

# **TRABAJO ESPECIAL DE GRADO**

## **Identificación de la Óptima Operación en Estructuras de Redes de Colas**

Presentado ante la ilustre  
Universidad Central de Venezuela  
por la Br. Aneliz B. Rico T.  
para optar al título de  
Ingeniera Electricista.

Caracas, 2016.

# **TRABAJO ESPECIAL DE GRADO**

## **Identificación de la Óptima Operación en Estructuras de Redes de Colas**

TUTOR ACADÉMICO: Dr. Ebert Brea

Presentado ante la ilustre  
Universidad Central de Venezuela  
por la Br. Anyeliz B. Rico T.  
para optar al título de  
Ingeniera Electricista.

Caracas, 2016

## CONSTANCIA DE APROBACIÓN

Caracas, 08 de noviembre de 2016

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por la Bachiller Anyeliz Rico:

***“IDENTIFICACIÓN DE LA ÓPTIMA OPERACIÓN EN ESTRUCTURAS DE REDES DE COLAS”***

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniera Electricista en la mención de Electrónica Computación y Control, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

Prof. Dan El Montoya  
Jurado

Prof. Rafael Díaz  
Jurado

Dr. Ebert Brea  
Tutor Académico

*“A mis padres y a mi novio, quienes me enseñaron que un mar calmado nunca formó  
a un buen marinero”*

## **RECONOCIMIENTOS Y AGRADECIMIENTOS**

Principalmente a Dios y a mis Ángeles de Luz, cuya creencia me ha dado fe y esperanza en cada meta trazada, a pesar de las adversidades me dieron fuerzas para seguir luchando y poder cumplir mis metas.

A mi querida y amada Madre, agradeciendo a Dios porque te haya mantenido con nosotros apoyándonos, gracias por ayudarme a seguir adelante. A mi Padre, quien lucha cada día por darnos lo mejor, gracias por cambiar y convertirte en un gran amigo. A mi hermano, cuya filosofía de vida me ha hecho ver la vida desde otra perspectiva.

Un especial agradecimiento a mi novio Daniel León, quien siempre me apoyo y creyó en mi, ha estado presente en mi vida ayudándome en cada cosa que necesitaba, gracias por ser tan atento conmigo. A mis suegros Guillermo y Marleny, quienes me han enseñado tantas cosas valiosas, gracias por extender sus manos para ayudarme.

A mi admirado tutor, Profesor Ebert Brea, por darme la oportunidad de desarrollar este proyecto.

A mi gran UCV, de la cual me siento muy orgullosa y afortunada por crecer como profesional en esta casa de estudio.

Gracias a todos.

Rico T., Anyeliz B.

## Identificación de la Óptima Operación en Estructuras de Redes de Colas

**Tutor Académico: Dr. Ebert Brea. Tesis. Caracas, Universidad Central de Venezuela. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Mención Electrónica. T.E.G. 2016. 70h+anexos.**

**Palabras Claves:** MIRPSA, redes de colas, función objetivo, variables enteras mixtas, restricciones no lineales, canales de servicio serie, mixto y etapas retroalimentadas.

**Resumen.-** En este trabajo de grado se presenta la identificación de la óptima operación en estructuras de redes de cola con base al algoritmo denominado “Mixed Integer Randomized Pattern Search Algorithm” (MIRPSA), el cual trata acerca de la búsqueda de al menos un mínimo local mediante un patrón de prueba aleatoria para encontrar soluciones a problemas de optimización enteros mixtos enfocado a la Cadena de Markov. En la búsqueda de puntos aleatorios sus principales operaciones son: operación de contracción y operación de movimiento. Con el uso del mencionado algoritmo se realizó la interacción con el software de simulación SIMUL8 para identificar los valores óptimos de sistemas de redes de colas. Partiendo de que existen variables reales y enteras que conllevan a sistemas no lineales de variables enteras mixtas, la función objetivo a optimizar, siendo ésta la representación del desempeño del sistema, es el tiempo promedio de clientes dentro del sistema adicionando el número de clientes que salen del mismo sin ser atendidos, donde para poder emplear el MIRPSA se hizo uso de funciones de penalizaciones ya que la función objetivo está sujeta a restricciones no lineales. Se realizaron experimentos con canales de servicio en serie, paralelos, mixto y etapas retroalimentadas, donde el orden del servicio es FIFO (primero en entrar primero en salir) y las tasas de llegadas y de servicios por parte de los clientes siguen una distribución exponencial.

# ÍNDICE GENERAL

<b>CONSTANCIA DE APROBACIÓN</b>	iii
<b>DEDICATORIA</b>	iv
<b>RECONOCIMIENTOS Y AGRADECIMIENTOS</b>	v
<b>RESUMEN</b>	vi
<b>ÍNDICE GENERAL</b>	vii
<b>LISTA DE FIGURAS</b>	xii
<b>LISTA DE TABLAS</b>	xv
<b>LISTA DE ACRÓNIMOS</b>	xvi
<b>INTRODUCCIÓN</b>	1
<b>GENERALIDADES DEL PROYECTO</b>	4
<b>Planteamiento del Problema</b> . . . . .	4
<b>Objetivos</b> . . . . .	5
<b>Objetivo General</b> . . . . .	5
<b>Objetivos Específicos</b> . . . . .	5
<b>Justificación</b> . . . . .	6
<b>Condiciones del estudio</b> . . . . .	6
<b>Antecedentes del Estudio</b> . . . . .	7

<b>MARCO TEÓRICO</b>	<b>9</b>
<b>Procesos estocásticos</b> . . . . .	9
<b>Cadena de Markov</b> . . . . .	10
<b>Teoría de colas</b> . . . . .	12
<b>Notación</b> . . . . .	15
<b>Simulación</b> . . . . .	17
<b>Técnicas de reducción de la varianza</b> . . . . .	18
<b>Números aleatorios comunes</b> . . . . .	18
<b>Método antitético</b> . . . . .	18
<b>Media Muestral</b> . . . . .	20
<b>Optimización</b> . . . . .	22
<b>Tipos de Optimización</b> . . . . .	23
<b>MARCO METODOLÓGICO</b>	<b>26</b>
<b>DESCRIPCIÓN DEL MODELO</b>	<b>28</b>
<b>Diseño de optimización de sistemas de colas</b> . . . . .	28
<b>Detalles del Algoritmo</b> . . . . .	31
<b>Descripción del problema</b> . . . . .	34
<b>Interacción entre el MIRPSA y el software Simul8</b> . . . . .	36
<b>Inicialización del sistema</b> . . . . .	39
<b>PRUEBAS EXPERIMENTALES Y RESULTADOS</b>	<b>42</b>
<b>Diseño en Serie</b> . . . . .	45
<b>Diseño en Paralelo</b> . . . . .	46
<b>Diseño Mixto</b> . . . . .	48
<b>Diseño de Etapas Retroalimentadas</b> . . . . .	50



<b>ANÁLISIS DE LOS RESULTADOS</b>	54
<b>CONCLUSIONES</b>	57
<b>RECOMENDACIONES</b>	58
<b>REFERENCIAS</b>	59
<b>ANEXOS</b>	61
<b>Valores de <math>t_{\alpha}</math></b>	62
<b>Valores de <math>\chi^2</math></b>	63
<b>Parámetros iniciales</b>	64
<b>Valores de distribución y función de desempeño</b>	65
<b>Tabla de Resultados</b>	66
<b>Penalización I</b>	67
<b>Penalización II</b>	68
<b>Reset Logic I</b>	69
<b>Reset Logic II</b>	70
<b>Reset Logic III</b>	71
<b>Reset Logic IV</b>	72
<b>Valores actuales para la penalización</b>	73

<b>Búsqueda Aleatoria VZ I</b>	74
<b>Búsqueda Aleatoria VZ II</b>	75
<b>Búsqueda Aleatoria VZ III</b>	76
<b>Búsqueda Aleatoria VZ IV</b>	77
<b>Stop Run Logic</b>	78
<b>Time Check Logic I</b>	79
<b>Time Check Logic II</b>	80
<b>Time Check Logic III</b>	81
<b>Time Check Logic IV</b>	82
<b>Time Check Logic V</b>	83
<b>Time Check Logic VI</b>	84
<b>Actividad: Asesor I</b>	85
<b>Actividad: Asesor II</b>	86
<b>Modelo paralelo, mixto y retroalimentado</b>	87
<b>Cambios en la Función objetivo</b>	88
<b>Salida de clientes del sistema</b>	89
<b>Evaluación de las capacidades</b>	90

<b>Servidor 1 Logic</b>	<b>91</b>
<b>Servidor 2 Logic</b>	<b>92</b>
<b>Servidor 3 Logic</b>	<b>93</b>

## LISTA DE FIGURAS

2.1.	Funcionamiento principal de un sistema de colas	14
2.2.	Sistema multiservicio con una única cola de espera	15
2.3.	Sistema multiservicio con múltiples colas de espera	15
4.1.	Pseudocódigo de el MIRPSA	32
4.2.	Mensaje de Continuar	38
4.3.	Mensaje de Penalización	38
5.1.	Modelo en Serie	45
5.2.	Gráfica de la mejora de la función objetivo en el Modelo Serie	46
5.3.	Modelo en Paralelo	46
5.4.	Gráfica de la mejora de la función objetivo en el Modelo Paralelo	47
5.5.	Modelo Mixto	48
5.6.	Gráfica de la mejora de la función objetivo en el Modelo Mixto	49
5.7.	Modelo de Etapas Retroalimentadas	50
5.8.	Gráfica de la mejora de la función objetivo en el Modelo de Etapas Retroalimentadas	51
5.9.	Gráfica de la función objetivo del Modelo Serie con $\alpha_1 = 0,2$ y $\alpha_2 = 0,8$	52
5.10.	Gráfica de la función objetivo del Modelo Serie con $\alpha_1 = 0,7$ y $\alpha_2 = 0,3$	53
I.1.	Valores críticos de la distribución t	63
II.1.	Valores críticos de la distribución chi cuadrado	63
III.1.	Spreadsheets: Parámetros iniciales I	64

III.2.	Spreadsheets: Parámetros iniciales II	64
III.3.	Spreadsheets: Parámetros iniciales II	64
IV.1.	Spreadsheets: Función menor	65
IV.2.	Spreadsheets: Función	65
V.1.	Spreadsheets: Resultados I	66
V.2.	Spreadsheets: Resultados II	66
VI.1.	Visual Logic: Penalización I	67
VII.1.	Visual Logic: Penalización II	68
VIII.1.	Visual Logic: Reset I	69
IX.1.	Visual Logic: Reset II	70
X.1.	Visual Logic: Reset III	71
XI.1.	Visual Logic: Reset IV	72
XII.1.	Visual Logic: Valores Actuales	73
XIII.1.	Visual Logic: Valores de Vz I	74
XIV.1.	Visual Logic: Valores de Vz II	75
XV.1.	Visual Logic: Valores de Vz III	76
XVI.1.	Visual Logic: Valores de Vz IV	77
XVII.1.	Visual Logic: Stop Run	78
XVIII.1.	Visual Logic: Time Check Logic I	79

XIX.1. Visual Logic: Time Check Logic II	80
XX.1. Visual Logic: Time Check Logic III	81
XXI.1. Visual Logic: Time Check Logic IV	82
XXII.1. Visual Logic: Time Check Logic V	83
XXIII.1. Visual Logic: Time Check Logic VI	84
XXIV.1. Visual Logic: Asesor del modelo paralelo	85
XXV.1. Visual Logic: Asesor del modelo paralelo	86
XXVI.1. Visual Logic: Salida en el modelo paralelo, mixto y retroalimentado	87
XXVII.1. Visual Logic: Cambios en el Time Check para el modelo paralelo, mixto y retroalimentado	88
XXVII.2. Visual Logic: Inicialización de Valores de la Función	88
XXVIII.1. Visual Logic: Clientes no atendidos	89
XXIX.1. Visual Logic: Capacidad 1	90
XXIX.2. Visual Logic: Capacidad 2	90
XXIX.3. Visual Logic: Capacidad 3	90
XXX.1. Visual Logic: Servidor 1	91
XXXI.1. Visual Logic: Servidor 2	92
XXXII.1. Visual Logic: Servidor 3	93

## LISTA DE TABLAS

4.1. Parámetros Iniciales . . . . .	36
5.1. Parámetros Iniciales . . . . .	42
5.2. Parámetros Iniciales $x^n$ e $y^m$ . . . . .	43
5.3. Intervalos para la Penalización . . . . .	44
5.4. Resultados del Modelo Serie . . . . .	45
5.5. Resultados del modelo Paralelo . . . . .	47
5.6. Resultados del modelo Mixto . . . . .	48
5.7. Resultados del Modelo de Etapas Retroalimentadas . . . . .	50
5.8. Resultados del Modelo Serie con $\alpha_1 = 0,2$ y $\alpha_2 = 0,8$ . . . . .	52
5.9. Resultados del Modelo Serie con $\alpha_1 = 0,7$ y $\alpha_2 = 0,3$ . . . . .	53

## **LISTA DE ACRÓNIMOS**

FIFO: First in, first out.

LIFO: Last in, first out.

MIRPSA: Mixed Integer Randomized Pattern Search Algorithm.

RSS: Random Selection of Service.



# INTRODUCCIÓN

La mayoría de los problemas dentro de una red de comunicaciones, tal como el envío de paquetes en redes, o de igual forma en el comportamiento del tráfico, la espera en un banco, servicio técnico, entre otros, requiere de tiempo de servicio y de tiempo de llegada por parte de un conjunto de clientes. Este tipo de tráfico se modela por la teoría de colas, la cual se ocupa del análisis matemático de los fenómenos de las líneas de espera o colas. El origen de la teoría de colas viene dado por el matemático danés Agner Kraup Erlang, quien publicó en 1909 el primer artículo referente a teoría de colas para analizar la congestión de tráfico en la conmutación telefónica. La formación de colas ocurre debido a un desequilibrio entre los servidores, el número de clientes de llegada al sistema y el tiempo de servicio para cada cliente, y un máximo de clientes permitidos, por lo que puede ocurrir que el cliente prefiera abandonar la cola o rechazarla debido a la magnitud de la misma, este tipo de caso se conoce como caso de frustración. El orden que se utilizará para recibir el servicio es de tipo FIFO (primero en entrar primero en salir) y el estudio del modelo de tiempo discreto está asociado a un sistema de colas  $M/M/1$  y  $G/M/1$  que se considera en este caso un solo servidor atendiendo a un cliente, cuyas tasas de llegadas para el primer servicio siguen una distribución exponencial y una distribución general en la llegada de los clientes a los siguientes servidores.

Durante el progreso del proyecto se tomarán medidas para utilizar el canal de servicio más adecuado tales como canales en serie, paralelos o mixtos, etapas retroalimentadas como se encuentran en los sistemas de producción. Esto es realizado para que el cliente pueda ser atendido con el menor tiempo posible y de esta manera tener un sistema óptimo. Este tipo de sistemas utiliza el tiempo en medida real, por lo que se deben considerar tanto variables enteras para el número de servidores e individuos en el sistema como variables reales para los retardos de tiempo. Esto lo encontramos en

las agencias bancarias, en el que tenemos números de cajeros y su respectiva tasa de atención o de servicio de cada cajero. Debido a la complejidad de estos sistemas en la búsqueda de soluciones óptimas, se ha de realizar un modelo de simulación de redes de colas en conjunto con un algoritmo que identifique el mejor desempeño de la red de colas.

Brea (2005) presenta en su trabajo un estudio de la relación simulación optimización, en el cual muestra una breve explicación sobre su propuesta de modificar el método simplex de Nelder Mead (MSMN), cuando se tienen problemas bajos restricciones lineales. En esta propuesta, la eficacia para la identificación de óptimos a problemas bajo ruido ha sido comprobada de manera experimental.

Por otra parte, Brea (2013) propone una extensión del método de Nelder Mead a problemas de optimización enteros mixtos, a través de la incorporación de un conjunto de operaciones en el campo entero.

Adicionalmente, Brea (2015) desarrolló un nuevo algoritmo que permite llevar a cabo una búsqueda de al menos un mínimo local. Este tiene dos operaciones principales sobre lo que denomina el autor, patrón de búsqueda aleatorio: operación de movimiento y operación de encogimiento. Basándose en la localización de un conjunto de puntos de prueba aleatorio para la búsqueda de soluciones a los problemas de optimización entero mixtos, y en un enfoque por cadenas de Markov, Brea (2015) demuestra la convergencia de su método algorítmico.

Este proyecto está estructurado como sigue: el Capítulo 1 hace referencia a la finalidad del proyecto, sus objetivos, la justificación, las herramientas a utilizar y los antecedentes de estudio; en el Capítulo 2 se presenta el marco teórico basado en definiciones de teoría de colas, diseños de sistemas de colas Marcovianos, la notación de Kendall y los diferentes tipos de optimización; en el Capítulo 3 se hace referencia a la metodología del proyecto que se lleva a cabo mediante pasos para poder alcanzar

los objetivos planteados; en el Capítulo 4 se muestra la función objetivo a utilizar, el uso del algoritmo mostrado en Brea (2015), la descripción del problema y se detalla la programación en el software simul8; en el Capítulo 5 se realizan las pruebas experimentales en diseños en serie, paralelo, mixto y retroalimentado asociados a sistemas de colas M/M/1 mostrándose los resultados; en el Capítulo 6 se muestra el análisis de los distintos canales de servicios. En el Capítulo 7 se concluye respecto al algoritmo y los valores obtenidos de los experimentos. Finalmente en el Capítulo 8 se mencionan algunas recomendaciones para realizar una simulación efectiva.

# CAPÍTULO I

## GENERALIDADES DEL PROYECTO

### Planteamiento del Problema

Una herramienta que permite realizar un estudio detallado de la operación de los sistemas, desde la perspectiva de la teoría de colas, es el software de simulación de eventos discretos [SIMUL8](#) ([2013](#)). Éste es un producto de la SILMUL8 Corporation que nos permite realizar simulaciones del sistema involucrando entidades discretas en tiempo discreto. Mediante la simulación de los sistemas es posible modelar situaciones hipotéticas que nos permitan realizar la toma de decisiones transcendentales que impactan directamente en el desempeño de dicho sistema.

Para encontrar el diseño óptimo de redes de colas, se emplea en este trabajo el algoritmo diseñado por [Brea](#) ([2015](#)) y el software de simulación [SIMUL8](#) ([2013](#)). Este algoritmo permitirá identificar el óptimo del sistema, a través de la evaluación de su desempeño. Para ello se diseña un modelo inicial, el cual variará según el resultado de cada iteración realizada, con la intención de alcanzar el óptimo mediante un patrón de búsqueda aleatorio denominado “Mixed Integer Randomized Pattern Search Algorithm” (MIRPSA), el cual tiene dos operaciones principales: operación de movimiento y operación de contracción.

Partiendo de que existen variables reales y enteras, lo que conlleva a sistemas no

lineales de variables enteras mixtas, se presenta el siguiente problema matemático:

$$\underset{(x,y) \in \mathbb{R}^n \times \mathbb{Z}^m}{\text{minimize}} \bar{f}(x,y), \quad (1.1a)$$

sujeto a:

$$\begin{aligned} l^{(i)} \leq x^{(i)} \leq h^{(i)}, \quad \forall i \in \{1, 2, \dots, n\}; \\ l^{(j)} \leq y^{(j)} \leq h^{(j)}, \quad \forall j \in \{1, 2, \dots, m\}, \end{aligned} \quad (1.1b)$$

donde  $\bar{f}(x,y) : \mathbb{R}^n \times \mathbb{Z}^m \rightarrow \mathbb{R}$  es una función objetivo que representa el desempeño del sistema, la cual estará definida por medio de su valor promedio, las cantidades  $x \in \mathbb{R}^n$  e  $y \in \mathbb{Z}^m$  representan las variables de decisión del problema.

Esto implica plantear los problemas de optimización, a través de funciones de penalización, para así poder emplear el MIRPSA.

## Objetivos

### Objetivo General

Identificar la óptima operación en estructuras de redes de colas.

### Objetivos Específicos

1. Recopilar información acerca de la teoría de colas y sus aplicaciones.
2. Establecer estrategias para alcanzar la compatibilidad entre el algoritmo y el ambiente de simulación.
3. Implementar estructuras de redes de colas en el ambiente de simulación.
4. Codificar el algoritmo utilizando el ambiente visual de la simulación y realizar

experimentaciones con las redes diseñadas.

5. Verificar y analizar los resultados obtenidos de estos experimentos numéricos.
6. Concluir y realizar recomendaciones del algoritmo y de los experimentos diseñados.

## **Justificación**

En el análisis del comportamiento de las redes de colas es necesario realizar varias estructuras utilizando los métodos derivados de la teoría de colas y haciendo uso de un software de simulación que permita modelar sistemas en tiempo real, para lo cual se utilizará [SIMUL8 \(2013\)](#). El estudio de las redes de colas se concibe con la intención de reducir los tiempos de atención del cliente fomentando la rápida respuesta de los requerimientos.

Idealmente se persigue la premisa de que ningún cliente abandone la cola, por lo que mediante el algoritmo diseñado por [Brea \(2015\)](#), con notación genérica permita la interacción con el simulador y con las diferentes estructuras de colas, obteniendo valores numéricos que garanticen un sistema óptimo y eficaz.

