 teléfonos y 10 enlaces. Para esta combinación se obtiene una proporción de llamadas exitosas del 0.503.

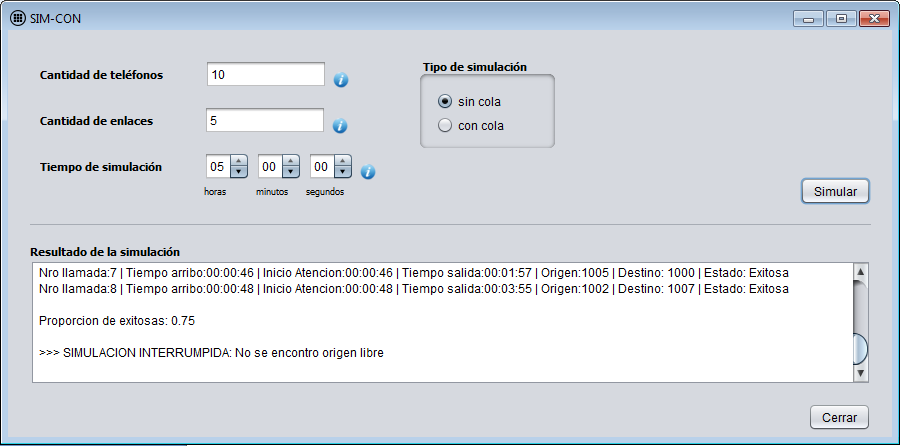
El mínimo valor se observa fijando la cantidad de enlaces en uno, cualquiera sea la cantidad de teléfonos. En este caso la mínima proporción observada fue del 0.047.

**Tratamiento números de teléfono pares:**

Dado que los tiempos de servicio son considerablemente superiores a los tiempos entre llegadas, al fijar una cantidad de teléfonos par y solo para ciertas combinaciones entre cantidad de teléfonos y enlaces, sucede que en algún instante de la simulación quedan todos los teléfonos emparejados y al llegar la siguiente llamada no encuentra ningún teléfono libre al intentar seleccionar el origen. Ante esta situación la simulación se detiene

.

Para todas estas combinaciones se opto por hacer que el sistema se detenga e informe al respecto en el reporte de la ejecución. La **figura XX** permite visualizar la manera en que el sistema lo informa.



Otra alternativa para manejar esta restricción hubiese sido descartar directamente la llamada, por ser imposible poder efectuarla desde un teléfono ocupado, sin embargo esto hubiese roto con la distribución de llegadas previamente fijada.

Así, para evitar que el sistema se detenga inesperadamente al seleccionar una cantidad de teléfonos (x) par, habría que asegurarse que la cantidad de enlaces (y) seleccionada sea menor a la cantidad de teléfonos dividida por 2.

Por ej. para 6 teléfonos, al seleccionar 3 o más enlaces el sistema se detiene. La razón es que al crecer la cantidad de enlaces, el sistema permite el establecimiento de mayor cantidad de llamadas de manera simultánea, comprometiendo de esta manera mayor cantidad de teléfonos hasta el punto en que todos quedan apareados.

Siguiendo con el ejemplo, al seleccionar 1 o 2 enlaces, el sistema puede comprometer simultáneamente como máximo 2 o 4 teléfonos respectivamente, pero nunca 6, ya que para ello se necesitan al menos 3 enlaces.

Las **figuras XX y XX** permiten visualizar gráficamente cual sería el límite de enlaces validos para una cantidad de teléfonos par.

La **figura XX** hace una extrapolación para cualquier cantidad fijada de x e y.

La figura XX presenta el umbral dentro de los intervalos fijados para el problema.

**Resultados ejercicio 2:**

Las proporciones de exitosas para la primera variante del modelo se aprecian en la **Figura xx**

Al igual que en la primera variante del sistema, la proporción de llamadas exitosas crece a medida que se incrementa la cantidad de enlaces, y a medida que se incrementa la cantidad de teléfonos. También se observan variaciones en las velocidades de crecimiento análogas a la de la primera variante.

En cuanto a los valores de la proporción, el máximo valor también se obtiene fijando la máxima cantidad de teléfonos y enlaces, es decir 100 teléfonos y 10 enlaces. Para esta combinación se obtiene una proporción de llamadas exitosas del 0.564.

El mínimo valor se observa fijando la cantidad de enlaces en uno, cualquiera sea la cantidad de teléfonos. En este caso la mínima proporción observada fue del 0.056

Los tiempos medios de espera se aprecian en la **Figura xx**

El tiempo promedio de espera de una llamada por falta de enlace crece a medida que aumenta la cantidad de teléfonos y disminuye a medida que se aumenta la cantidad de enlaces.

En cuanto a los valores de espera, el máximo valor se obtiene fijando la cantidad de teléfonos en 99 y la cantidad de enlaces en 1. Para esta combinación se obtiene un tiempo de espera promedio de 7898 segundos, es decir aproximadamente 2 horas con 11 minutos.