## **Condiciones del estudio**

- Este proyecto estará orientado a la interacción del algoritmo MIRPSA con el ambiente simul8.
- Se obtendrá una sola muestra por simulación.
- En los modelos paralelo y mixto los clientes se dirigen hacia la cola mas corta.

## Antecedentes del Estudio

La demanda de servicios ha traído un crecimiento y un mejoramiento para las empresas, por lo que se ha tenido la necesidad de realizar estudios acerca de la teoría de colas la cual inicia en 1909 por el matemático danés Agner Kraup Erlang, en la que analizó el tráfico en los sistemas telefónicos, investigación que analiza la espera y los factores que intervienen en ella. Basándose en esos principios y realizando otros estudios referentes a una red, tales como las redes abiertas y cerradas, concluyó bajo demostraciones matemáticas que es posible analizar la red mediante sistemas de espera independientes de la tasa de llegada. Éste ha tenido un impacto en la teoría de colas. Trabajos realizados por [Gong, Lai, y Wang \(2008\)](#) utilizan una red cerrada de Jackson para explicar algunas propiedades de una red de cadena de suministro y determinar el número de servidores y estaciones para hallar el equilibrio del sistema y no formar cuellos de botella.

En el transcurso de los años, buscando el perfeccionamiento y con el uso de los computadores se han encontrados métodos que permiten obtener resultados óptimos. Muchos investigadores en busca de minimizar los tiempos y los costos de procesamiento han aplicado modelos de optimización no lineal, en este caso tenemos a [Berman, Larson, y Chiu \(1985\)](#) cuyas demandas de servicios surgen en los nodos de un red. Otros han realizado algoritmos para llegar a ello, en este caso [Feldman, Mandelbaum, Massey, y Whitt \(2008\)](#) en el cual determinan los niveles de atención con tasas de llegada variables en el tiempo; Proyecto de grado de [Guillermo \(2008\)](#) que mediante el uso de algoritmos y aplicando técnicas sobre diferentes modelos de sistema de colas M/M/1 (Markovianos) estudia sus diferentes aproximaciones. Es importante destacar que dentro de la investigación de [Guillermo \(2008\)](#) se estudia la convergencia de algunos algoritmos de optimización basados en los resultados obtenidos por [Kumin \(1969\)](#); [Dora y Jorge \(2006\)](#) realizaron evaluaciones sobre el sistema en el área de Consulta Externa del Hospital de pequeñas especies en la cual se emplearon las teorías de colas

y evaluaciones sobre la optimización del sistema, encontrando un punto de equilibrio entre la tasa de llegada y las tasas de servicios. El principal objetivo de los estudios mencionado anteriormente, es reducir el tiempo de servicio y de espera en varias aplicaciones, empleando métodos que permitan la optimización del sistema, adicionando posibles simulaciones para la visualización en tiempo real y de esta manera resolver los distintos problemas que puedan ocurrir en el transcurso del tiempo, colocando bajo análisis el comportamiento y el desempeño de las redes de colas.



## CAPÍTULO II

### MARCO TEÓRICO

#### Procesos estocásticos

Los procesos estocásticos son una sucesión de observaciones  $X_1, X_2, \dots$  donde los valores de estas observaciones no se pueden predecir exactamente. Sin embargo, se pueden especificar las probabilidades para los distintos valores posibles en cualquier instante de tiempo.

Se tiene un sistema que cambia su estado a lo largo del tiempo, siendo  $X_t$  el estado del sistema al tiempo  $t$ . Si se considera que el sistema no es determinista en el momento de su evolución, sino que se comporta como un fenómeno de naturaleza aleatoria entonces puede considerarse que  $X_t$  es una variable aleatoria para cada tiempo  $t$  cambiando probabilísticamente de estado. Sería lo mismo decir, al realizar una serie de pruebas bajo idénticas condiciones se obtiene resultados diferentes a lo largo del tiempo  $t$ . Este conjunto o colección de variables aleatorias es lo que define el proceso estocástico, describiendo el comportamiento en operación durante algunos periodos. Las variables aleatorias no son independiente entre sí, sino que están ciertamente relacionadas unas con otras.

Los valores posibles que toma la variable aleatoria se denominan estados, por lo que se puede obtener un espacio de estados discretos y un espacio de estados continuo, de igual forma la variable tiempo puede ser de tipo continuo o de tipo discreto.

Se supone que el subíndice  $t$  toma valores de un conjunto  $T$ , llamado rango de tiempo, siendo  $T$  el subconjunto de  $(-\infty, +\infty)$  y la variable  $X_t$  es la observación en el tiempo  $t$ . Si  $T$  es discreto, entonces  $T = \{0, 1, 2, 3, \dots\}$  se dice que es un proceso estocástico de tiempo discreto denotado por  $\{X_n : n = 0, 1, 2, \dots\}$  para  $n \in \mathbb{Z}$ . Si  $T$  es continuo, entonces  $T = \{t \geq 0\}$  siendo un proceso estocástico de tiempo continuo.

$X_0$  es la variable aleatoria que define el estado inicial del proceso y  $X_n$  es la variable aleatoria que define el estado del proceso en el instante de tiempo  $n$ .

Para posibles valores del estado inicial  $s_1$  y para cada valor sucesivo de  $s_n$  de los estados  $X_n$ ,  $n = 1, 2, 3, \dots$  se tiene:

$$P(X_{n+1} = s_{n+1} \mid X_0 = s_0, X_1 = s_1, X_2 = s_2, \dots, X_n = s_n) \quad (2.1)$$

### **Cadena de Markov**

Una Cadena es un proceso estocástico con tiempo discreto  $\{X_n : n = 0, 1, 2, 3, \dots\}$  para  $n \in \mathbb{Z}$  que ocurre de manera probabilista. Cuando  $X_n$  representa la magnitud discreta tal como la cantidad de clientes en un sistema de atención o de espera se hablará de un proceso de Markov con estados discretos o Cadena de Markov. Las Cadenas de Markov no guardan su historia pasada en la memoria, siendo independiente de la sucesión de todas las situaciones anteriores. Para ello se debe satisfacer la propiedad de Markov:

$$P(X_{n+1} = \mathcal{X}_{n+1} \mid X_0 = \mathcal{X}_0, \dots, X_n = \mathcal{X}_n) = P(X_{n+1} = \mathcal{X}_{n+1} \mid X_n = \mathcal{X}_n) \quad (2.2)$$

Todas aquellas cadenas que cumplen con la propiedad de Markov se llaman Ca-

denas de Markov. Una cadena de Markov se pueden escribir como el producto de una distribución de probabilidad inicial ( $X_0$ ) y las probabilidades de pasar de un estado a otro en dos tiempos sucesivos.

La distribución inicial de la cadena de Markov es:

$$P(X_0 = K); \quad \forall k \quad (2.3)$$

Las probabilidades condicionadas de una cadena de Markov son llamadas probabilidades de transición del estado  $i$  en el tiempo  $n$  al estado  $j$  en el tiempo  $n + 1$  también conocidas como probabilidades de transición de un paso:

$$P(X_{n+1} = j | X_n = i) = P_{ij}(n, n+1); \quad i, j \in \mathbb{Z} \quad (2.4)$$

Donde  $i$  es el estado origen y  $j$  el estado destino. Si estas probabilidades se mantiene constante para cualquier valor de  $n$  se dice que es estacionaria, homogénea en el tiempo. Entonces, el tener probabilidades de transición estacionarias implica que no cambian en el tiempo.

$$P_{ij}(n, n+1) = P_{ij}(0, 1) = P_{ij}(1) = P_{ij} \quad (2.5)$$

Considerando los distintos valores de  $i$  y  $j$  en el espacio de estados se obtienen la matriz de probabilidades de transición en un paso. La matriz de probabilidades de

transición en un paso para una cadena de Markov (estacionaria) es:

$$P = \begin{pmatrix} p_{00} & p_{01} & \cdots & p_{0k} \\ p_{10} & p_{11} & \cdots & p_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k0} & p_{k1} & \cdots & p_{kk} \end{pmatrix} \quad (2.6)$$

Basándose en el concepto de  $p_{ij}$  las filas de la matriz suman la unidad,

$$\sum_j p_{ij} = 1, \forall i \quad (2.7)$$

La matriz  $P = (p_{ij})$  de la Ecuación 2.6 es conocida como matriz estocástica de tamaño  $k \times k$  donde  $k \in \mathbb{Z}$  y  $p_{ij}$  definen un conjunto de probabilidades de transición, la cual se interpreta como una Cadena de Markov estacionaria con  $k$  estados. La representación matricial facilita el calculo de las probabilidades de transición en más de un paso. De igual manera se conoce que  $P \times P = P^2$  corresponde a las probabilidades de transición en dos pasos y  $P^3, P^4, P^5, \dots, P^m$  corresponden respectivamente a las probabilidades de transición en 3, 4, 5, ...,  $m$  pasos. Entonces,  $P^m$  representa la matriz de transición en  $m$  pasos de la Cadena de Markov.

### Teoría de colas

La teoría de colas en su primera aplicación se da a conocer por el matemático danés Agner Kraup Erlang, quien analizó la congestión de tráfico en conversaciones telefónicas en 1909, para el cálculo de tamaño de centrales telefónicas.

La teoría de colas es el estudio de la espera en cualquier sistema donde haya una cola, ocupándose del análisis matemático de los fenómenos de las líneas de esperas o colas.

El uso de los modelos y las fórmulas que representen el desempeño del sistema permite tener la visualización de la cantidad de clientes en una cola así como el tiempo de espera y el tiempo de servicio para ser atendidos.

La línea de espera está constituida por clientes que esperan para ser atendidos por un servidor en un determinado tiempo. Los clientes ingresan al sistema de manera aleatoria mediante una distribución probabilística, en este caso exponencial. Dependiendo de las disciplinas del servicio se proporciona la atención a los clientes de la cola en un periodo de tiempo y luego abandonan el sistema. Estas líneas de espera poseen las siguientes características:

1. Número de clientes que pueden permanecer en la cola.
2. Tiempo entre dos llegadas de clientes sucesivos al sistema.
3. Tiempo que proporciona el servicio para atender al cliente (tiempo de servicio).
4. Disciplina del servicio (FIFO, LIFO, RSS, entre otras)
5. Cantidad de servidores.
6. Estructura del modelo.

Desde el momento que ingresa un cliente al sistema y el servidor está disponible, el cliente es atendido. Si el servidor se encuentra en operaciones, el cliente debe aguardar en cola hasta recibir el servicio, en el caso que la cola se encuentre en su máxima capacidad los clientes tienden a rechazar o abandonar el sistema sin recibir el servicio. La fuente de entrada determina el número total de clientes que requerirán el servicio.

Se tiene un patrón de llegada de los clientes de manera estocástica, es decir la llegada depende de una cierta variable aleatoria siendo necesario conocer la distribución

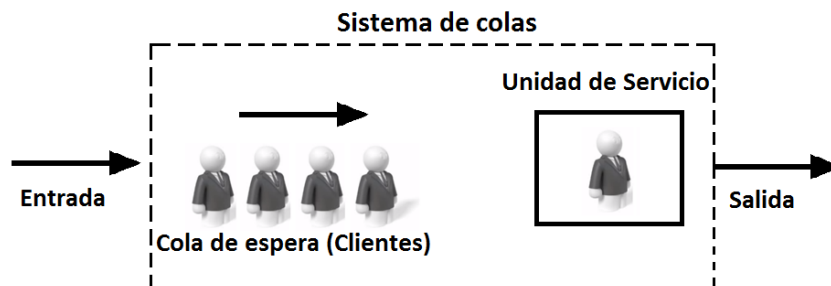
probabilística entre dos llegadas de clientes sucesivos. En este caso los clientes llegan de manera independiente y su distribución de probabilidad es exponencial.

La disciplina de la cola representa el orden en el cual son seleccionados los clientes en la cola para ser atendidos o recibir el servicio. Entre las clasificaciones de las disciplinas tenemos: FIFO (primero en entrar primero en salir), LIFO (último en entrar primero en salir) y RSS (selección aleatoria del servicio).

El canal de servicio es el proceso en el cual se efectúa el servicio para cada cliente, el mismo puede ser un canal en serie, paralelo, mixto o etapas retroalimentadas. Dependiendo del canal de servicio los clientes podrán ser atendidos de manera simultánea o individual.

La cantidad de clientes permitidos en una cola dependen de las características del sistema, de acuerdo a esto el sistema tendrá una cola finita o infinita. Si la cola es infinita todos los clientes que llegan al sistema podrán ser atendidos o por lo menos tendrán la oportunidad de entrar a formar parte del sistema; mientras si es finita hay un límite máximo y cuando se encuentre llena la cola los clientes serán rechazados.

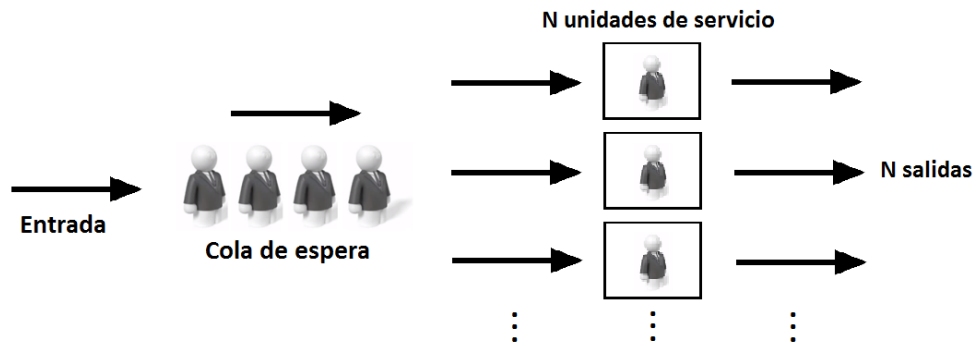
Los elementos que componen a un sistema de colas son representados en la Figura 2.1:



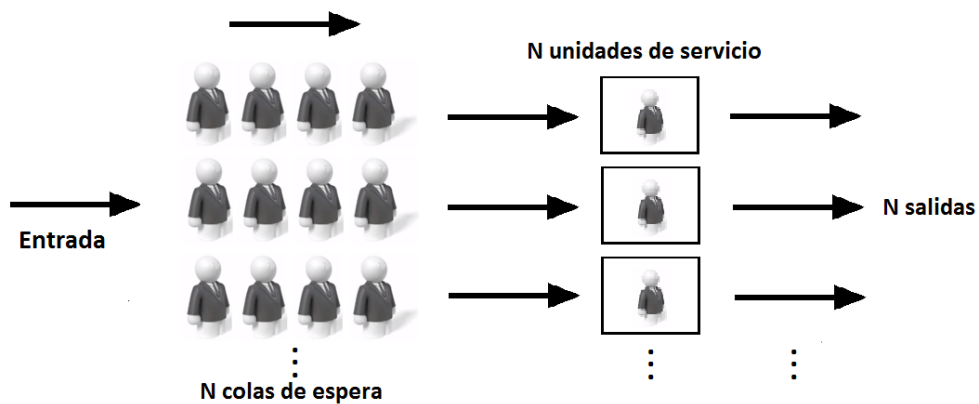
**Figura 2.1.** Funcionamiento principal de un sistema de colas

Se presenta en la Figura 2.2 un sistema multicanal con una única cola de espera

y múltiples servidores en paralelo. Mientras que en la Figura 2.3 se tienen diferentes colas independientes con sus servidores correspondientes. En este trabajo se realizarán diversos sistemas tal como el sistema en serie, paralelo y retroalimentado para observar su comportamiento.



**Figura 2.2.** Sistema multiservicio con una única cola de espera



**Figura 2.3.** Sistema multiservicio con múltiples colas de espera

Cuando los tiempo de llegadas y de servicio dentro de un sistema de colas tiene distribución exponencial, el mismo es llamado sistemas de Colas Markoviano.

### Notación

David G. Kendall elaboró en 1953 una notación que permite describir las colas y de esta manera clasificar los distintos tipos de colas. Se tiene que se determina de la

siguiente manera :

$$A/B/X/Y/Z/V$$

donde

- $A$  describe la distribución de probabilidad del tiempo entre llegadas.
  - $M$  tiempo entre llegadas exponencial.
  - $D$  tiempo entre llegadas determinístico.
  - $G$  representa la distribución general de los tiempos entre llegadas.
- $B$  representa el tipo de distribución de probabilidad para el tiempo de servicio.
  - $M$  tiempo de servicio con distribución exponencial.
  - $D$  representa un tipo de distribución determinístico.
  - $G$  el tiempo de servicio está regido por una distribución general.
- $X$  número de servidores.
- $Y$  capacidad de clientes permitidos incluyendo los actuales en servicio, siendo rechazados los siguientes clientes. En el modelo de Erlang-B ( $M/M/N/N$ ) no hay cola de espera sino que hay  $n$  servidores y  $n$  usuarios, si llega el usuario  $n + 1$  será rechazado; en el modelo de Erlang-c ( $M/M/n$ ) la capacidad del sistema es ilimitada, con  $n$  servidores y  $n + 1$  clientes.
- $Z$  representa el tipo de servicio (FIFO, LIFO, RSS, entre otras).
- $V$  representa el tamaño de la población de posibles usuarios.

El sistema a utilizar en este proyecto es  $M/M/1/\infty/FIFO/\infty$ , la cual significa que los tiempos entre llegadas y de atención al cliente siguen una distribución exponencial, tiene un solo servidor atendiendo a un solo cliente, la capacidad del sistema es infinita, la disciplina del servicio es FIFO y el número de posibles usuarios es infinito.



## **Simulación**

Se habla por primera vez de simulación en el año 1949 cuando John Von Neumann y Stanislaw Ulam presentaron el método de Monte Carlo. En el transcurso del tiempo la simulación ha sufrido un crecimiento debido al desarrollo computacional. La simulación es el desarrollo de un modelo lógico-matemático de un sistema, con ello se logra describir el comportamiento y la estructura del sistema, obteniendo una imitación de la operación de un proceso en la vida real o mediante un computador a través del tiempo. La simulación se puede definir también como una técnica consistente en la realización de experimentos de muestreo sobre un modelo construido a partir de un sistema real.

Para el desarrollo de un modelo de simulación se debe construir las ecuaciones representativas del sistema y anexarla a un programa computacional, dicho modelo es una representación de los objetos del sistema.

La simulación puede llegar a ser estática o dinámica según la evolución en el tiempo. Las estáticas representan un sistema en un instante particular. A este tipo de simulación se le denomina simulación de Monte Carlo, se usa frecuentemente en estadísticas para comprobar el comportamiento de diferentes métodos con tamaños muestrales finitos. Las dinámicas representan un sistema que evoluciona en el tiempo y se tratan de analizar los diversos estados ocurridos en el sistema. La simulación dinámica según las variables de estado puede ser continua o discreta: la simulación continua es cuando el sistema cambia constantemente de estado o las variables de estado cambian de forma continua con el tiempo y la simulación discreta si las variables de estado cambian solo en determinados instantes de tiempo.

El sistema utilizado en este trabajo según las variables de estado es discreto, según la evolución en el tiempo es dinámico y su aleatoriedad es probabilista o estocástica.

## Técnicas de reducción de la varianza

Cuando se pretende ofrecer respuestas lo más preciso posible sobre cantidades medias, es necesario utilizar técnicas para reducir la varianza o tratar de estimarla en un estudio de simulación. Los métodos mas utilizados son: los números aleatorios comunes y la utilización de las variables antitéticas.

### Números aleatorios comunes

Se desea comparar dos estrategias distintas mediante  $N$  repeticiones, esta técnica se usa para determinar diferencias entre resultados:  $X_1, X_2, X_3, \dots, X_N$  para la primera e  $Y_1, Y_2, Y_3, \dots, Y_N$  para la segunda, bajo las mismas condiciones aleatorias, es decir utilizando la misma semilla. Se desea estimar la diferencia de las medias de las variables de salidas:  $E(X) - E(Y) = E(X - Y)$ , puede usarse  $\bar{X} - \bar{Y}$ , siendo esto:

$$E(\bar{X}) - E(\bar{Y}) = E(\bar{X} - \bar{Y}) \quad (2.8a)$$

$$Var(\bar{X} - \bar{Y}) = Var(\bar{X}) + Var(\bar{Y}) - 2Cov(\bar{X}, \bar{Y}) \quad (2.8b)$$

siendo  $\bar{X}$  e  $\bar{Y}$  el valor de las medias, es decir  $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$  e  $\bar{Y} = \frac{1}{N} \sum_{i=1}^N Y_i$ . Si  $\bar{X}$  e  $\bar{Y}$  son independientes, entonces  $Cov(\bar{X}, \bar{Y}) = 0$ . Siempre que se usen números aleatorios diferentes para cada distribución de los experimentos no se introduce la correlación entre ellos. En simulaciones complejas donde cambia la secuencia de números, los efectos en los resultados pueden ser impredecibles.

### Método antitético

Este se basa en la evaluación del resultado de una única estrategia. Se tendrán después de  $N$  corridas valores numéricos como:  $X_1, X_2, X_3, \dots, X_N$ , luego se procede a hallar la media  $E(X)$  teórica mediante:  $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$ . Si las variables son independientes se

tiene que  $Var(\bar{X}) = \frac{1}{N^2} \sum_{i=1}^N Var(X_i)$ , mientras que si no son independientes:

$$V(\bar{x}) = \frac{1}{N^2} \left( \sum_{i=1}^N Var(X_i) + 2 \sum_{i,j=1, i < j}^N Cov(X_i, X_j) \right) \quad (2.9)$$

La expresión 2.9 se utiliza para reducir la varianza, haciendo que cada variable con índice impar sea negativamente correlacionada con la variable de índice siguiente. Para ello, se utiliza el método de inversión para simular  $X_i$  tomando un valor  $U \sim U(0, 1)$  para  $X_{2i-1}$  y su complementario  $1 - U$  para simular  $X_{2i}$ , para  $i = 1, 2, \dots, \frac{N}{2}$  con  $N$  par.

El simulador a utilizar será [SIMUL8](#) (2013):

Simul8 es una herramienta de software para simular o modelar procesos en tiempo real, que permite documentar complejidades de un sistema. Este tipo de simulador identifica respuestas futuras del sistema y nos ayuda a comprender lo que ocurre dentro del mismo e identificar las posibilidades de mejora. De igual manera mediante gráficos, valores estadísticos se observa el comportamiento del sistema.

Para realizar el estudio de un problema de simulación se debe obtener de ella una gran cantidad de muestras que permita llegar a valores esperados y conocer de ella algunos datos estadísticos. La teoría de muestreo relaciona una población y las muestras extraídas, a partir del conocimiento con base en las características muestrales. La teoría de muestreo también sirve para determinar si las diferencias observadas entre dos muestras se deben a variaciones al azar o si son significativas. Esto se encuentra al decidir si un proceso de producción o de atención en establecimientos es mejor que otro. Al considerar las muestras posibles de tamaño  $n$  que se puede obtener de una población, en cada muestra se suele calcular pruebas estadísticas como la media o la desviación estándar que varían de una muestra a otra.

## Media Muestral

Dado un conjunto de  $n$  medidas  $x_1, x_2, x_3, \dots, x_n$  se puede describir su centro o lugar central, esto se obtiene de la siguiente manera:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.10)$$

Cuando el conjunto de observaciones es de muestra aleatoria recibe el nombre de media muestral.

Una de las características más importantes de un conjunto de datos es la varianza entre esos valores. Las medidas de la media o la mediana que también es el calculo del promedio de un conjunto de datos, no aporta nada a la dispersión de un conjunto de datos sea reducida o ampliada alrededor de la media. Siendo razonable el estudio de la variación de un conjunto de números en términos de los montos por los cuales los valores se desvían de la media. Si el conjunto de números  $x_1, x_2, x_3, \dots, x_n$  tiene una media de  $\bar{x}$ , las diferencias  $x_1 - \bar{x}, x_2 - \bar{x}, x_3 - \bar{x}, \dots, x_n - \bar{x}$  se llaman desviación de la media, pero se sabe que la suma de las desviaciones es siempre igual a cero, esto es  $\sum_{i=1}^n x_i - \bar{x}$ , de modo que la media de las desviaciones siempre es cero.

Por ello, se eleva al cuadrado cada desviación. La varianza de la muestra suele usarse como  $s^2$  o  $\sigma^2$  y se define con la siguiente fórmula:

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2.11)$$

La desviación estándar de  $n$  observaciones  $x_1, x_2, x_3, \dots, x_n$  es:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.12)$$

Ahora, se quiere conocer con intervalos que contendrán el parámetro en consideración. Para ilustrar la formación de tal intervalo, se supone que se dispone de una muestra aleatoria  $n$  grande de una población con media y varianza conocidas, con probabilidad  $1 - \alpha$  se tiene:

$$\bar{x} - z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} < \mu < \bar{x} + z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} \quad (2.13)$$

De esta manera, cuando se ha obtenido una muestra y se ha calculado el valor de la media  $\bar{x}$ , se puede sostener con  $(1 - \alpha)100\%$  de confianza que el intervalo de  $\bar{x} - z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}$  a  $\bar{x} + z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}$  contiene a  $\mu$ , donde  $z_{\alpha/2}$  representa el valor crítico de la distribución chi-cuadrado.

Cuando se utilizan muestras pequeñas, la cual suele ser  $n < 30$ , se muestrea a partir de una población normal y es representado mediante la siguiente fórmula:

$$\bar{x} - t_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} < \mu < \bar{x} + t_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} \quad (2.14)$$

donde  $t_{\alpha/2}$  es el valor crítico de la distribución de  $t$  de Student con  $\nu = n - 1$  grados de libertad.

Cuando nos enfrentamos a decisiones sobre el tamaño relativo de las medias de dos o más poblaciones los intervalos de confianza son:

$$\bar{x}_1 - \bar{x}_2 \pm t_{\alpha/2} \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \sqrt{\frac{n_1 + n_2}{n_1 n_2}} \quad (2.15)$$

donde  $t_{\alpha/2}$  se basa en  $v = n_1 + n_2 - 2$  grados de libertad.

Para intervalos de confianza de muestras grandes se tiene:

$$\bar{x}_1 - \bar{x}_2 \pm z_{\alpha/2} \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \quad (2.16)$$

En este proyecto se obtendrá una sola muestra debido al consumo computacional. Por lo tanto  $\mu$  será el único valor obtenido de la muestra.

## Optimización

Son modelos matemáticos que buscan obtener una solución para la toma de decisiones, con el objetivo de mejorar el rendimiento del sistema minimizando o maximizando el rendimiento o la eficiencia. Técnicamente es un patrón de búsqueda que permite con probabilidades altas y bajas determinar el mejor valor, cuando este proceso es guiado gradualmente hacia un estado específico, a través de la búsqueda de objetivos específicos, se puede decir que prefiere ese estado. Este tipo de problemas en la búsqueda de un sistema óptimo que puede ser caracterizado como procesos de optimización estáticos o dinámicos cuyo resultado es el comportamiento de las probabilidades del estado, permitiendo un mejor sistema para optimizar a largo plazo el costo o la utilidad. La configuración de un sistema estático establece que las características del sistema no varían con el tiempo. En cambio los modelos dinámicos sus características operativas pueden cambiar en el tiempo.

En el caso de un individuo que desea maximizar su utilidad sujeto a restricciones de ingresos, es posible obtener la función de demanda por algún bien de interés para el consumidor pero solamente para un periodo de tiempo por lo que la variable temporal no es relevante.

## Tipos de Optimización

Los diferentes tipos de optimización son:

- **Locales:** este se realiza sobre los módulos de programas a través de funciones, métodos, entre otros. Se utiliza cuando se requiere de rapidez y confiabilidad mediante un conjunto de instrucciones, siendo el más rápido por tener un espacio de soluciones más pequeño.
- **Bucles:** se trata de ciclos que forman parte del rendimiento del programa, ya que realizan acciones repetitivas. La mayoría de este tipo de optimización trata de encontrar elementos que no deben repetirse en un ciclo.
- **Globales:** este tipo de optimización es lenta sin embargo mejora el desempeño del programa. Se basa en todo código y se puede trabajar con variables globales que agilicen el sistema pero consume más memoria todo depende de la arquitectura de la máquina.
- **De Mirilla:** trata de estructurar de manera eficiente el flujo del programa sobre todo en instrucciones de bifurcación como son las decisiones, ciclos y saltos de rutinas.

Dependiendo de lo requerido o las situaciones de complejidad que forma un sistema son:

- **Programación lineal:** este tipo de programación para la resolución de problemas utiliza las ecuaciones lineales dando a conocer la función objetivo con restricciones lineales.
- **Programación entera:** se basa en variables que deben ser valores enteros cuyo procedimiento o algoritmo es igual a la lineal.

- Programación dinámica: en la programación dinámica se toman decisiones en etapas sucesivas. Estas decisiones condicionan el futuro del sistema afectando las decisiones planteadas en ese momento.
- Metaheurísticas: se usan parámetros dados por el usuario sobre unos procedimientos genéricos y abstractos de una manera que se espera sea eficiente. Este es un procedimiento para resolver problemas de optimización mediante una aproximación intuitiva. Comparando los métodos exactos que proporcionan una solución óptima, los métodos heurísticos se limitan a proporcionar la mejor solución quizás no óptima. Sin embargo, en el método exacto se vuelve tedioso encontrar la solución óptima por lo que el orden de magnitud puede ser muy superior al del heurístico.

Existen diversos métodos heurísticos de naturaleza muy diferentes y diseñados para problemas específicos. A continuación se mencionaran algunos métodos heurísticos mas comunes:

- Métodos de Descomposición: este método trata de descomponer el problema en subproblemas mas sencillos de resolver, sin olvidar que todo forma parte del mismo problema.
- Métodos Inductivos: en estos métodos se pretende generalizar de versiones pequeñas o sencillas al caso completo, es decir se obtienen conclusiones generales a partir de premisas particulares.
- Métodos de Reducción: se utilizan para identificar propiedades en la que mayormente se obtienen buenas soluciones, para luego ser introducidas como restricción del problema. De esta manera se restringe y se reduce el espacio de soluciones del problema. La desventaja de este método es que puede dejar fuera las soluciones óptimas del problema original.
- Métodos Constructivos: usualmente son deterministas y están basados en la mejor elección en cada iteración, la cual consiste en construir paso a paso



la solución del problema.

- Métodos de Búsqueda Local: la búsqueda local inicia con una solución del problema y la mejora progresivamente, realizando operaciones de movimiento hasta mejorar la solución anterior.

## CAPÍTULO III

### MARCO METODOLÓGICO

La metodología empleada para alcanzar los objetivos propuestos en este proyecto es:

1. Recopilación de información y documentación relacionada con teoría de colas, sus aplicaciones y notaciones, así como analizar los diferentes tipos de optimización para sustentar teóricamente el funcionamiento del MIRPSA mostrado en [Brea \(2015\)](#) para la búsqueda de al menos un mínimo local.
2. Documentación acerca del software de simulación [SIMUL8 \(2013\)](#), para familiarizarse con el ambiente de simulación y lograr implementar mediante códigos las redes de colas. Para ello, se realizó la búsqueda de vídeos, proyectos e información otorgado por [SIMUL8 \(2013\)](#) relacionadas con simulación de sistemas. De igual manera, se documentó acerca de la estructuración del software para especificar en que momento ocurre una lógica y poder acceder en el momento determinado a la programación de los objetos. Esto se realiza mediante el uso del visual del Time Check que permite llevar el control del tiempo de simulación, para realizar las evaluaciones de cada cambio de distribución mediante el patrón de búsqueda aleatorio y obtener el valor de la función objetivo. El visual del Stop Run permitió llevar el control del criterio de parada descrito en el MIRPSA, basado en la tolerancia y los parámetros iniciales de una distribución continua y discreta.

3. Se implemento la codificación del MIRPSA al ambiente lógico del software. Utilizando los códigos y diseños visuales de SIMUL8 se realizó una estructura de redes de colas con los distintos canales de servicio tales como serie, paralelo, mixto y etapas retroalimentadas, requiriendo del estudio de teorías de colas.
4. Con las redes de colas diseñadas se realizaron experimentos para identificar la estructura de redes de colas óptima mediante el uso del algoritmo. Se inicializán los parámetros del sistemas para que el mismo indique los valores numéricos de cada variable a medida que ocurre el proceso de simulación. Con el uso de la Ecuación 3.1a, la cual representa la función de desempeño, se inicia con el estudio en los diferentes modelos de redes de colas.

$$\underset{(x,y) \in \mathbb{R}^n \times \mathbb{Z}^m}{\text{minimize}} \alpha_1 \bar{f}(x,y) + \alpha_2 \eta(x,y), \quad (3.1a)$$

donde:

$$\alpha_1, \alpha_2 > 0 \quad (3.1b)$$

$$\alpha_1 + \alpha_2 = 1$$

siendo  $\alpha_1$  y  $\alpha_2$  los factores multiplicativos de importancia. En el modelo serie se estudia el comportamiento del sistema al variar los factores de importancia  $\alpha_1$  y  $\alpha_2$ .

5. Al finalizar la simulación de cada estructura de redes de colas mediante la inicialización de las variables antes mencionadas e interactuándolo con el algoritmo se tomaron los valores numéricos y se analizaron dichos resultados para indicar el diseño de redes de colas óptimo.
6. Se realiza un resumen de los resultados para concluir con aportes significativos y analizar los distintos canales de servicio para obtener un sistema óptimo y que el cliente sea atendido en el menor tiempo posible.

## CAPÍTULO IV

### DESCRIPCIÓN DEL MODELO

Los sistemas de colas pueden ser muy diversos de acuerdo al tipo de sistema necesario por lo que puede tener varios servidores o etapas de servicio de igual manera diferentes colas. Utilizando distintos canales de servicios se obtendrá mediante la búsqueda de al menos un mínimo local el diseño óptimo, seleccionando los mejores puntos que permitan que el cliente esté en el sistema el menor tiempo posible. Los diseños de optimización se basan en una función objetivo que permite minimizar o maximizar las variables del modelo. En este proyecto se tratará de minimizar el tiempo promedio de atención al cliente, el cual tiene efecto sobre la cantidad de clientes no atendidos que abandonan el sistema sujeto a restricciones no lineales.

#### **Diseño de optimización de sistemas de colas**

La necesidad de obtener expresiones para valores esperados a través del sistema para la optimización, dadas las limitaciones o restricciones con variables que necesitan ser minimizadas o maximizadas, ha llevado a la utilización de funciones objetivo. A partir de esas expresiones se crea una función objetivo siendo ésta una ecuación matemática, para luego ser optimizada. De acuerdo a los valores requeridos o el objetivo del problema se crea dicha función.

La función objetivo para minimizar (optimizar) dentro de este trabajo es función

del tiempo promedio del cliente dentro del sistema. Para ello se tiene que:

$$FO : \underset{(x,y) \in \mathbb{R}^n \times \mathbb{Z}^m}{\text{minimize}} = \bar{f}(x,y) + \eta(x,y) \quad (4.1a)$$

donde:

$$\bar{f}(x,y) = \bar{m}_{c+1} = \frac{1}{c+1} (c \bar{m}_c + m_{c+1}) \quad (4.1b)$$

siendo  $\bar{f}(x,y)$  la función de tiempo promedio de atención al cliente en el sistema;  $\eta(x,y)$  es el número de clientes no atendidos por los servidores;  $c \in \mathbb{Z}$  representa el número de clientes en el sistema;  $m_{c+1}$  es el tiempo que tarda el siguiente cliente dentro del sistema y  $\bar{m}_c$  es el tiempo promedio de los  $c$  primeros clientes. Ahora, con  $k \in \mathbb{Z}$  se tiene:

$$\bar{m}_c = \frac{1}{c} \sum_{k=1}^c m_k \quad (4.2)$$

La función objetivo está sujeta a las siguientes restricciones no lineales:

$$\begin{aligned} l^{(i)} \leq x^{(i)} \leq h^{(i)}, \quad \forall i \in \{1, 2, \dots, n\}; \\ l^{(j)} \leq y^{(j)} \leq h^{(j)}, \quad \forall j \in \{1, 2, \dots, m\}, \end{aligned} \quad (4.3)$$

Como el MIRPSA considera restricciones no lineales, se emplean funciones de penalización, la cual tiene como factor multiplicativo una constante llamada en este trabajo como  $k_{m\acute{a}x}$ , alejándonos del valor óptimo. En la Sección 4 se explicará con más detalles su uso.

Uno de los principales objetivos del diseño consiste en hallar el tiempo que tarda el cliente dentro del sistema y de esta manera minimizar, con el uso de la función

objetivo, mostrada en la Ecuación 4.1a, el tiempo promedio de los clientes por ser atendidos en los distintos servicios de igual manera se reduce la tasa de atención de cada cliente. Esto se logra con el uso del MIRPSA que permite realizar la búsqueda de al menos un mínimo local mediante un patrón de búsqueda aleatorio. Basándose en números reales y enteros con algunas operaciones principales, al igual que la utilización de parámetros iniciales para realizar la búsqueda local cuyo dominio y rango están dadas por  $\mathbb{R}^n \times \mathbb{Z}^m \rightarrow \mathbb{R}$ .

Para la búsqueda de valores esperados se colocan parámetros iniciales donde comienza la búsqueda local mediante números aleatorios. Estos valores de inicialización del MIRPSA formados por el vector entero mixto  $v_0 = (x^{(1)}, \dots, x^{(n)}, y^{(1)}, \dots, y^{(m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m$ , en el espacio Euclideo de dimensión  $(n + m)$  es el centro de búsqueda de un patrón de estimación inicial con un contador de iteración  $k$  que iniciaría al obtener el número completado de puntos de pruebas pertenecientes a la búsqueda del modelo, simultáneamente se puede lograr obtener la mejor prueba otorgado por el MIRPSA.

De igual manera se cuenta con un indicador inicial de movimiento booleano  $s$ , inicializado en 0, cambiando su valor a 1 cuando se obtiene la mejor prueba dada por el MIRPSA, es decir:

$$f(z) < f(v_k) \tag{4.4a}$$

donde:

$$z = (z^{(1)}, \dots, z^{(n)}, z^{(n+1)}, \dots, z^{(n+m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m \tag{4.4b}$$

Al activar el indicador inicial de movimiento permanecen constantes los parámetros iniciales de una distribución uniforme continua y la distribución uniforme discreta, esto será explicado con más detalles en la Sección 4.

## Detalles del Algoritmo

Para demostrar la convergencia del MIRPSA se muestra un enfoque de Cadena de Markov, representado por el siguiente conjunto infinito de estados:

$$\Lambda = \{d(q)\}_{q=0}^{\infty} \quad (4.5)$$

donde los estados  $d(q)$  se definen por una medida de los  $q$ th patrones de búsqueda aleatorio  $\mathcal{H}_q$ ,  $\forall q \in \mathbb{N}$ . [Brea \(2015\)](#) incluye casos de baja dimensión para mostrar con más detalles el MIRPSA, cuando el algoritmo es usado para identificar el mínimo de una función cuadrática entero mixto y también algunos ejemplos numéricos que muestran el rendimiento del MIRPSA.

Cuando se lleva a cabo una operación de movimiento en los  $q$ th estados para el patrón de búsqueda aleatorio  $\mathcal{H}_q$ , el MIRPSA mantiene su estado mientras si se lleva a cabo una operación de contracción sobre los  $q$ th para el patrón de búsqueda aleatorio  $\mathcal{H}_q$ , el algoritmo visitará el siguiente estado, siendo esto una operación de contracción cambiando el estado a  $(q + 1)$ th estados. El MIRPSA nunca regresa a los estados anteriores luego de haber pasado por la operación de contracción.

El Algoritmo del Patrón de Búsqueda Aleatorio Enteros Mixtos diseñado por [Brea \(2015\)](#) se muestra en la Figura [4.1](#).

---

**Pseudocode of the MIRPSA**

---

**Initialization:**

Let  $f(x, y) : \mathbb{R}^n \times \mathbb{Z}^m \rightarrow \mathbb{R}$  be given.

Let  $v_0 = (v_0^{(1)}, \dots, v_0^{(n)}, v_0^{(n+1)}, \dots, v_0^{(n+m)})^t = (x^{(1)}, \dots, x^{(n)}, y^{(1)}, \dots, y^{(m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m$  be an initial pattern search center guess.

Let  $k = 0$  be an initial iteration counter.

Let  $s = 0$  be an initial boolean moving indicator.

**Declare:**

$z = (z^{(1)}, \dots, z^{(n)}, z^{(n+1)}, \dots, z^{(n+m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m$  as any trial point.

$c = (c^{(1)}, \dots, c^{(n)}, c^{(n+1)}, \dots, c^{(n+m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m$  as temporary point.