El mínimo valor se observa combinando una cantidad de teléfonos baja con una cantidad de enlaces alta. En este caso el menor tiempo de espera promedio observado fue de 0.

**Tratamiento números de teléfono pares:**

Para evitar las detenciones inesperadas por falta de origen en esta variante, no alcanza con asegurarse que la cantidad de enlaces (y) seleccionada sea menor a la cantidad de teléfonos dividida por 2, como sucedía con la primera variante.

Vamos a recordar que en la primera variante del sistema las llamadas que no encuentran enlaces disponibles se descartan, en cambio en la segunda variante, se colocan en una cola FIFO siempre que el destino este libre. Esto hace que la velocidad en que los teléfonos se comprometen aumente considerablemente.

Por ej. para 6 teléfonos, al seleccionar 2 enlaces, el sistema puede establecer simultáneamente 2 llamadas (comprometiendo 4 teléfonos), pero además puede comprometer los 2 restantes si llega una llamada que los coloque como origen y destino, agotando de esta manera todos los teléfonos del sistema y como consecuencia la próxima llamada no encontrara teléfono disponible al intentar seleccionar el origen.

Con el aumento del número de teléfonos y la cantidad de enlaces esta situación se revierte. La grafica de la figura XX permite visualizar con rojo el área en la que se manifiestan detenciones del sistema, con verde el área donde la simulación transcurre normalmente y con amarillo un área donde dada la naturaleza estocástica del sistema, se han presentado tanto ejecuciones satisfactorias como detenciones inesperadas.

**El tiempo de simulación:**

Para cualquiera de las dos variantes del modelo se observo que la proporción de llamadas exitosas disminuye a medida que aumenta el tiempo de simulación, es decir, son inversamente proporcionales. Es lógico que así sea en un modelo de colas en el cual los tiempos entre arribos son considerablemente superiores a los tiempos de servicio, por ende los recursos se ven comprometidos inmediatamente y los próximos arribos son descartados o colocados en cola hasta tanto se liberen recursos.

Esta tendencia se observa hasta aproximadamente un tiempo de simulación de 30 minutos. A partir de este tiempo la proporción de exitosas permanece prácticamente constante, es decir converge, por mas que se fijen tiempos de simulación de 5 o 10 horas.

# 4. Conclusión

En base a los resultados obtenidos, cual es su apreciación personal, que mejoras sugiere.

//En esta misma sección se coloca la discusión y conclusión.

El sistema desarrollado puede ejecutarse en cualquier computadora que tenga instalada la maquina virtual de java

A partir de los resultados obtenidos se observa lo siguiente:

**Ejercicio 1:**

**Ejercicio 2:**

A medida que disminuye la cantidad de teléfonos, disminuye la proporción de llamadas exitosas, pero también disminuye el tiempo promedio de espera por falta de enlace. Esta disminución en el tiempo promedio de espera se explica al haber menos teléfonos en el sistema se producen más casos de llamadas ocupadas.

Sin embargo al parecer la velocidad de disminución del tiempo promedio de espera es mayor que la velocidad de disminución de la proporción.

Se puede apreciar al fijar una determinada cantidad de teléfonos que a medida que crece la proporción de llamadas exitosas, decrece el tiempo medio de espera. La justificación es que la proporción de exitosas crece por aumentar la cantidad de enlaces y tal incremento como consecuencia hace disminuir los tiempos de espera promedio. De esta manera, la proporción de llamadas exitosas es inversamente proporcional al tiempo de espera promedio.

**Para ejercicio 1 y 2:**

El reporte post ejecución permite monitorear el estado de cada llamada que ingresó al sistema. Una mejora a implementar a futuro seria un reporte grafico y en tiempo real que permita monitorizar en avance del experimento y generar reportes en cualquier instante de la simulación.

¿De qué manera se podría incrementar la proporción de llamadas exitosas?

Naturalmente, disminuyendo el tiempo entre arribos y los tiempos de servicio la proporción de llamadas exitosas tiende a crecer. Suponiendo que estos datos son obtenidos a través de la observación del sistema real, o de datos históricos, no es viable alterarlos.

A modo de hipótesis, algunas maneras de incrementar la proporción de exitosas son:

* Comprometer el teléfono de destino recién cuando la llamada es atendida.
* Aplicando alguna política de interrupción de llamadas que han superado determinada duración, siempre que la llamada entrante tenga como destino algunas de las líneas comprometidas.
* Colocando una cola FIFO por cada enlace.

# 

# Glosario de Siglas y Acrónimos

FIFO: First In First Out

OOP: Object-Oriented Programming

# Referencias

* Ross, Sheldon M. (1999). Simulación. Prentice Hall.
* Winston, Wayne L. (2005). Investigación de operaciones. Aplicaciones y algoritmos. Thomson.