**Data:**

Let  $M$  be a number of trial points belonging to each  $k$ -th pattern search;

Let  $\delta^{[0]} \in \mathbb{R}_+$  be the initial parameter of continuous uniform distribution;

Let  $\Delta^{[0]} \in \mathbb{N}_+$  be the initial parameter of discrete uniform distribution;

Let  $\alpha \in \mathbb{R} | 0 < \alpha < 1$  be the real shrinking parameter;

Let  $\beta \in \mathbb{R} | 0 < \beta < 1$  be the integer shrinking parameter;

Let  $\varepsilon \in \mathbb{R}_+$  be the tolerance used to test convergence;

$d^{[0]} \leftarrow \delta^{[0]} + \Delta^{[0]}$ ;

$c \leftarrow v_k$ ;

**while**  $d^{[k]} > \varepsilon$  **do**

**for**  $j \leftarrow 1$  **to**  $M$  **do**

**for**  $i \leftarrow 1$  **to**  $n$  **do**

$z^{(i)} \leftarrow c^{(i)} + u(-\delta^{[k]}, \delta^{[k]})$

**end**

**for**  $i \leftarrow n + 1$  **to**  $n + m$  **do**

$z^{(i)} \leftarrow c^{(i)} + \bar{u}(-\max(1, \Delta^{[k]}), \max(1, \Delta^{[k]}))$

**end**

**if**  $f(z) < f(v_k)$  **then**

$s \leftarrow 1$ ;

$v_k \leftarrow z$ ;

**end**

**end**

$k \leftarrow k + 1$ ;

$v_k \leftarrow v_{k-1}$ ;

**if**  $s = 1$  **then**

$s \leftarrow 0$ ;

$c \leftarrow v_k$ ;

$\delta^{[k]} \leftarrow \delta^{[k-1]}$ ;

$\Delta^{[k]} \leftarrow \Delta^{[k-1]}$ ;

**else**

$\delta^{[k]} \leftarrow \alpha \delta^{[k-1]}$ ;

$\Delta^{[k]} \leftarrow \beta \Delta^{[k-1]}$ ;

**end**

$d^{[k]} \leftarrow \delta^{[k]} + \Delta^{[k]}$ ;

**end**

**Report:** the optimum  $v_k$ , and then **terminate**.

---

**Figura 4.1.** Pseudocódigo de el MIRPSA



Para explorar dentro del algoritmo se deberá inicializar los parámetros con valores descritos por el usuario o programador, para ello es importante conocer el significado de cada parámetro pertenecientes a dicho algoritmo:

Vector Entero mixto: centro de búsqueda del patrón aleatorio inicial.

$$v_0 = (v_0^{(1)}, \dots, v_0^{(n)}, v_0^{(n+1)}, \dots, v_0^{(n+m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m \quad (4.6)$$

Punto de prueba aleatorio: conjunto de puntos de pruebas aleatorias, que es generada por localizaciones aleatorias alrededor de cada  $k$ th del centro de patrón de búsqueda usando distribuciones uniformes continuas  $\delta \in \mathbb{R}^+$  y discretas  $\Delta \in \mathbb{N}^+$ .

$$z = (z^{(1)}, \dots, z^{(n)}, z^{(n+1)}, \dots, z^{(n+m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m \quad (4.7)$$

Punto de prueba temporal: centro del patrón de búsqueda inicial que toma valores de un conjunto de puntos provenientes del vector  $v_k$  inicial, siendo  $v_k$  los valores de prueba inicial para comenzar con la búsqueda y obtener la función objetivo actual  $f(v_k)$ . Cuando se ha obtenido la mejor prueba es decir  $f(z) < f(v_k)$ , donde  $f(z)$  es la función evaluada en el punto de prueba aleatorio entonces  $v_k \leftarrow v_z$ .

$$c = (c^{(1)}, \dots, c^{(n)}, c^{(n+1)}, \dots, c^{(n+m)})^t \in \mathbb{R}^n \times \mathbb{Z}^m \quad (4.8)$$

Como ya se ha mencionado el MIRPSA tiene dos operaciones principales la operación de movimiento y la operación de contracción. La operación de movimiento se lleva a cabo cuando se mejora el valor de la función objetivo actual  $f(v_k)$ , en cualquiera de las  $k$ th iteraciones, la cual moverá el centro de la búsqueda a un mejor punto de un conjunto de  $M$  puntos de prueba pertenecientes a cada búsqueda del modelo  $k$ th.

La operación de contracción se ejecuta en la  $k$ th iteración, cuando el actual centro del patrón de búsqueda no ha mejorado el punto de prueba de la  $k$ th iteración. Por lo tanto, se contrae el tamaño del patrón de búsqueda aleatorio con el parámetro  $\alpha \in \mathbb{R}$  para reducir el rango de una distribución uniforme continua y  $\beta \in \mathbb{R}$  para contraer la longitud de una distribución discreta.

Estas operaciones se llevan a cabo hasta que se cumpla el criterio de parada del MIRPSA  $d > \varepsilon$ , es decir  $\alpha\delta^{[k-1]} + \beta\Delta^{[k-1]} \leq \varepsilon$ , donde  $\varepsilon \in \mathbb{R}$  es la tolerancia para probar la convergencia.

Para codificar el MIRPSA dentro del ambiente de simulación, se le agregan algunas subrutinas que permiten relacionar el algoritmo con el ambiente de [SIMUL8 \(2013\)](#) a tiempo de reloj. La simulación se realiza con variables reales siendo éste la representación del tiempo dentro del simulador, por lo que es necesario realizar modificaciones para tener sincronización con los tiempos, y las variables reales y enteras utilizadas logrando interactuar el algoritmo con el software. En la siguiente Sección [4](#) se explicará con más detalle la interacción del MIRPSA con el simul8.

## Descripción del problema

Para el estudio del comportamiento de tráfico que se presenta con frecuencia cuando se solicita un servicio por parte de un conjunto de clientes lo cual requiere mejorar el sistema disminuyendo el tiempo de servicio, se debe realizar un estudio detallado para su optimización. En este trabajo se realiza el estudio del modelo en tiempo discreto y el número de servidores y de colas dependerá de las variables de decisión  $x \in \mathbb{R}^n$  e  $y \in \mathbb{Z}^m$ . Como se menciona en la Sección [4](#) se requiere minimizar el tiempo promedio de atención del cliente dentro del sistema lo que conlleva al uso de la función objetivo de la Ecuación [4.1a](#), que depende del tiempo promedio que el cliente ha permanecido en el sistema  $\bar{f}(x, y)$  y del número de clientes no atendidos  $\eta(x, y)$ , cuyas variables de decisión son  $x$  e  $y$ . Los clientes no atendidos ocurren cuando la capacidad de la cola es

pequeña con respecto a la cantidad de clientes que entran a las etapas del sistema. Es decir, cuando el cliente se encuentra con la cola a su máxima capacidad lo que implica un caso de frustración el cliente sale del sistema, ocasionando esto un aumento a la función objetivo. Por lo que el sistema tratará de aumentar la capacidad de las colas y disminuir el tiempo de atención, pero este análisis tomará sentido mas adelante.

Se realiza un modelo de  $n$  servidores y  $m$  colas para poder aplicar el algoritmo y encontrar un valor óptimo al sistema mediante la búsqueda de un mínimo local. Los servidores  $n$  se relacionan con la variable  $x$  que representa el tiempo de servicio en cada servidor y  $m$  el número de colas que se encuentran en el sistema que están relacionados con la capacidad de la cola  $y$ . Por ejemplo, si nuestro modelo se basa en 3 colas quiere decir que  $m = 3$  e  $y^m$  contiene los valores de la capacidad de cada cola, en el caso real si  $n = 3$  el número de servidores o de actividades en el sistema serán 3 y el tiempo de cada servidor se encontrará en  $x^n$ .

La función objetivo está sujeta a restricciones no lineales tal como se muestra en la Ecuación 4.3, donde las variables de decisión están restringidas. Cuando las variables de decisión se encuentran fuera del intervalo, el valor de la función objetivo se alejará del punto óptimo, esto es debido a que se agrega una función de penalización a la función objetivo. La función de penalización viene dada por:

$$\hat{f}(x, y) = \bar{f}(x, y) + P(z) \quad (4.9a)$$

siendo:

$$P(z) = K_{max} \sum_{i=1}^{n+m} (\max(z_i - h_i, 0) + \max(l_i - z_i, 0)) \quad (4.9b)$$

donde  $k_{max}$  es la constante de penalización.

## Interacción entre el MIRPSA y el software Simul8

Para interactuar el MIRPSA con el ambiente de simulación de simul8 se comienza realizando algunos cambios en los parámetros iniciales quedando de la siguiente manera:

Cambio de Parámetros		
MIRPSA	Simulador	Descripción
$M$	$q$	Puntos de pruebas de cada iteración
$\delta^{[0]}$	$a$	Distribución uniforme continua
$\Delta^{[0]}$	$b$	Distribución uniforme discreta
$\alpha$	$alfa$	Parámetro de contracción real
$\beta$	$beta$	Parámetro de contracción entero
$\varepsilon$	$e$	Tolerancia

**Tabla 4.1.** Parámetros Iniciales

Estos parámetros iniciales están ubicados en la hoja de cálculo llamado dentro del ambiente de trabajo de simul8 como Spreadsheets “PARAMETROS INICIALES”, así como las variables de decisión  $x$ ,  $y$ ,  $n$ ,  $m$  y los parámetros para la penalización  $k_{max}$ ,  $l_i$ ,  $h_i$  correspondientes para las variables reales y enteras.

Para realizar la estructura lógica se hace uso del Visual Logic que permite la codificación de la interacción compleja, siendo éste un enfoque alternativo para el control de un modelo mediante un lenguaje de programación.

Antes de ello, se crean etiquetas que permiten tener atributos que distinguen las entidades en un determinado proceso, además se crean variables globales y se utilizan dentro del Visual Logic. Las etiquetas utilizadas se relacionan con: el tiempo en que cliente sale de una etapa de servicio, éste es representado como  $L_\phi$ , donde  $1 \leq \phi \leq n$ ,  $n \in \mathbb{Z}$  el cual es el número de servidores dentro del sistema; el tiempo en el cual el cliente ha estado dentro del sistema corresponde a  $T_s$ , éste es la sumatoria de los tiempo de servicio, es decir:

$$Ts = \sum_{\phi=1}^n L_{\phi} \quad (4.10)$$

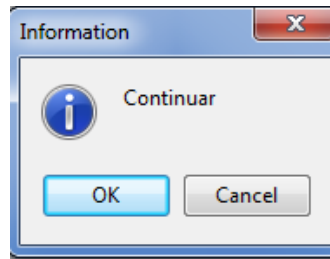
Las variables globales representan las hojas de cálculo, los contadores, las variables aleatorias y variable booleana.

En el Spreadsheets “RESULTADOS” se mostrarán los valores otorgados por el MIRPSA para los reales y enteros tales como  $v_z$ ,  $v_c$  y  $v_k$ , siendo  $v_k$  la representación de los valores de distribución más cercanos al óptimo,  $v_z$  el punto de prueba aleatorio y  $v_c$  centro de búsqueda del patrón aleatorio. Se crean distribuciones dado por una exponencial según los valores de los parámetros iniciales reales correspondiente a la variable de decisión  $x$ . En el visual de cada servidor o actividad se crea una función para representar el tiempo en que el cliente es atendido, tal como se muestra en la Ecuación 4.10 y una variable correspondiente al tiempo promedio del cliente dentro del sistema, éste es llamado *VariabletiempoAct*.

Para la búsqueda del mínimo local se requiere de las variables aleatorias, basadas en los parámetros iniciales uniformes continuas y discretas. En la simulación para probar cualquier punto de rastro llamado en el modelo  $v_z$  como el punto de prueba aleatorio se realiza el cambio en la distribución. Para simular el tiempo en que ocurre el cambio se utiliza el Time Check Logic del simul8, donde se programa para cada seis horas el cambio del valor de la distribución o la siguiente evaluación de los puntos otorgados por el MIRPSA.

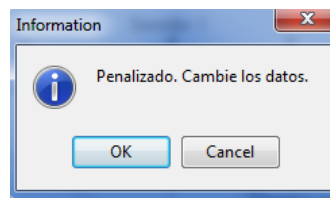
Antes de comenzar con la simulación se realiza una inicialización para probar los parámetros iniciales y que estos no se encuentren dentro de la penalización. Para ello, al presionar el botón de reinicio de reloj inmediatamente llama al Reset Logic donde se trasladan los parámetros iniciales a la hoja de cálculo llamada “RESULTADOS”, allí se guardan los distintos puntos de prueba.

Si los primeros puntos no son penalizados el mensaje será el siguiente:



**Figura 4.2.** Mensaje de Continuar

De lo contrario, se mostrará la Figura 4.3 la cual estará penalizada y se deberán cambiar los valores de las variables de decisión iniciales o los parámetros de restricción.



**Figura 4.3.** Mensaje de Penalización

En el Visual Logic “PENALIZACIÓN” se lee como valores actuales a las variables de decisión  $x$  e  $y$ , similarmente fueron colocadas en la hoja de cálculo “VALORES ACTUALES”, la cual muestra los últimos tres puntos de evaluación del patrón de búsqueda y representa los valores de nuestra distribución exponencial. De igual manera, se realiza la evaluación de las restricciones para conocer si el sistema está penalizado o no, esto es señalado en la Ecuación 4.9b.

El criterio de parada se lleva a cabo mediante la tolerancia “ $e$ ” y la suma de los parámetros de distribución uniforme continua  $a$  y discreta  $b$ , donde  $d = a + b$  y mientras  $d > e$  se activará el MIRPSA. El visual Logic del Stop Run ocurre cuando la simulación se ha detenido usando el botón de parada o en cualquier tiempo de reloj donde la simulación es detenida, en este caso se detiene cuando transcurren las seis

horas para el servicio. Al conocer esto el criterio de parada es colocado en el Stop Run donde aumentan las  $k$  iteraciones cuando ha completado el número de pruebas correspondientes a la búsqueda y finalizará la búsqueda del modelo al cumplir con el criterio de parada, es decir  $d < e$ .

Para representar el número de puntos de prueba pertenecientes a cada búsqueda del modelo, como se muestra en la Tabla 4.1 representado por  $q$ , se considera como criterio mínimo  $q = 2 \cdot (n + m)$  y se ejecutada en el Stop Run.

### Inicialización del sistema

A continuación se mostrará como iniciar las pruebas dentro del simulador:

Tal como se muestra en la Figura III.1 ubicada en anexos, se colocan los valores de distribución continua  $a$  y discreta  $b$  en las dos primeras columnas.

Los parámetros de contracción real y entero seguidamente llamadas en el simulador como  $alfa$  y  $beta$  se colocan en las siguientes dos columnas. Recordando que:  $0 < alfa < 1$  y  $0 < beta < 1$

El valor de la tolerancia  $e$  en este caso se asumirá como 0,1 pero puede ser modificado en el Spreadsheets o se podrá modificar en el Visual Logic del Stop Run, como se muestra en la Figura XVII.1.

El parámetro que representa el número de pruebas pertenecientes a la búsqueda del modelo, es decir  $q$  se puede modificar en el Visual Logic del Stop Run y es representado como  $Mnm$ .

La variable booleana  $s$  se debe inicializar en cero, ya que se activará cuando se ha mejorado la función de desempeño, es decir  $f(z) < f(vk)$ .

Las variables de decisión  $x^n$  que representa los tiempos de servicios y  $y^m$  que representan como se ha dicho anteriormente las capacidades de las colas son colocadas en el Spreadsheet de “PARÁMETROS INICIALES”, tal como se muestra en la Figura [III.2](#). En la fila 2 columna j, se coloca el valor en tiempo de las llegadas de clientes al sistema, y a partir de la fila 3 de la misma columna se colocan los tiempos de los diferentes servidores. En la fila 2 columna K, se deberán colocar las capacidades de los servidores seguidamente como capacidad 1, capacidad 2 y capacidad 3.

Las dimensiones del sistema  $n$  y  $m$  deben coincidir con los números de actividades y colas del sistema ya que  $n$  representa la dimensión real y  $m$  la entera. Por lo tanto, se debe colocar el valor de la dimensión incluyendo la actividad que representa la entrada de clientes al sistema. Es decir, si se tiene 3 servidores y una entrada el valor que corresponde a la columna L fila 2 del Spreadsheet de los Parámetros iniciales es 4.

Los valores de los intervalos para las restricciones no lineales se modifican en las columnas N, O, P y Q, tal como se muestra en la Figura [III.3](#).

En el Spreadsheet “RESULTADOS” se refleja los últimos valores otorgados por el MIRPSA representado en el simul8 como Vc(Reales) y Vc(Enteros) siendo estos el centro del patrón de búsqueda, tal como se muestra en la Figura [V.1](#). Como simul8 trabaja con el tiempo de simulación, toma los valores de penalización (si ocurre la penalización) en la siguiente evaluación del punto aleatorio. Por este motivo se anexan tres columnas que muestran las últimas tres penalizaciones, estas columnas son L, M y N.

Ahora, si se quiere conocer cuantos clientes fueron atendidos en cada prueba aleatoria, esto se ubica en el Spreadsheets Función [IV.1](#) columna G y los clientes no atendidos se pueden ver en la columna L del mismo Spreadsheets. Al igual que el tiempo del cliente en cada servidor en las columnas A, B y C.



Los valores de distribución correspondiente al último valor de la función objetivo obtenido por el MIRPSA se muestran en el Spreadsheets “FUNCIÓN MENOR” columna H y J. El valor de la función de desempeño en las  $k$ th iteraciones estarán mostradas en la columna M y el último valor de la función de desempeño en la columna K del mismo Spreadsheets.

## CAPÍTULO V

### PRUEBAS EXPERIMENTALES Y RESULTADOS

En este capítulo se presenta el estudio de diferentes diseños, ya sea en serie, paralelo, mixtos y retroalimentadas. La idea es observar y analizar las diferencias en los valores del patrón de búsqueda por medio del software de [SIMUL8 \(2013\)](#).

Los diseños a estudiar poseen los parámetros iniciales de la [Tabla 5.1](#) para la contracción, movimiento y el criterio de parada:

Valor inicial		
MIRPSA	Simulador	
$M$	$q$	2 (n+m)
$\delta^{[0]}$	$a$	2
$\Delta^{[0]}$	$b$	4
$\alpha$	<i>alfa</i>	0.9
$\beta$	<i>beta</i>	0.6
$\varepsilon$	$e$	0.1

**Tabla 5.1.** Parámetros Iniciales

Las tasas de llegada y de servicio así como la dimensión en el espacio euclidiano serán los siguientes de la Tabla 5.2:

Valores iniciales			
$n$	$m$	$x(\text{Reales})$	$y(\text{enteros})$
3	3	5	8
		12	8
		4	8
		4	

**Tabla 5.2.** Parámetros Iniciales  $x^n$  e  $y^m$

donde 5 es el parámetro de la distribución exponencial del tiempo entre llegadas de los clientes al sistema y no forma parte de las variables de decisión, por este motivo  $n \in \mathbb{R}$  toma el valor de 3, lo que quiere decir que las variables de decisión son tomadas desde la segunda fila para representar los tiempos de servicio con una distribución exponencial, estos son: 12, 4 y 4 con  $n = 3$ , siendo respectivamente el tiempo de atención del servidor 1, servidor 2 y servidor 3. Para representar el valor de las capacidades se tiene: 8, 8 y 8 para  $m = 3$  donde  $m \in \mathbb{Z}$ , quedando un sistema con las siguientes características:

- tres colas;
- tres servidores;
- una entrada;
- tasas de llegada y de servicio con distribución de probabilidad exponencial;
- la disciplina de la cola es FIFO.

La función de penalización está sujeta a restricciones no lineales como es mostrado en la Ecuación 4.3. Los intervalos de penalización están dadas por la siguiente Tabla 5.3:

Valores iniciales				
$l_i$ (Reales)	$h_i$ (Reales)	$l_i$ (Enteros)	$h_i$ (Enteros)	kmax
2	200	1	50	$1 \times 10^6$
2	150	1	300	
2	300	1	100	

**Tabla 5.3.** Intervalos para la Penalización

La función objetivo a optimizar viene dada de la Ecuación 4.1a, sin embargo para los experimentos a realizar se le ha anexado unos factores multiplicativos en el tiempo promedio del cliente dentro del sistema y en la cantidad de clientes que salen del sistema. Esto con el fin de darle importancia o peso a los clientes que salen del sistema o al tiempo promedio del cliente. Por lo tanto para las evaluaciones de los distintos canales de servicio serie, paralelo, mixto y etapas retroalimentadas es la siguiente:

$$\underset{(x,y) \in \mathbb{R}^n \times \mathbb{Z}^m}{\text{minimize}} \alpha_1 \bar{f}(x,y) + \alpha_2 \eta(x,y), \quad (5.1a)$$

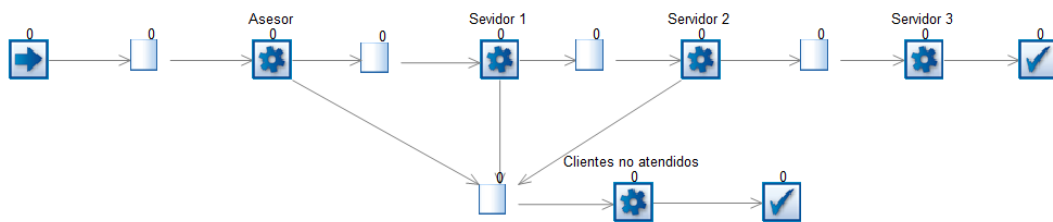
donde:

$$\alpha_1 = 0,5, \alpha_2 = 0,5 \quad (5.1b)$$

con factores multiplicativos  $\alpha_1$  y  $\alpha_2$  de igual importancia.

## Diseño en Serie

Para el primer experimento se considera el caso serie mostrado en la Figura 5.1 y con los parámetros iniciales ya mencionados, basado en la función objetivo de la Ecuación 5.1a cuyo factor multiplicativo de importancia son los mostrados en la Ecuación 5.1b, donde al ocurrir la penalización el algoritmo hace uso de la Ecuación 4.9a, entonces:



**Figura 5.1.** Modelo en Serie

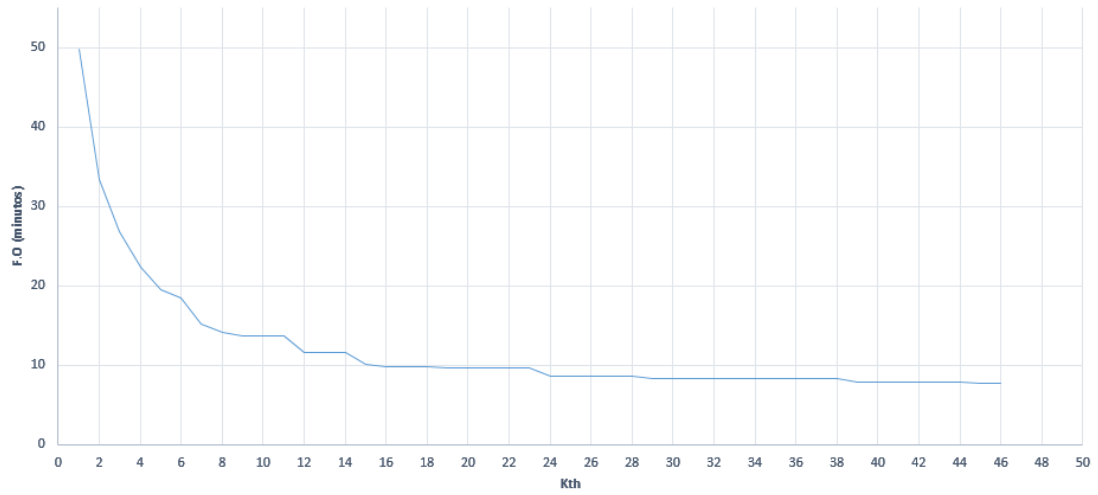
Al finalizar la simulación se tiene que el tiempo promedio del cliente dentro del sistema es 7,6809 minutos. El tiempo de atención por parte de los distintos servidores así como las capacidades de las colas obtenidos en el Spreadsheets RESULTADOS es mostrada en la Tabla 5.4:

Resultados	
$v_k$ (Reales)	$v_k$ (Enteros)
5	9
2.9203	9
2.0013	13
2.0013	

**Tabla 5.4.** Resultados del Modelo Serie

Con los valores de distribución obtenidos para los servidores y las capacidades de las colas no ocurría el caso de frustración por lo que el número de clientes no atendidos es nulo, logrando satisfactoriamente la atención de 87 clientes.

A continuación se mostrará en la Figura 5.2 la mejora de la función de desempeño:

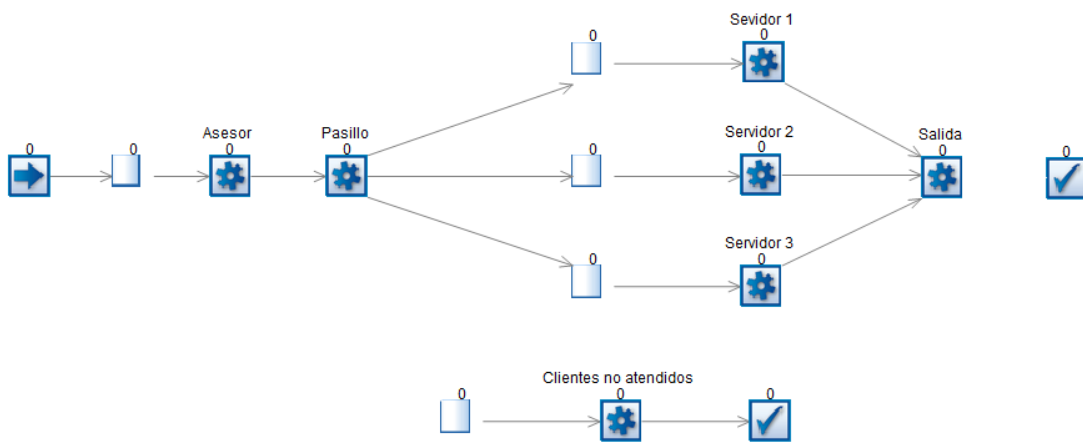


**Figura 5.2.** Gráfica de la mejora de la función objetivo en el Modelo Serie

donde *kth* es el número de iteraciones.

### Diseño en Paralelo

Con las mismas condiciones iniciales se tiene el siguiente modelo:



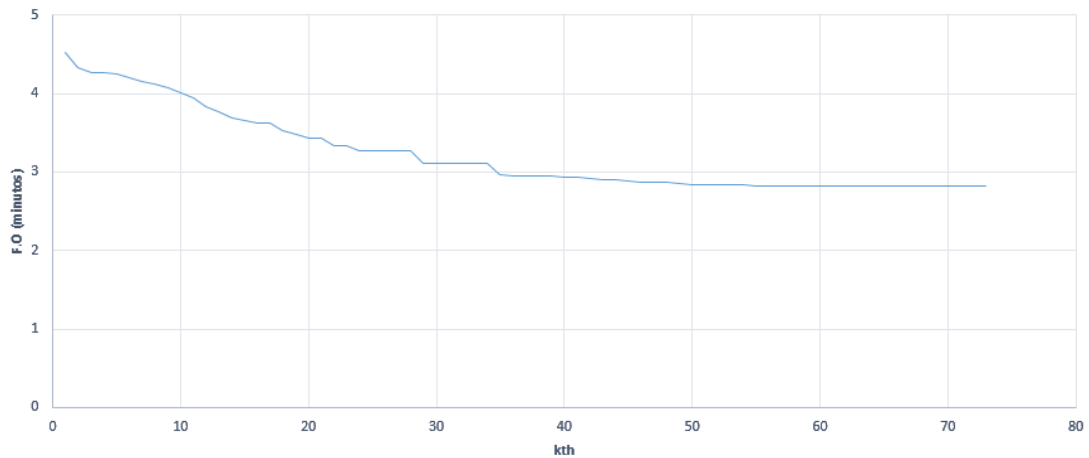
**Figura 5.3.** Modelo en Paralelo

Se obtiene que el tiempo promedio del cliente dentro del sistema es de 2,8242 minutos y la cantidad de clientes no atendido es nulo. El tiempo de atención para los distintos servidores y las capacidades de las colas son mostradas en la Tabla 5.5:

Resultados	
$v_k$ (Reales)	$v_k$ (Enteros)
5	10
2.8906	10
3.6983	11
3.6983	

**Tabla 5.5.** Resultados del modelo Paralelo

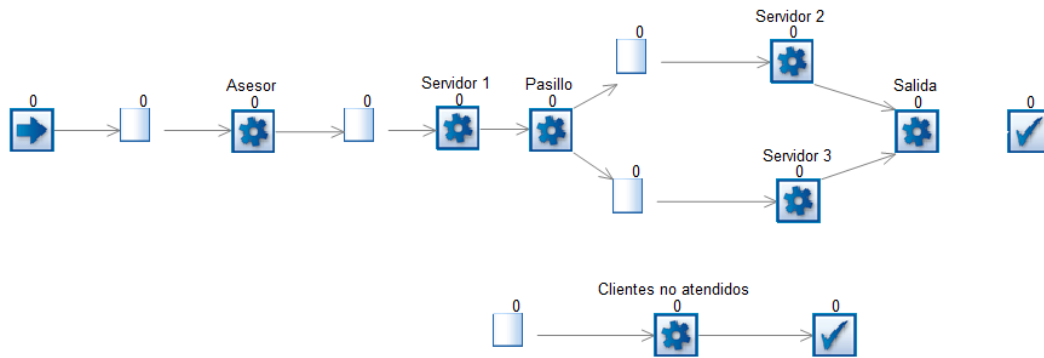
Luego de realizar la prueba del modelo, el número de clientes atendidos fue de 78 clientes. La representación de la disminución de la función objetivo del modelo paralelo es mostrada en la Figura 5.4:



**Figura 5.4.** Gráfica de la mejora de la función objetivo en el Modelo Paralelo

## Diseño Mixto

Para este experimento el modelo a utilizar es la Figura 5.5:



**Figura 5.5.** Modelo Mixto

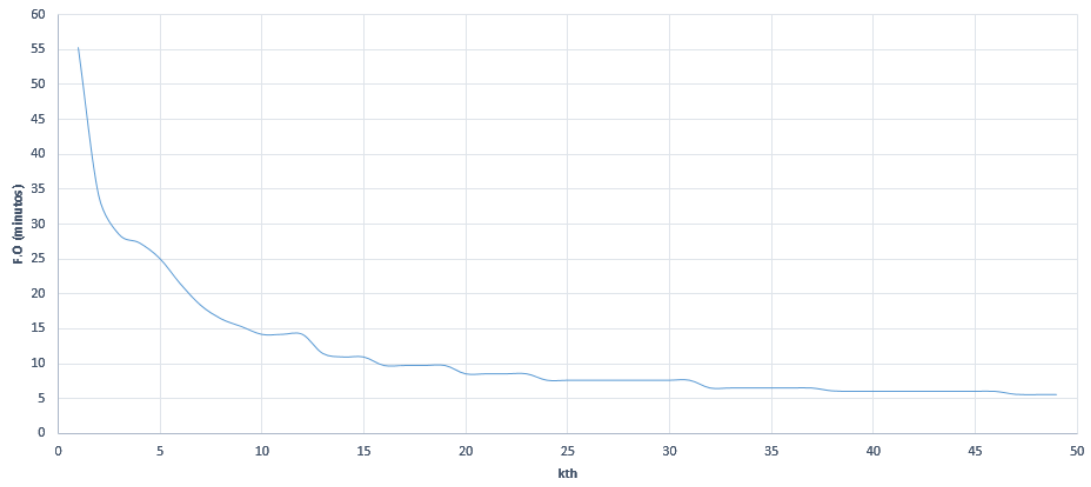
El tiempo promedio del cliente dentro del sistema es de 5,6260 minutos y la cantidad de clientes no atendido es nulo. El tiempo de atención para los distintos servidores y las capacidades de las colas es mostrada en la Tabla 5.6:

Resultados	
$v_k$ (Reales)	$v_k$ (Enteros)
5	8
2.0005	7
3.1433	7
3.1433	

**Tabla 5.6.** Resultados del modelo Mixto

Con los valores obtenidos en la Tabla 5.6 se atendieron 63 clientes y la gráfica de la función objetivo es la mostrada en la Figura 5.6:

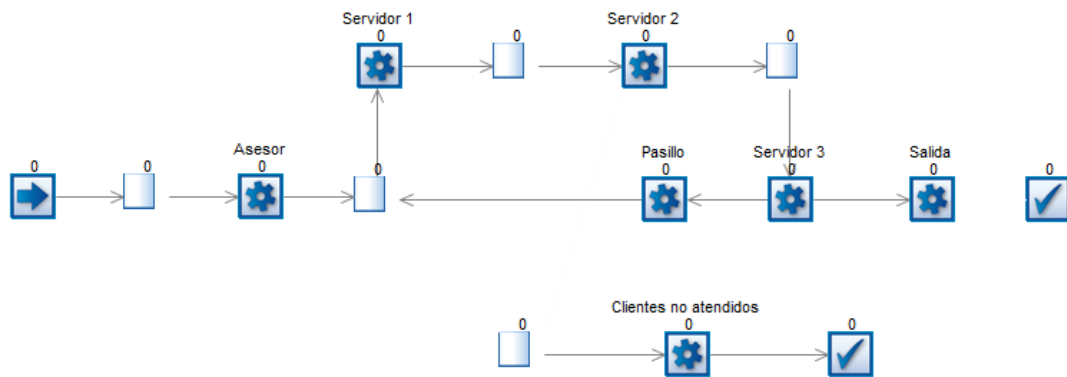




**Figura 5.6.** Gráfica de la mejora de la función objetivo en el Modelo Mixto

## Diseño de Etapas Retroalimentadas

A diferencia de los modelos anteriores, este posee una etapa de retroalimentación donde al finalizar las dos primeras etapas de servicios una cantidad de clientes retornan a la atención de dichos servicios. Cabe destacar que no todos los clientes se regresan al sistema, por lo que se ha aproximado la cantidad de clientes de retorno, esto es 70% de clientes que salen de manera satisfactoria del sistema y el resto retorna. A continuación el modelo a representar es el siguiente:



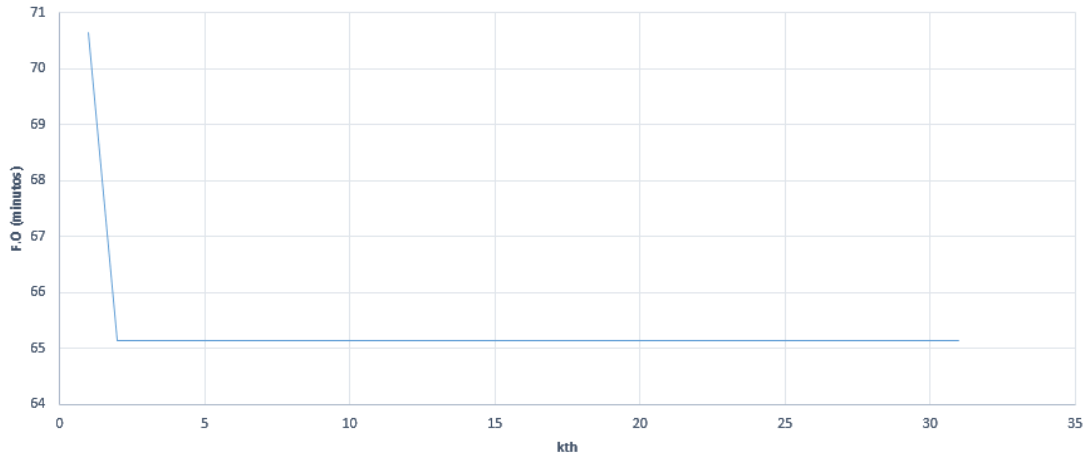
**Figura 5.7.** Modelo de Etapas Retroalimentadas

El tiempo promedio del cliente dentro del sistema obtenido es de 65,1439 minutos y los clientes no atendido es nulo. El tiempo de atención para los distintos servidores y las capacidades de las colas son mostrados en Tabla [5.7](#):

Resultados	
$v_k$ (Reales)	$v_k$ (Enteros)
5	25
5.0681	13
2.6386	29
2.6386	

**Tabla 5.7.** Resultados del Modelo de Etapas Retroalimentadas

La cantidad de clientes atendidos es de 59 clientes y la gráfica de la función de desempeño es la Figura 5.8:



**Figura 5.8.** Gráfica de la mejora de la función objetivo en el Modelo de Etapas Retroalimentadas

Con el modelo serie de la Figura 5.1 se realizaron dos experimentos adicionales modificando los valores de los factores de importancia  $\alpha_1$  y  $\alpha_2$  en la función objetivo a minimizar mostrado en la Ecuación 5.1a. El primer experimento tiene los siguientes factores:

$$\alpha_1 = 0,2; \alpha_2 = 0,8 \quad (5.2)$$

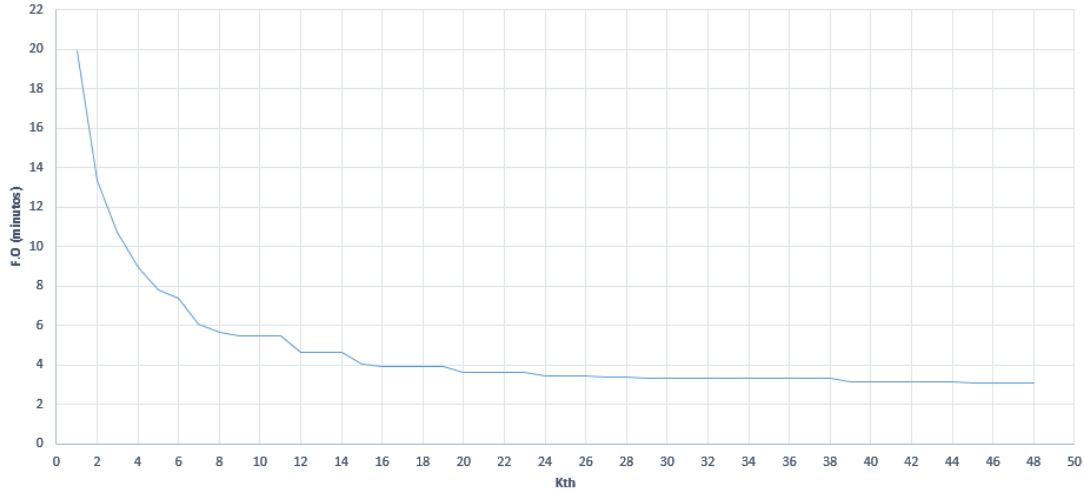
Los valores de distribución para los servidores y capacidades de las colas es mostrada en la Tabla 5.8:

Resultados	
$v_k$ (Reales)	$v_k$ (Enteros)
5	11
2.9192	14
2.0002	13
2.0002	

**Tabla 5.8.** Resultados del Modelo Serie con  $\alpha_1 = 0,2$  y  $\alpha_2 = 0,8$

Se obtuvo que el tiempo promedio de los clientes dentro del sistema con los valores de la Tabla 5.8 es 3,0662 minutos, siendo atendidos 63 clientes y no atendidos nulo.

A continuación se mostrará de manera gráfica en la Figura 5.9 los valores de la función objetivo a optimizar:



**Figura 5.9.** Gráfica de la función objetivo del Modelo Serie con  $\alpha_1 = 0,2$  y  $\alpha_2 = 0,8$

En el segundo experimento se toman los factores de importancia como sigue:

$$\alpha_1 = 0,7; \alpha_2 = 0,3 \quad (5.3)$$

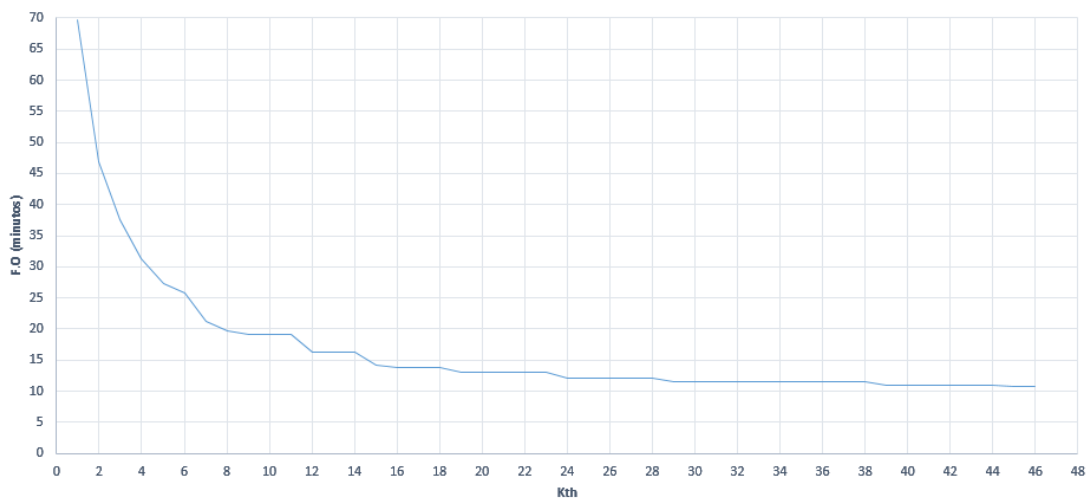
Los valores de distribución para los servidores y capacidades de las colas obtenidos fueron:

Resultados	
$v_k$ (Reales)	$v_k$ (Enteros)
5	9
2.9203	9
2.0013	13
2.0013	

**Tabla 5.9.** Resultados del Modelo Serie con  $\alpha_1 = 0,7$  y  $\alpha_2 = 0,3$

El tiempo promedio de los cliente dentro del sistema con los valores de la Tabla 5.9 es 10,7532 minutos, siendo atendidos 67 clientes y no atendidos nulo.

La gráfica de los valores de la función objetivo a optimizar es mostrada en la Figura 5.10:



**Figura 5.10.** Gráfica de la función objetivo del Modelo Serie con  $\alpha_1 = 0,7$  y  $\alpha_2 = 0,3$

## CAPÍTULO VI

### ANÁLISIS DE LOS RESULTADOS

Para el primer modelo en serie se observa en la Tabla 5.4, en comparación a la inicial que el tiempo de atención al cliente disminuye y las colas aumentan. El tiempo de atención inicial del servidor 1 es mayor al tiempo entre llegadas de los clientes, por lo que se forma cola de espera en ese servidor, el MIRPSA intenta disminuir el valor de atención del servidor y aumentar la cola para evitar la fuga de clientes. Las capacidades de las colas también logran disminuir si no se encuentran con cuellos de botella, a medida que el número de puntos de prueba pertenecientes a la búsqueda del modelo logra alcanzar el valor del parámetro inicial  $q$  y encontrar una función menor los parámetros de distribución continua y discreta permanecen constantes y al no encontrarla el parámetro de contracción real y entero disminuyen, achicando el patrón de búsqueda lo que implica que el nuevo valor de los parámetros de distribución continua, discreta y el intervalo de búsqueda aleatoria disminuyan donde el movimiento del patrón de búsqueda es menor.

El factor de importancia dentro de la función objetivo de la Ecuación 5.1a, es primordial en el diseño colas en este proyecto, ya que a medida que variamos estos factores multiplicativos toma importancia las salidas de los clientes y el tiempo promedio que el cliente se encuentra en el sistema. En la Figura 5.2 se observa que disminuye el valor de la función objetivo tomando en cuenta que tienen el mismo factor de importancia los clientes que salen del sistema sin ninguna atención y el promedio de tiempo del cliente en el sistema a medida que ha completado el número de pruebas pertenecientes

al modelo  $q$ , lo que implica que el MIRPSA está moviendo el centro del patrón a medida que aumentan las iteraciones  $kth$  hasta cumplir el criterio de parada  $d > \epsilon$ .

En el modelo paralelo los clientes se dirigen a la cola más corta impidiendo el colapso en las etapas de servicio, por lo que con un  $\lambda$  para la distribución exponencial igual a los valores de Tabla 5.2, no existen clientes no atendidos. El MIRPSA a diferencia del modelo serie donde el patrón de búsqueda se mueve dentro del espacio euclidiano encuentra valores de la función objetivo rápidamente con intervalos entre ellos pequeños a medida que ocurren cambios aleatorios en la búsqueda del modelo, aumentando las iteraciones de la función objetivo  $kfo$ , es decir al disminuir el tiempo de servicio según el patrón de búsqueda aleatorio el valor de la función objetivo disminuye de manera casi inmediata, tal como se observa en la Figura 5.4 al culminar  $q$  con  $kth$  iteraciones, por lo que al comparar la Figura 5.2 del modelo serie con la Figura 5.4 del modelo en paralelo el número de las iteraciones  $kth$  es mayor que en serie, esto es debido a que este tipo de sistema en paralelo la atención al cliente es individual y el estudio de los tiempos de servicios y las capacidades de colas son más fáciles y rápidas en analizar. El MIRPSA al realizar cambios aleatorios para la búsqueda haya valores de la función objetivo menor rápidamente permitiendo que la variable Booleana  $s$  sea la mayoría de veces 1 y se mantengan los valores de distribución uniforme continua y discreta  $a$  y  $b$ , lo que implica que la gráfica de la función objetivo tenga una disminución leve.

Para realizar una combinación del modelo serie y paralelo se realizó el modelo mixto de la Figura 5.5, al comienzo de la simulación se nota que existe un caso de frustración en el la cola del servidor 1, debido al tiempo del servicio de esa etapa, tal como ocurre en el modelo serie por tener el valor de tiempo de entre llegadas de los clientes menor al tiempo del servicio. Cuando el MIRPSA va disminuyendo los valores de las distribuciones exponenciales de los servidores ésta trata de disminuir la cantidad de clientes que salen del sistema. En la gráfica de la Figura 5.6 se observa que en las

primeras iteraciones se asemeja a la gráfica de la Figura 5.2 del modelo serie, esto ocurre por el desbordamiento del servidor 1. Al encontrar puntos de equilibrio donde la cantidad de clientes que salen del sistema sin ser atendidos disminuye, la función objetivo decae lentamente para ubicarse en el valor más cercano al óptimo, es decir que en mayores cantidades de iteraciones  $kth$  permanece constante la función objetivo. Observando de esta manera la combinación de la brusca caída de la función objetivo de un modelo serie y finalizando con la disminución leve de el modelo paralelo al establecerse el centro del patrón de búsqueda adecuado.

En el modelo de etapas retroalimentadas de la Figura 5.7 se colapsa la cola del servidor 1, esto es debido a que el tiempo de llegadas de clientes dentro del sistema es menor que la de atención en el servidor adicionando el 30% de clientes que retornan al sistema. Los valores de tiempo de la variable de decisión  $x^n$  disminuyen para evitar la salida de clientes a la etapa de no atendidos y de esta manera disminuir la función de desempeño donde los factores de importancia  $\alpha_1$  y  $\alpha_2$  de la función objetivo 5.1a es lo mostrado en la Ecuación 5.1b. A medida que se aumentan las iteraciones  $kth$  se obtendrá el valor óptimo para este sistema sin embargo, el criterio mínimo a utilizar es  $2 \cdot (n + m)$  o simplemente 10.

Para finalizar se realizó la modificación de la función objetivo variando los valores de  $\alpha_1$  y  $\alpha_2$ . En el primer experimentos con factores  $\alpha_1 = 0,2$  y  $\alpha_2 = 0,8$  los valores de las capacidades obtenidas y mostrada en la Tabla 5.8 aumentan, esto es debido a que el sistema trata de darle importancia a la cantidad de clientes que salen del sistema que al valor de la función promedio observándose en la Figura 5.9, aumentando las capacidades si fuese necesario para evitar la salida de los mismos. En el segundo experimento con valores de importancia de  $\alpha_1 = 0,7$  y  $\alpha_2 = 0,3$  el valor de la función objetivo es mayor comparada a la anterior, ya que tiene mayor significancia el valor de la función promedio la cual la mayoría de veces es muy alta comparada con la cantidad de clientes que salen del sistema, esto se puede observar en la Figura 5.10.



## CAPÍTULO VII

### CONCLUSIONES

Mediante el desarrollo de esta investigación y de los resultados experimentales se puede comprobar que el MIRPSA converge y se puede implementar en diseños de sistemas de colas. En la interacción del algoritmo con el ambiente simul8 se tuvo que realizar modificaciones, una de ellas fue en la salida de los clientes dentro del sistema, ya que uno de los supuestos del proyecto es disminuir los clientes en la cola y evitar que los clientes salgan del sistema sin ser atendidos, debido al caso de frustración. Los modelos de optimización en redes de colas deben enfocarse a liberar la congestión en las etapas de servicio, que luego se equilibran las capacidades de atención según las llegadas de clientes en cada punto de atención. La implementación de estos algoritmos de optimización en este caso el MIRPSA mediante el diseño de sistema de colas consigue obtener los valores cercanos al óptimo basados en parámetros iniciales del sistema aplicando las propiedades de la Cadena de Markov.

En los experimentos de los distintos canales de servicio se puede observar como disminuye los tiempos de servicio cuando nos encontramos en colapso o colas en las etapas del servicio, donde también ocurre el aumento de las capacidades si es necesario.

La limitación computacional impidió que se realizarán  $n$  muestras en la simulación para los distintos canales de servicio. Por lo que solo se obtuvo una muestra y el valor cercano al óptimo será el último valor obtenido otorgado por el MIRPSA.

## **CAPÍTULO VIII**

### **RECOMENDACIONES**

Para el análisis de redes de colas se deben realizar estudios estadísticos que nos permitan conocer con seguridad que el valor obtenido mediante este algoritmo es el óptimo y de esta manera tomar decisiones acerca de los distintos canales de servicio y dar a conocer cual es el mejor diseño.

Se recomienda para el uso de los parámetros iniciales, no penalizar los primeros valores para evitar volver a ingresar valores no penalizados, ya sea modificando los intervalos de restricción de la penalización o las variables de decisión.

Para cambiar las dimensiones del sistema, es decir anexar servidores y colas se recomienda copiar y pegar los visual de los servidores en los actuales del servicio a anexar y modificar las variables, por lo tanto se deben analizar los parámetros del sistema.

## REFERENCIAS

- Balsa, E. (2010). *Algoritmos eficientes para la optimización dinámica de procesos distribuidos* (n.º 9/12/2010).
- Berman, O., Larson, R. C., y Chiu, S. S. (1985). Optimal server location on a network operating as an M/G/1 queue. *Operations Research*, 33(4), 746-771.
- Brea, E. (2005). ¿Optimización por Simulación u Optimización para la Simulación? En T. Pérez, D. Montoya, P. Tremante, y R. Díaz (Eds.), *II seminario de modelos y modelado: Concepto, técnicas y aplicaciones* (Vol. 1, p. 1-12). Caracas. (doi:10.13140/2.1.1392.8329)
- Brea, E. (2013). Una extensión del método de Nelder Mead a problemas de optimización no lineales enteros mixtos. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 29(3), 163-174. (doi:/10.1016/j.rimni.2013.06.005)
- Brea, E. (2015, July 2015). *On the convergence of the Mixed Integer Randomized Pattern Search Algorithm*. (Manuscript accepted for the ICAEM 2015: 17th International Conference on Applied and Engineering Mathematics, October 2015. doi:10.13140/RG.2.1.1363.7845)
- Dora, G., y Jorge, C. (2006). Teoría de colas en la optimización de un sistema. *Revista AMMVEPE*.
- Feldman, Z., Mandelbaum, A., Massey, W. A., y Whitt, W. (2008). Staffing of time-varying queues to achieve time-stable performance. *Management Science*, 54(2), 324-338.
- Gong, Q., Lai, K., y Wang, S. (2008). Supply chain networks: Closed Jackson network models and properties. *International Journal of Production Economics*, 113(2), 567-574.

- Guillermo, V. (2008). Aplicación y Validación de dos Algoritmos, para el Diseño Óptimo de Sistemas de Colas Markovianos. *Proyecto de grado para optar por el título de Ingeniero Industrial*, 173.
- Kumin, H. J. (1969). *The design of Markovian congestion systems* (Tesis Doctoral no publicada).
- Rincón, L. (2008). Introducción a los procesos estocásticos.. (Departamento de de Matemáticas, Facultad de Ciencias UNAM)
- Rojo, H., y Miranda, M. (2009). *Cadenas de Markov* (Vol. 2013) (n.º 102). (Investigación Operativa)
- SIMUL8. (2013). *SIMUL8* (Vol. 2013) (n.º 12/02/2013). (<http://www.simul8.com/>)

## **ANEXOS**

# Apéndice I

## Valores de $t_\alpha$

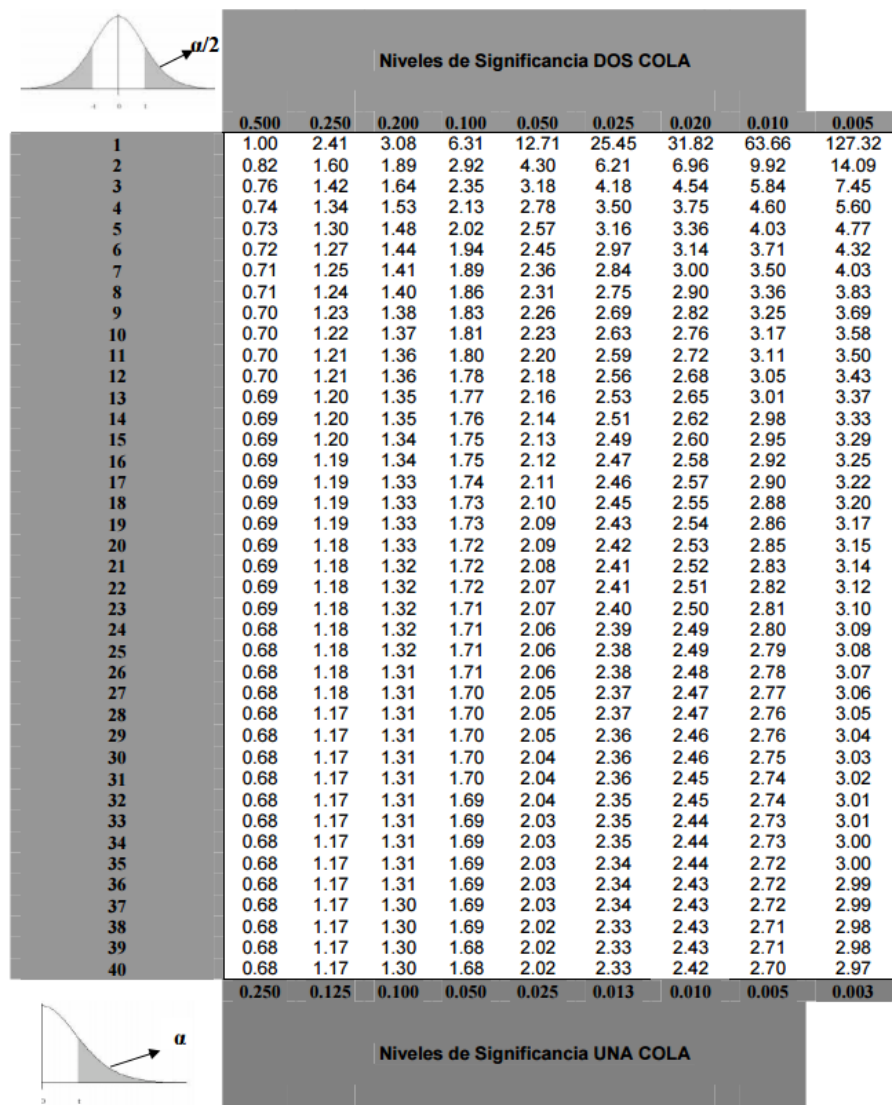


Figura I.1. Valores críticos de la distribución t

## Apéndice II

### Valores de $\chi^2$

v	α										
	0.001	0.025	0.05	0.1	0.25	0.5	0.75	0.9	0.95	0.975	0.999
1	10.827	5.024	3.841	2.706	1.323	0.455	0.102	0.016	0.004	0.001	0
2	13.815	7.378	5.991	4.605	2.773	1.386	0.575	0.211	0.103	0.051	0.002
3	16.266	9.348	7.815	6.251	4.108	2.366	1.213	0.584	0.352	0.216	0.024
4	18.466	11.143	9.488	7.779	5.385	3.357	1.923	1.064	0.711	0.484	0.091
5	20.515	12.832	11.07	9.236	6.626	4.351	2.675	1.61	1.145	0.831	0.21
6	22.457	14.449	12.592	10.645	7.841	5.348	3.455	2.204	1.635	1.237	0.381
7	24.321	16.013	14.067	12.017	9.037	6.346	4.255	2.833	2.167	1.69	0.599
8	26.124	17.535	15.507	13.362	10.219	7.344	5.071	3.49	2.733	2.18	0.857
9	27.877	19.023	16.919	14.684	11.389	8.343	5.899	4.168	3.325	2.7	1.152
10	29.588	20.483	18.307	15.987	12.549	9.342	6.737	4.865	3.94	3.247	1.479
11	31.264	21.92	19.675	17.275	13.701	10.341	7.584	5.578	4.575	3.816	1.834
12	32.909	23.337	21.026	18.549	14.845	11.34	8.438	6.304	5.226	4.404	2.214
13	34.527	24.736	22.362	19.812	15.984	12.34	9.299	7.041	5.892	5.009	2.617
14	36.124	26.119	23.685	21.064	17.117	13.339	10.165	7.79	6.571	5.629	3.041
15	37.698	27.488	24.996	22.307	18.245	14.339	11.037	8.547	7.261	6.262	3.483
16	39.252	28.845	26.296	23.542	19.369	15.338	11.912	9.312	7.962	6.908	3.942
17	40.791	30.191	27.587	24.769	20.489	16.338	12.792	10.085	8.672	7.564	4.416
18	42.312	31.526	28.869	25.989	21.605	17.338	13.675	10.865	9.39	8.231	4.905
19	43.819	32.852	30.144	27.204	22.718	18.338	14.562	11.651	10.117	8.907	5.407
20	45.314	34.17	31.41	28.412	23.828	19.337	15.452	12.443	10.851	9.591	5.921
21	46.796	35.479	32.671	29.615	24.935	20.337	16.344	13.24	11.591	10.283	6.447
22	48.268	36.781	33.924	30.813	26.039	21.337	17.24	14.041	12.338	10.982	6.983
23	49.728	38.076	35.172	32.007	27.141	22.337	18.137	14.848	13.091	11.689	7.529
24	51.179	39.364	36.415	33.196	28.241	23.337	19.037	15.659	13.848	12.401	8.085
25	52.619	40.646	37.652	34.382	29.339	24.337	19.939	16.473	14.611	13.12	8.649
26	54.051	41.923	38.885	35.563	30.435	25.336	20.843	17.292	15.379	13.844	9.222
27	55.475	43.195	40.113	36.741	31.528	26.336	21.749	18.114	16.151	14.573	9.803
28	56.892	44.461	41.337	37.916	32.62	27.336	22.657	18.939	16.928	15.308	10.391
29	58.301	45.722	42.557	39.087	33.711	28.336	23.567	19.768	17.708	16.047	10.986
30	59.702	46.979	43.773	40.256	34.8	29.336	24.478	20.599	18.493	16.791	11.588
35	66.619	53.203	49.802	46.059	40.223	34.336	29.054	24.797	22.465	20.569	14.688
40	73.403	59.342	55.758	51.805	45.616	39.335	33.66	29.051	26.509	24.433	17.917
45	80.078	65.41	61.656	57.505	50.985	44.335	38.291	33.35	30.612	28.366	21.251
50	86.66	71.42	67.505	63.167	56.334	49.335	42.942	37.689	34.764	32.357	24.674
55	93.167	77.38	73.311	68.796	61.665	54.335	47.61	42.06	38.958	36.398	28.173
60	99.608	83.298	79.082	74.397	66.981	59.335	52.294	46.459	43.188	40.482	31.738
65	105.988	89.177	84.821	79.973	72.285	64.335	56.99	50.883	47.45	44.603	35.362
70	112.317	95.023	90.531	85.527	77.577	69.334	61.698	55.329	51.739	48.758	39.036
75	118.599	100.839	96.217	91.061	82.858	74.334	66.417	59.795	56.054	52.942	42.757
80	124.839	106.629	101.879	96.578	88.13	79.334	71.145	64.278	60.391	57.153	46.52
85	131.043	112.393	107.522	102.079	93.394	84.334	75.881	68.777	64.749	61.389	50.32
90	137.208	118.136	113.145	107.565	98.65	89.334	80.625	73.291	69.126	65.647	54.156
95	143.343	123.858	118.752	113.038	103.899	94.334	85.376	77.818	73.52	69.925	58.022
100	149.449	129.561	124.342	118.498	109.141	99.334	90.133	82.358	77.929	74.222	61.918

Figura II.1. Valores críticos de la distribución chi cuadrado

# Apéndice III

## Parámetros iniciales

Sheet: Parametros Iniciales

	A	B	C	D	E	F	G	H
1	a	b	alfa	beta	e	d	s	q
2	2	4	0,9	0,6	0,1	6	0	
3								
4	Distribución Uniforme		Contracción		Esta en el Visual Stop			Esta en el Visual Stop
5								
6	Continua	Discreta	Real	Entero				

Figura III.1. Spreadsheets: Parámetros iniciales I

	I	J	K	L	M
		X(Reales)	Y(Enteros)	n+1	m
Llegadas		5	c1 8	4	3
s1		12	c2 8		
s2		4	c3 8	Dimensión	
s3		4			
		Variables de decisión		Real	Entero

↑
↑  
 Tiempo de servicio      Capacidades

Figura III.2. Spreadsheets: Parámetros iniciales II

N	O	P	Q	S	T	U
li (Reales)	hi (Reales)	li (enteros)	hi (enteros)		K	
		c1 1	50		1000000	
s1 2	200	c2 1	300			
s2 2	150	c3 1	100	Constante de Penalización		
s3 2	100					
X(Reales)		Y(Reales)				

}  
 Restricción no lineal

Figura III.3. Spreadsheets: Parámetros iniciales II



## Apéndice IV

### Valores de distribución y función de desempeño

Sheet: Funcion menor						
	G	H	I	J	K	M
1	Distribucion Real		Distribucion Entera		Tiempo Promedio Menor	Funcion Objetivo (kth)
2	Distribucion Start Point		Distribucion Cola 1			
3	Distribucion Servidor 1		Distribucion Cola 2			
4	Distribucion Servidor 2		Distribucion Cola 3			
5	Distribucion Servidor 3	Reales		Enteros		
6						Mejora de la Función de
7			Últimos valores otorgados por el MIRPSA	Función de desempeño óptimo		desempeño en las kth

Figura IV.1. Spreadsheets:Función menor

Sheet: FUNCION							
	A	B	C	G	K	L	M
1	Servidor 1	Servidor 2	Servidor 3	Contadores	Tiempo Prom	Cientes no a	
2	Cliente 1						
3	Cliente 2						
4	Cliente 3						
5	Cliente 4				Evaluación de la F.O		
6	Cliente n						
7			Tiempo de Servicio				

↑
↑

Clientes atendidos por puntos de prueba "q"      Clientes no atendidos por puntos de prueba "q"

Figura IV.2. Spreadsheets: Función

## Apéndice V

### Tabla de Resultados

Sheet: Resultados

	A	B	C	D	E	F	G
1	Vk Reales	Vc Reales	Vz Reales		Vk Enteros	Vc Enteros	Vz Enteros
2	5	5	0		8	8	0
3	12	12	0		8	8	0
4	4	4	0		8	8	0
5	4	4	0				Patrón aleatorio
6		Reales	Patrón aleatorio actual Real			Enteros	actual Entero

Último patrón otorgado por el MIRPSA

Figura V.1. Spreadsheets: Resultados I

I	J	K	L	M	N
xi-hi	li-yi	yi-hi	Penalizacion	Valor de Penalizacion	Ultima evaluacion
0	0	0	0	0	0
0	0	0			
0	0	0			
0					
Real	Entero			P(z) anterior	P(z) evaluado

Evaluación de restricción      Penalización Actual "P(z)"

Figura V.2. Spreadsheets: Resultados II

# Apéndice VI

## Penalización I

```
-- PENALIZACION
-- Funcion de Penalizacion
-- Lectura de Valores Actules
-- SET Valores Actuales[1,1] = "X Reales"
-- SET Valores Actuales[2,1] = "Y Enteros"
-- PR: Penalizacion Real
-- PE: Penalizacio Entero
-- SET i PR = 2
-- SET j PE = 2
-- Intervalos Reales
-- Parametros Iniciales[12,2]: n (Reales)
-- WHILE i PR <= Parametros Iniciales[12,2]+1
--   -- Resultado[8,i PR]: li-X (Reales)
--   -- SET Resultados[8,i PR] = Parametros Iniciales[14,i PR]-Valores Actuales[1,i PR]
--   -- Resultado[9,i PR]: X-hi (Reales)
--   -- SET Resultados[9,i PR] = Valores Actuales[1,i PR]-Parametros Iniciales[15,i PR]
--   -- IF Resultados[8,i PR] <= 0
--   --   -- SET Resultados[8,i PR] = 0
--   -- ELSE
--   --   -- Parametros Iniciales[20,2]: K Penalizacion = 1000000
--   --   -- SET Resultados[8,i PR] = Resultados[8,i PR]*Parametros Iniciales[20,2]
--   -- IF Resultados[9,i PR] <= 0
--   --   -- SET Resultados[9,i PR] = 0
--   -- ELSE
--   --   -- Parametros Iniciales[20,2]: K Penalizacion = 1000000
--   --   -- SET Resultados[9,i PR] = Resultados[9,i PR]*Parametros Iniciales[20,2]
--   -- SET Resultados[12,2] = [Resultados[12,2]+Resultados[8,i PR]]+Resultados[9,i PR]
--   -- SET i PR = i PR+1
```

Figura VI.1. Visual Logic: Penalización I

## Apéndice VII

### Penalización II

```
--- Intervalos Enteros
--- Parametros Iniciales[13,2]: m (Enteros)
--- WHILE j PE <= Parametros Iniciales[13,2]+1
    --- Resultado[10,j PE]: li-Y (Enteros)
    --- SET Resultados[10,j PE] = Parametros Iniciales[16,j PE]-Valores Actuales[2,j PE]
    --- Resultado[11,j PE]: Y-hi (Enteros)
    --- SET Resultados[11,j PE] = Valores Actuales[2,j PE]-Parametros Iniciales[17,j PE]
    --- IF Resultados[10,j PE] <= 0
        --- SET Resultados[10,j PE] = 0
    --- ELSE
        --- Parametros Iniciales[20,2]: K Penalizacion = 1000000
        --- SET Resultados[10,j PE] = Resultados[10,j PE]*Parametros Iniciales[20,2]
    --- IF Resultados[11,j PE] <= 0
        --- SET Resultados[11,j PE] = 0
    --- ELSE
        --- Parametros Iniciales[20,2]: K Penalizacion = 1000000
        --- SET Resultados[11,j PE] = Resultados[11,j PE]*Parametros Iniciales[20,2]
    --- Reasultado[12,2]: Penalizacion Total
    --- SET Resultados[12,2] = Resultados[12,2]+[Resultados[10,j PE]+Resultados[11,j PE]]
    --- SET j PE = j PE+1
```

Figura VII.1. Visual Logic: Penalización II

## Apéndice VIII

### Reset Logic I

```
[-]-- Reset Logic
    |-- Obeeyed just after all simulation objects are initialized at time zero
    |-- Cargar valores actuales
    |-- VAR: Valores Actuales Reales
    |-- VAE: Variables Actuales Enteros
    |-- SET i VAR = 2
    |-- SET j VAE = 2
    |-- Parametros Iniciales [12,2]: n (Reales)
    |-- Parametros Iniciales [10,i VAR]: X(Reales)
    |-- Traslado de los Parametros Iniciales X (Reales) a los Valores Actuales
    [-]-- WHILE i VAR <= Parametros Iniciales[12,2]+1
        |-- SET Valores Actuales[1,i VAR] = Parametros Iniciales[10,i VAR]
        |-- SET i VAR = i VAR+1
    |-- Parametros Iniciales [13,2]: m (Enteros)
    |-- Parametros Iniciales [11,j VAE]: Y(Enteros)
    |-- Traslado de los Parametros Iniciales Y (Enteros) a los Valores Actuales
    [-]-- WHILE j VAE <= Parametros Iniciales[13,2]+1
        |-- SET Valores Actuales[2,j VAE] = Parametros Iniciales[11,j VAE]
        |-- SET j VAE = j VAE+1
    |-- CALL PENALIZACION
    [-]-- IF Resultados[12,2] > 0
        |-- Display Message Penalizado
    [-]-- ELSE
        |-- Display Message No Penalizado
```

Figura VIII.1. Visual Logic: Reset I

## Apéndice IX

### Reset Logic II

```
-- SET i ALR = 2
-- SET j ALE = 2
-- SET i ALR Vz = 2
-- SET j ALE Vz = 2
-- WHILE i ALR <= Parametros Iniciales[12,2]+1
--   -- Resultado[1,i ALR]: Vk (Reales)
--   -- SET Resultados[1,i ALR] = Parametros Iniciales[10,i ALR]
--   -- Resultados[2,i ALR]: Vc (Reales)
--   -- Vc (Reales)= Vk (Reales)
--   -- SET Resultados[2,i ALR] = Resultados[1,i ALR]
--   -- SET i ALR = i ALR+1
-- WHILE j ALE <= Parametros Iniciales[13,2]+1
--   -- Resultado[5,j ALE]: Vk (Enteros)
--   -- SET Resultados[5,j ALE] = Parametros Iniciales[11,j ALE]
--   -- Resultados[6,j ALE]: Vc (Enteros)
--   -- Vc (Enteros)= Vk (Enteros)
--   -- SET Resultados[6,j ALE] = Resultados[5,j ALE]
--   -- SET j ALE = j ALE+1
-- Inicializacion de los parametros
-- SET Parametros Iniciales[2,2] = 4
-- SET Parametros Iniciales[1,2] = 2
-- SET Parametros Iniciales[7,2] = 0
-- SET Parametros Iniciales[6,2] = Parametros Iniciales[1,2]+Parametros Iniciales[2,2]
-- LOOP 1 >>> Loop VF R >>> Parametros Iniciales[12,2]
--   -- SET Valores de la Funcion[1,1+Loop VF R] = Parametros Iniciales[10,1+Loop VF R]
-- LOOP 1 >>> Loop VF Z >>> Parametros Iniciales[13,2]
--   -- SET Valores de la Funcion[2,1+Loop VF Z] = Parametros Iniciales[11,1+Loop VF Z]
```

Figura IX.1. Visual Logic: Reset II

# Apéndice X

## Reset Logic III

```
-- Inicializar
[-]-- LOOP 1 >>> Loop F s1 >>> Inicializar las Tablas[1,2]
    |-- SET FUNCION[1,1+Loop F s1] = 0
[-]-- LOOP 1 >>> Loop F s2 >>> Inicializar las Tablas[2,2]
    |-- SET FUNCION[2,1+Loop F s2] = 0
[-]-- LOOP 1 >>> Loop F s3 >>> Inicializar las Tablas[3,2]
    |-- SET FUNCION[3,1+Loop F s3] = 0
    |-- SET FUNCION[4,1+Loop F s3] = 0
-- SET Inicializar las Tablas[1,2] = 0
-- SET Inicializar las Tablas[2,2] = 0
-- SET Inicializar las Tablas[3,2] = 0
-- HC FUNCION
[-]-- LOOP 1 >>> Loop VF c3 >>> Inicializar las Tablas[4,2]
    |-- SET FUNCION[6,Loop VF c3+1] = 0
    |-- SET FUNCION[7,Loop VF c3+1] = 0
    |-- SET FUNCION[8,Loop VF c3+1] = 0
    |-- SET FUNCION[10,1+Loop VF c3] = 0
    |-- SET FUNCION[11,1+Loop VF c3] = 0
    |-- SET FUNCION[12,1+Loop VF c3] = 0
-- SET Inicializar las Tablas[4,2] = 0
```

**Figura X.1.** Visual Logic: Reset III

## Apéndice XI

### Reset Logic IV

```
-- HC Funcion menor
[ ]-- LOOP 1 >>> Loop Fm R >>> Parametros Iniciales[12,2]
    -- SET Funcion menor[8,1+Loop Fm R] = 0
[ ]-- LOOP 1 >>> Loop Fm Z >>> Parametros Iniciales[13,2]
    -- SET Funcion menor[10,1+Loop Fm Z] = 0
-- SET Funcion menor[11,2] = 0
[ ]-- LOOP 1 >>> Loop kth >>> 100
    -- SET Funcion menor[13,Loop kth+1] = 0
[ ]-- LOOP 1 >>> Loop FmC3B >>> 500
    -- SET Funcion menor[12,Loop FmC3B+1] = 0
-- HC Valores Actuales y Resultados
[ ]-- LOOP 1 >>> Loop VA R >>> Parametros Iniciales[12,2]
    -- SET Valores Actuales[5,1+Loop VA R] = 0
    -- SET Valores Actuales[9,1+Loop VA R] = 0
    -- SET Resultados[3,1+Loop VA R] = 0
[ ]-- LOOP 1 >>> Loop VA Z >>> Parametros Iniciales[13,2]
    -- SET Valores Actuales[6,1+Loop VA Z] = 0
    -- SET Valores Actuales[10,1+Loop VA Z] = 0
    -- SET Resultados[7,1+Loop VA Z] = 0
-- Tiempo promedio de espera en la cola
-- SET Tiempo promedio de espera en la cola[1,2] = 0
-- SET Tiempo promedio de espera en la cola[2,2] = 0
-- SET Tiempo promedio de espera en la cola[3,2] = 0
```

**Figura XI.1.** Visual Logic: Reset IV



## Apéndice XII

### Valores actuales para la penalización

```
▣-- VALORES ACTUALES VZ
  -- Transportar el valor de Vz a Valores Actuales
  -- VAR Vz: Valores Actuales Reales
  -- VAE Vz: Valores Actuales Enteros
  -- Contadores Vz
  -- SET i VAR Vz = 2
  -- SET j VAE Vz = 2
  -- Valores Actuales VZ Reales
  ▣-- WHILE i VAR Vz <= Parametros Iniciales[12,2]+1
    -- SET Valores Actuales[1,i VAR Vz] = Resultados[3,i VAR Vz]
    -- SET i VAR Vz = i VAR Vz+1
  -- Valores Actuales VZ Enteros
  ▣-- WHILE j VAE Vz <= Parametros Iniciales[13,2]+1
    -- SET Valores Actuales[2,j VAE Vz] = Resultados[7,j VAE Vz]
    -- SET j VAE Vz = j VAE Vz+1
```

**Figura XII.1.** Visual Logic: Valores Actuales

## Apéndice XIII

### Búsqueda Aleatoria VZ I

```

-- VALORES DE VZ
-- Vz Reales
-- SET Aleatorio Real = [[RANDOM[Semilla]*2]*Parametros Iniciales[1,2]]-Parametros Iniciales[1,2]
-- Contador: i llenado Vz R
-- IF BTA = 1
-- SET KA = [Valores de la Funcion[1,3]-Parametros Iniciales[14,3]]/ABS[Aleatorio Real]
-- SET KA S2 = [Valores de la Funcion[1,4]-Parametros Iniciales[14,4]]/ABS[Aleatorio Real]
-- SET KA S3 = [Valores de la Funcion[1,5]-Parametros Iniciales[14,5]]/ABS[Aleatorio Real]
-- IF VT s1 = 1
-- IF Tiempo promedio de espera en la cola[1,2] >= 60
-- SET AR N = ABS[Aleatorio Real]
-- SET Resultados[3,3] = [Resultados[2,3]-[KA*AR N]]
-- SET B cd R = 1
-- ELSE
-- SET AR N = ABS[Aleatorio Real]
-- SET Resultados[3,3] = [Resultados[2,3]-AR N]
-- ELSE IF B cd R <> 0
-- SET B cd R = B cd R+1
-- IF B cd R <= 10
-- SET AR N = ABS[Aleatorio Real]
-- SET Resultados[3,3] = [Resultados[2,3]-[KA*AR N]]
-- ELSE
-- SET B cd R = 0
-- ELSE
-- IF Tiempo promedio de espera en la cola[1,2] > 1
-- SET AR N = ABS[Aleatorio Real]
-- SET Resultados[3,3] = [Resultados[2,3]-AR N]
-- ELSE
-- SET Resultados[3,3] = Resultados[2,3]+Aleatorio Real

```

Figura XIII.1. Visual Logic: Valores de Vz I

## Apéndice XIV

### Búsqueda Aleatoria VZ II

```
--- Servidor 2
└─ IF VT s2 = 1
  └─ IF Tiempo promedio de espera en la cola[2,2] >= 60
    └─ SET AR P = ABS[Aleatorio Real]
    └─ SET Resultados[3,4] = [Resultados[2,4]-[KA S2*AR P]]
    └─ SET B cd R 2 = 1
  └─ ELSE
    └─ SET AR P = ABS[Aleatorio Real]
    └─ SET Resultados[3,4] = [Resultados[2,4]-AR P]
└─ ELSE IF B cd R 2 <> 0
  └─ SET B cd R 2 = B cd R 2+1
  └─ IF B cd R 2 <= 10
    └─ SET AR P = ABS[Aleatorio Real]
    └─ SET Resultados[3,4] = [Resultados[2,4]-[KA S2*AR P]]
  └─ ELSE
    └─ SET B cd R 2 = 0
└─ ELSE
  └─ IF Tiempo promedio de espera en la cola[2,2] > 1
    └─ SET AR P = ABS[Aleatorio Real]
    └─ SET Resultados[3,4] = [Resultados[2,4]-AR P]
  └─ ELSE
    └─ SET Resultados[3,4] = Resultados[2,4]+Aleatorio Real
```

Figura XIV.1. Visual Logic: Valores de Vz II

## Apéndice XV

### Búsqueda Aleatoria VZ III

```
--- Servidor 3
-- IF VT s3 = 1
  -- IF Tiempo promedio de espera en la cola[3,2] >= 60
    -- SET AR NP = ABS[Aleatorio Real]
    -- SET Resultados[3,5] = [Resultados[2,5]-[KA S3*AR NP]]
    -- SET B cd R 3 = 1
  -- ELSE
    -- SET AR NP = ABS[Aleatorio Real]
    -- SET Resultados[3,5] = [Resultados[2,5]-AR NP]
  -- ELSE IF B cd R 3 <> 0
    -- SET B cd R 3 = B cd R 3+1
    -- IF B cd R 3 <= 10
      -- SET AR NP = ABS[Aleatorio Real]
      -- SET Resultados[3,5] = [Resultados[2,5]-[KA S3*AR NP]]
    -- ELSE
      -- SET B cd R 3 = 0
  -- ELSE
    -- IF Tiempo promedio de espera en la cola[3,2] > 1
      -- SET AR NP = ABS[Aleatorio Real]
      -- SET Resultados[3,5] = [Resultados[2,5]-AR NP]
    -- ELSE
      -- SET Resultados[3,5] = [Resultados[2,5]+Aleatorio Real]
  -- SET Resultados[3,2] = Resultados[2,2]
-- ELSE
  -- SET Aleatorio Real = Aleatorio Real
  -- SET i llenado Vz R = 3
  -- WHILE i llenado Vz R <= Parametros Iniciales[12,2]+1
    -- Vz (Real)= Vc (Real)+ Aleatorio (-a,a)
    -- SET Resultados[3,i llenado Vz R] = Resultados[2,i llenado Vz R]+Aleatorio Real
    -- SET i llenado Vz R = i llenado Vz R+1
```

Figura XV.1. Visual Logic: Valores de Vz III

## Apéndice XVI

### Búsqueda Aleatoria VZ IV

```
-- Cola 2
IF B S1C D = 1
  SET AE S1N = ABS[Aleatorio Entero]
  SET Resultados[7,3] = [Resultados[6,3]+AE S1N]
  SET B cd 2 = 1
ELSE
  IF B cd 2 <> 0
    SET B cd 2 = B cd 2+1
    IF B cd 2 <= 10
      SET AE S1N = ABS[Aleatorio Entero]
      SET Resultados[7,3] = [Resultados[6,3]+AE S1N]
    ELSE
      SET B cd 2 = 0
  ELSE
    SET AE S1N = ABS[Aleatorio Entero]
    SET Resultados[7,3] = [Resultados[6,3]-[2*AE S1N]]
-- Cola 3
IF B S2C D = 1
  SET AE S2N = ABS[Aleatorio Entero]
  SET Resultados[7,4] = [Resultados[6,4]+AE S2N]
  SET B cd3 = 1
ELSE
  IF B cd3 <> 0
    SET B cd3 = B cd3+1
    IF B cd3 <= 10
      SET AE S2N = ABS[Aleatorio Entero]
      SET Resultados[7,4] = [Resultados[6,4]+AE S2N]
    ELSE
      SET B cd3 = 0
  ELSE
    SET AE S2N = ABS[Aleatorio Entero]
    SET Resultados[7,4] = [Resultados[6,4]-[2*AE S2N]]
ELSE
  SET Aleatorio Entero = Aleatorio Entero
  Contador: j llenado Vz E
  SET j llenado Vz E = 2
  WHILE j llenado Vz E <= Parametros Iniciales[13,2]+1
    Vz (Entero)= Vc (Entero)+ Aleatorio (-max(1,b),max(1,b))
    SET Resultados[7,j llenado Vz E] = Resultados[6,j llenado Vz E]+Aleatorio Entero
    SET j llenado Vz E = j llenado Vz E+1
```

Figura XVI.1. Visual Logic: Valores de Vz IV

## Apéndice XVII

### Stop Run Logic

```
-- Stop Run Logic
-- Warning: Obeeyed every time the user clicks the STOP button (at any simulation time)
-- SET stop = stop+1
-- for j = 1 to q
-- SET M nm = [[Parametros Iniciales[12,2]-1]+Parametros Iniciales[13,2]]*2
-- IF stop < M nm
-- Run Simulation 1000000000
-- IF stop = M nm
-- Contador de Iteracion
-- SET K (Iteracion) = K (Iteracion)+1
-- SET Funcion menor[13,K (Iteracion)+1] = Funcion menor[11,2]
-- If S = 1
-- IF Parametros Iniciales[7,2] = 1
-- SET Parametros Iniciales[7,2] = 0
-- LOOP 1 >>> Loop Vk a Vc R >>> Parametros Iniciales[12,2]
-- SET Resultados[2,1+Loop Vk a Vc R] = Resultados[1,1+Loop Vk a Vc R]
-- LOOP 1 >>> Loop Vk a Vc Z >>> Parametros Iniciales[13,2]
-- SET Resultados[6,1+Loop Vk a Vc Z] = Resultados[5,1+Loop Vk a Vc Z]
-- ELSE
-- Parametro distribucion Uniforme Continuo
-- SET Parametros Iniciales[1,2] = Parametros Iniciales[3,2]*Parametros Iniciales[1,2]
-- Parametro distribucion Uniforme Discreto
-- SET Parametros Iniciales[2,2] = Parametros Iniciales[4,2]*Parametros Iniciales[2,2]
-- d= Parametro distribucion Uniforme Continuo + Parametro distribucion Uniforme Discreta
-- SET Parametros Iniciales[6,2] = Parametros Iniciales[1,2]+Parametros Iniciales[2,2]
-- While d mayor que e
-- IF Parametros Iniciales[6,2] > Parametros Iniciales[5,2]
-- SET stop = 0
-- Run Simulation 1000000000
```

Figura XVII.1. Visual Logic: Stop Run

## Apéndice XVIII

### Time Check Logic I

```
-- Time Check Logic
-- Repeated at a set time interval
-- SET DSII = 0
-- SET hour = hour+1
-- IF hour = 6
  -- Guardar los valores de la distribucion anterior
  -- LOOP 1 >>> Loop D anterior R2 >>> Parametros Iniciales[12,2]
  -- SET Valores Actuales[9,1+Loop D anterior R2] = Valores Actuales[5,1+Loop D anterior R2]
  -- LOOP 1 >>> Loop D anterior Z2 >>> Parametros Iniciales[13,2]
  -- SET Valores Actuales[10,1+Loop D anterior Z2] = Valores Actuales[6,1+Loop D anterior Z2]
  -- LOOP 1 >>> Loop D anterior R >>> Parametros Iniciales[12,2]
  -- SET Valores Actuales[5,1+Loop D anterior R] = Valores de la Funcion[1,1+Loop D anterior R]
  -- LOOP 1 >>> Loop D anterior Z >>> Parametros Iniciales[13,2]
  -- SET Valores Actuales[6,1+Loop D anterior Z] = Valores de la Funcion[2,1+Loop D anterior Z]
  -- Evaluacion de la funcion
  -- IF C3 B >= 5
    -- IF FUNCION[11,C3 B+1] < Funcion menor[11,2]
      -- SET Funcion menor[11,2] = FUNCION[11,C3 B+1]
      -- SET Funcion menor[12,C3 B+1] = Funcion menor[11,2]
      -- SET Parametros Iniciales[7,2] = 1
      -- LOOP 1 >>> Loop D menor R2 >>> Parametros Iniciales[12,2]
      -- SET Funcion menor[8,1+Loop D menor R2] = Valores Actuales[9,1+Loop D menor R2]
      -- LOOP 1 >>> Loop D menor Z2 >>> Parametros Iniciales[13,2]
      -- SET Funcion menor[10,1+Loop D menor Z2] = Valores Actuales[10,1+Loop D menor Z2]
      -- LOOP 1 >>> Loop Distribucion Rsc >>> Parametros Iniciales[12,2]
      -- SET Resultados[1,1+Loop Distribucion Rsc] = Funcion menor[8,1+Loop Distribucion Rsc]
      -- LOOP 1 >>> Loop Distribucion Zsc >>> Parametros Iniciales[13,2]
      -- SET Resultados[5,1+Loop Distribucion Zsc] = Funcion menor[10,1+Loop Distribucion Zsc]
    -- ELSE
      -- SET Funcion menor[11,2] = Funcion menor[11,2]
  -- ELSE
    -- SET Band = 1
```

Figura XVIII.1. Visual Logic: Time Check Logic I

## Apéndice XIX

### Time Check Logic II

```

|-- ELSE IF Resultados[9,3] > 0
    |-- SET Valores de la Funcion[1,3] = Parametros Iniciales[15,3]
    |-- SET Valores Actuales[1,3] = Parametros Iniciales[15,3]
    |-- SET Resultados[9,3] = 0
|-- ELSE
    |-- SET Valores de la Funcion[1,3] = Valores Actuales[1,3]
|-- IF Resultados[8,4] > 0
    |-- SET Valores de la Funcion[1,4] = Parametros Iniciales[14,4]
    |-- SET Valores Actuales[1,4] = Parametros Iniciales[14,4]
    |-- SET Resultados[8,4] = 0
|-- ELSE IF Resultados[9,4] > 0
    |-- SET Valores de la Funcion[1,4] = Parametros Iniciales[15,4]
    |-- SET Valores Actuales[1,4] = Parametros Iniciales[15,4]
    |-- SET Resultados[9,4] = 0
|-- ELSE
    |-- SET Valores de la Funcion[1,4] = Valores Actuales[1,4]
|-- IF Resultados[8,5] > 0
    |-- SET Valores de la Funcion[1,5] = Parametros Iniciales[14,5]
    |-- SET Valores Actuales[1,5] = Parametros Iniciales[14,5]
    |-- SET Resultados[8,5] = 0
|-- ELSE IF Resultados[9,5] > 0
    |-- SET Valores de la Funcion[1,5] = Parametros Iniciales[15,5]
    |-- SET Valores Actuales[1,5] = Parametros Iniciales[15,5]
    |-- SET Resultados[9,5] = 0
|-- ELSE
    |-- SET Valores de la Funcion[1,5] = Valores Actuales[1,5]

```

**Figura XIX.1.** Visual Logic: Time Check Logic II



## Apéndice XX

### Time Check Logic III

```
-- Siguiente busqueda aleatoria y evaluacion
-- SET BTA = 1
-- SET B CA = 1
-- CALL VALORES DE VZ
-- CALL VALORES ACTUALES VZ
-- CALL PENALIZACION
-- IF Resultados[12,2] <> 0
--   X Display Message Penalizado
--   SET Resultados[14,2] = Resultados[13,2]
--   SET Resultados[13,2] = Resultados[12,2]
--   Frontera menor Real
--   IF Resultados[8,2] > 0
--     SET Valores de la Funcion[1,2] = Parametros Iniciales[14,2]
--     SET Valores Actuales[1,2] = Parametros Iniciales[14,2]
--     SET Resultados[8,2] = 0
--   ELSE IF Resultados[9,2] > 0
--     SET Valores de la Funcion[1,2] = Parametros Iniciales[15,2]
--     SET Valores de la Funcion[1,2] = Parametros Iniciales[15,2]
--     SET Valores Actuales[1,2] = Parametros Iniciales[15,2]
--     SET Resultados[9,2] = 0
--   ELSE
--     SET Valores de la Funcion[1,2] = Valores Actuales[1,2]
--   IF Resultados[8,3] > 0
--     SET Valores de la Funcion[1,3] = Parametros Iniciales[14,3]
--     SET Valores Actuales[1,3] = Parametros Iniciales[14,3]
--     SET Resultados[8,3] = 0
```

Figura XX.1. Visual Logic: Time Check Logic III

## Apéndice XXI

### Time Check Logic IV

```
-- Frontera menor Entero
-- IF Resultados[10,2] <> 0
--   SET Valores de la Funcion[2,2] = Parametros Iniciales[16,2]
--   SET Valores Actuales[2,2] = Parametros Iniciales[16,2]
--   SET Resultados[10,2] = 0
-- ELSE IF Resultados[11,2] <> 0
--   SET Valores de la Funcion[2,2] = Parametros Iniciales[17,2]
--   SET Valores Actuales[2,2] = Parametros Iniciales[17,2]
--   SET Resultados[11,2] = 0
-- ELSE
--   SET Valores de la Funcion[2,2] = Valores Actuales[2,2]
-- IF Resultados[10,3] <> 0
--   SET Valores de la Funcion[2,3] = Parametros Iniciales[16,3]
--   SET Valores Actuales[2,3] = Parametros Iniciales[16,3]
--   SET Resultados[10,3] = 0
-- ELSE IF Resultados[11,3] <> 0
--   SET Valores de la Funcion[2,3] = Parametros Iniciales[17,3]
--   SET Valores Actuales[2,3] = Parametros Iniciales[17,3]
--   SET Resultados[11,3] = 0
-- ELSE
--   SET Valores de la Funcion[2,3] = Valores Actuales[2,3]
-- IF Resultados[10,4] <> 0
--   SET Valores de la Funcion[2,4] = Parametros Iniciales[16,4]
--   SET Valores Actuales[2,4] = Parametros Iniciales[16,4]
--   SET Resultados[10,4] = 0
-- ELSE IF Resultados[11,4] <> 0
--   SET Valores de la Funcion[2,4] = Parametros Iniciales[17,4]
--   SET Valores Actuales[2,4] = Parametros Iniciales[17,4]
--   SET Resultados[11,4] = 0
```

Figura XXI.1. Visual Logic: Time Check Logic IV

## Apéndice XXII

### Time Check Logic V

```
-- ELSE
  --- SET Valores de la Funcion[2,4] = Valores Actuales[2,4]
  --- SET Resultados[12,2] = 0
  --- SET hour = 0
  --- Stop Simulation
-- ELSE
  --- SET Resultados[14,2] = Resultados[13,2]
  --- SET Resultados[13,2] = Resultados[12,2]
  --- LOOP 1 >>> Loop VACTUALES R S2 >>> Parametros Iniciales[12,2]
    --- SET Valores de la Funcion[1,1+Loop VACTUALES R S2] = Valores Actuales[1,1+Loop VACTUALES R S2]
    --- SET DSII = 1
    --- SET hour = 0
  --- LOOP 1 >>> Loop VACTUALES Z S2 >>> Parametros Iniciales[13,2]
    --- SET Valores de la Funcion[2,1+Loop VACTUALES Z S2] = Valores Actuales[2,1+Loop VACTUALES Z S2]
    --- Stop Simulation
-- Llenado de la hoja de calculo la funcion
-- SET VF c3 = VF c3+1
-- SET Inicializar las Tablas[4,2] = VF c3
-- SET c3pp = Contador tiempo Act 3-c3p
-- SET FUNCION[7,VF c3+1] = c3pp
-- LOOP 1 >>> Loop C3pp >>> c3pp
  --- SET FUNCION[6,VF c3+1] = FUNCION[6,VF c3+1]+FUNCION[4,[1+Loop C3pp]+c3p]
  --- SET FUNCION[10,1+VF c3] = FUNCION[10,1+VF c3]+FUNCION[3,[1+Loop C3pp]+c3p]
-- SET c3p = Contador tiempo Act 3
```

Figura XXII.1. Visual Logic: Time Check Logic V

## Apéndice XXIII

### Time Check Logic VI

```
-- Servicio Promedio
-- SET FUNCION[11,1+VF C3] = [[0.5*FUNCION[3,1+c3p]]+[0.5*C NA]]+Resultados[14,2]
-- SET FUNCION[12,1+VF C3] = C NA
-- SET C3 B = C3 B+1
-- IF C3 B = 4
  -- SET Funcion menor[11,2] = FUNCION[11,C3 B+1]
-- IF C3 B = 4
  -- SET Funcion menor[11,2] = FUNCION[11,C3 B+1]
  -- SET Funcion menor[12,C3 B+1] = Funcion menor[11,2]
  -- LOOP 1 >>> Loop D menor R >>> Parametros Iniciales[12,2]
    -- SET Funcion menor[8,1+Loop D menor R] = Valores Actuales[5,1+Loop D menor R]
  -- LOOP 1 >>> Loop D menor Z >>> Parametros Iniciales[13,2]
    -- SET Funcion menor[10,1+Loop D menor Z] = Valores Actuales[6,1+Loop D menor Z]
  -- LOOP 1 >>> Loop Distribucion Rsc >>> Parametros Iniciales[12,2]
    -- SET Resultados[1,1+Loop Distribucion Rsc] = Funcion menor[8,1+Loop Distribucion Rsc]
  -- LOOP 1 >>> Loop Distribucion Zsc >>> Parametros Iniciales[13,2]
    -- SET Resultados[5,1+Loop Distribucion Zsc] = Funcion menor[10,1+Loop Distribucion Zsc]
  -- SET Parametros Iniciales[7,2] = 1
-- ELSE
  -- SET Band = 1
-- Banderas para aleatoriedad
-- SET B ASC D = 0
-- SET B S1C D = 0
-- SET B S2C D = 0
-- SET B AST D = 0
-- SET VT s1 = 0
-- SET VT s2 = 0
-- SET VT s3 = 0
-- Limpieza
-- SET C NA = 0
```

Figura XXIII.1. Visual Logic: Time Check Logic VI

## Apéndice XXIV

### Actividad: Asesor I

```
[-] -- Asesor Action Logic
    -- SET Count S1 = Queue for Sevidor 1.Count Contents
    -- SET Count S2 = Queue for Servidor 2.Count Contents
    -- SET Count S3 = Queue for Servidor 3.Count Contents
    [-] -- IF Count S1 = Valores de la Funcion[2,2]
        -- SET VT s1 = 1
        -- SET B ASC D = 1
        -- SET B AST D = 1
        -- SET Tipo servicio 1 = 1
        [-] -- IF Count S2 = Valores de la Funcion[2,3]
            -- SET VT s2 = 1
            -- SET B S1C D = 1
            -- SET B AST D = 1
            -- SET Tipo servicio 2 = 1
            [-] -- IF Count S3 = Valores de la Funcion[2,4]
                -- SET Count = 2
                -- SET VT s3 = 1
                -- SET B S2C D = 1
                -- SET B AST D = 1
                -- SET Tipo servicio 3 = 1
            [-] -- ELSE
                -- SET Count = 1
        [-] -- ELSE IF Count S3 = Valores de la Funcion[2,4]
            -- SET Count = 1
            -- SET VT s3 = 1
            -- SET B S2C D = 1
            -- SET B AST D = 1
            -- SET Tipo servicio 3 = 1
```

Figura XXIV.1. Visual Logic: Asesor del modelo paralelo

## Apéndice XXV

### Actividad: Asesor II

```

|-- ELSE
  |-- SET Count = 1
|-- ELSE IF Count S2 = Valores de la Funcion[2,3]
  |-- SET VT s2 = 1
  |-- SET B S1C D = 1
  |-- SET B AST D = 1
  |-- SET Tipo servicio 2 = 1
  |-- IF Count S3 = Valores de la Funcion[2,4]
    |-- SET Count = 1
    |-- SET VT s3 = 1
    |-- SET B S2C D = 1
    |-- SET B AST D = 1
    |-- SET Tipo servicio 3 = 1
  |-- ELSE
    |-- SET Count = 1
|-- ELSE
  |-- IF Count S3 = Valores de la Funcion[2,4]
    |-- SET Count = 1
    |-- SET VT s3 = 1
    |-- SET B S2C D = 1
    |-- SET B AST D = 1
    |-- SET Tipo servicio 3 = 1
  |-- ELSE
    |-- SET Count = 1

```

Figura XXV.1. Visual Logic: Asesor del modelo paralelo

## Apéndice XXVI

### Modelo paralelo, mixto y retroalimentado

```
-- Salida Action Logic
-- Obtener Tiempo Promedio
-- SET Ts Salida = L2 Salida-L1
-- SET Variable tiempo Salida = Variable tiempo Salida+Ts Salida
-- SET Contador tiempo Salida = Contador tiempo Salida+1
-- IF B1 S = 0
--   SET Valores de la Funcion[17,2] = Variable tiempo Salida
--   SET Valores de la Funcion[17,3] = Contador tiempo Salida
--   SET FUNCION[5,Contador tiempo Salida+1] = Valores de la Funcion[17,2]/Valores de la Funcion[17,3]
--   SET B2 S = 0
-- ELSE
--   SET Valores de la Funcion[18,2] = Variable tiempo Salida
--   SET Valores de la Funcion[18,3] = Contador tiempo Salida
--   SET FUNCION[5,Contador tiempo Salida+1] = Valores de la Funcion[18,2]/Valores de la Funcion[18,3]
--   SET B1 S = 0
--   SET B2 S = 1
-- IF B2 S <> 1
--   SET B1 S = B1 S+1
-- SET Inicializar las Tablas[5,2] = Contador tiempo Salida
```

**Figura XXVI.1.** Visual Logic: Salida en el modelo paralelo, mixto y retroalimentado

## Apéndice XXVII

### Cambios en la Función objetivo

```
-- Servicio Promedio
-- SET FUNCION[11,1+VF c3] = [[0.5*FUNCION[5,1+c3p]]+[0.5*C NA]]+Resultados[14,2]
-- SET FUNCION[12,1+VF c3] = C NA
-- SET C3 B = C3 B+1
```

**Figura XXVII.1.** Visual Logic: Cambios en el Time Check para el modelo paralelo, mixto y retroalimentado

```
-- HC Valores de la Funcion
-- SET Valores de la Funcion[5,2] = 0
-- SET Valores de la Funcion[5,3] = 0
-- SET Valores de la Funcion[6,2] = 0
-- SET Valores de la Funcion[6,3] = 0
-- SET Valores de la Funcion[9,2] = 0
-- SET Valores de la Funcion[9,3] = 0
-- SET Valores de la Funcion[10,2] = 0
-- SET Valores de la Funcion[10,3] = 0
-- SET Valores de la Funcion[13,2] = 0
-- SET Valores de la Funcion[13,3] = 0
-- SET Valores de la Funcion[14,2] = 0
-- SET Valores de la Funcion[14,3] = 0
-- SET Resultados[14,2] = 0
-- SET Resultados[13,2] = 0
```

**Figura XXVII.2.** Visual Logic: Inicialización de Valores de la Función



## Apéndice XXVIII

### Salida de clientes del sistema

```
[-] Cientes no atendidos Action Logic
  -- SET C NA = C NA+1
  [-] IF Tipo Servicio = 1
    -- SET B ASC = 1
    -- SET B ASC D = 1
    -- SET B AST D = 1
    -- SET VT s1 = 1
  [-] ELSE
    -- SET B ASC = 0
  [-] IF Tipo Servicio = 2
    -- SET B S1C = 1
    -- SET B S1C D = 1
    -- SET B AST D = 1
    -- SET VT s2 = 1
  [-] ELSE
    -- SET B S1C = 0
  [-] IF Tipo Servicio = 3
    -- SET B S2C = 1
    -- SET B S2C D = 1
    -- SET B AST D = 1
    -- SET VT s3 = 1
  [-] ELSE
    -- SET B S2C = 0
```

Figura XXVIII.1. Visual Logic: Cientes no atendidos

## Apéndice XXIX

### Evaluación de las capacidades

```
[-] Queue for Activity 1 On Entry Logic
    -- Obeyed just after a work item enters the Queue
    -- Tamaño de la Cola 1
    -- SET Queue for Servidor 1.Max size = Valores de la Funcion[2,2]
```

**Figura XXIX.1.** Visual Logic: Capacidad 1

```
[-] Queue 1 On Entry Logic
    -- Obeyed just after a work item enters the Queue
    -- Tamaño de la Cola 2
    -- SET Queue for Servidor 2.Max size = Valores de la Funcion[2,3]
```

**Figura XXIX.2.** Visual Logic: Capacidad 2

```
[-] Queue 2 On Entry Logic
    -- Obeyed just after a work item enters the Queue
    -- Tamaño de la Cola 3
    -- SET Queue for Servidor 3.Max size = Valores de la Funcion[2,4]
```

**Figura XXIX.3.** Visual Logic: Capacidad 3

## Apéndice XXX

### Servidor 1 Logic

```
[-] Servidor 1 Action Logic
  -- Tiempo de Operacion de la Actividad 1
  -- Set Object Distribution Parameters  Servidor 1 , Operation Time , Exponential , Valores de la Funcion[1,3]
  -- Obtener Tiempo Promedio
  -- SET Tiempo promedio de espera en la cola[1,2] = Queue for Servidor 1.Average Queueing Time
  -- SET Ts Servidor 1 = L2 Servidor 1-L1
  -- SET Variable tiempo Act 1 = Variable tiempo Act 1+Ts Servidor 1
  -- SET Contador tiempo Act 1 = Contador tiempo Act 1+1
  [-] IF B1 = 0
    -- SET Valores de la Funcion[5,2] = Variable tiempo Act 1
    -- SET Valores de la Funcion[5,3] = Contador tiempo Act 1
    -- SET FUNCION[1,Contador tiempo Act 1+1] = Valores de la Funcion[5,2]/Valores de la Funcion[5,3]
    -- SET B2 = 0
  [-] ELSE
    -- SET Valores de la Funcion[6,2] = Variable tiempo Act 1
    -- SET Valores de la Funcion[6,3] = Contador tiempo Act 1
    -- SET FUNCION[1,Contador tiempo Act 1+1] = Valores de la Funcion[6,2]/Valores de la Funcion[6,3]
    -- SET B1 = 0
    -- SET B2 = 1
  [-] IF B2 <> 1
    -- SET B1 = B1+1
  -- SET Inicializar las Tablas[1,2] = Contador tiempo Act 1
```

Figura XXX.1. Visual Logic: Servidor 1

## Apéndice XXXI

### Servidor 2 Logic

```
[- Servidor 2 Action Logic
  -- Tiempo de Operacion de la Actividad 2
  -- Set Object Distribution Parameters   Servidor 2 , Operation Time , Exponential , Valores de la Funcion[1,4]
  -- Obtener Tiempo Promedio Actividad 2
  -- SET Tiempo promedio de espera en la cola[2,2] = Queue for Servidor 2.Average Queueing Time
  -- SET Ts Servidor 2 = L2 Servidor 2-L1
  -- SET Variable tiempo Act 2 = Variable tiempo Act 2+Ts Servidor 2
  -- SET Contador tiempo Act 2 = Contador tiempo Act 2+1
  -- Llenado de Valores de la Funcion
  [- IF B1 2 = 0
    -- SET Valores de la Funcion[9,2] = Variable tiempo Act 2
    -- SET Valores de la Funcion[9,3] = Contador tiempo Act 2
    -- SET FUNCION[2,Contador tiempo Act 2+1] = Valores de la Funcion[9,2]/Valores de la Funcion[9,3]
    -- SET B2 2 = 0
  [- ELSE
    -- SET Valores de la Funcion[10,2] = Variable tiempo Act 2
    -- SET Valores de la Funcion[10,3] = Contador tiempo Act 2
    -- SET FUNCION[2,Contador tiempo Act 2+1] = Valores de la Funcion[10,2]/Valores de la Funcion[10,3]
    -- SET B1 2 = 0
    -- SET B2 2 = 1
  [- IF B2 2 <> 1
    -- SET B1 2 = B1 2+1
  -- Fin del llenado
  -- SET Inicializar las Tablas[2,2] = Contador tiempo Act 2
```

Figura XXXI.1. Visual Logic: Servidor 2

## Apéndice XXXII

### Servidor 3 Logic

```
[-] Servidor 3 Action Logic
  -- Tiempo de Operacion de la Actividad 3
  -- Set Object Distribution Parameters   Servidor 3 , Operation Time , Exponential , Valores de la Funcion[1,5]
  -- Obtener Tiempo Promedio
  -- SET Tiempo promedio de espera en la cola[3,2] = Queue for Servidor 3.Average Queueing Time
  -- SET Ts Servidor 3 = L2 Servidor 3-L1
  -- SET Variable tiempo Act 3 = Variable tiempo Act 3+Ts Servidor 3
  -- SET Contador tiempo Act 3 = Contador tiempo Act 3+1
  -- Llenar Valores de la funcion
  [-] IF B1 3 = 0
    -- SET Valores de la Funcion[13,2] = Variable tiempo Act 3
    -- SET Valores de la Funcion[13,3] = Contador tiempo Act 3
    -- SET FUNCION[3,Contador tiempo Act 3+1] = Valores de la Funcion[13,2]/Valores de la Funcion[13,3]
    -- SET B2 3 = 0
  [-] ELSE
    -- SET Valores de la Funcion[14,2] = Variable tiempo Act 3
    -- SET Valores de la Funcion[14,3] = Contador tiempo Act 3
    -- SET FUNCION[3,Contador tiempo Act 3+1] = Valores de la Funcion[14,2]/Valores de la Funcion[14,3]
    -- SET B1 3 = 0
    -- SET B2 3 = 1
  [-] IF B2 3 <> 1
    -- SET B1 3 = B1 3+1
  -- Fin de Llenado de los valores de la funcion
  -- Tiempo Promedio Total
  -- SET FUNCION[4,Contador tiempo Act 3+1] = [FUNCION[1,Contador tiempo Act 3+1]+FUNCION[2,Contador tiempo Act 3+1]]
    -- +FUNCION[3,Contador tiempo Act 3+1]
  -- SET Inicializar las Tablas[3,2] = Contador tiempo Act 3
```

**Figura XXXII.1.** Visual Logic: Servidor 3