



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON
LAS TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES**

German Antonio Oliva Muralles

Asesorado por el Ing. Ariel Jesúa Hernández Monterroso

Guatemala, noviembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON LAS
TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

GERMAN ANTONIO OLIVA MURALLES

ASESORADO POR EL ING. ARIEL JESÚA HERNÁNDEZ MONTERROSO

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Herman Igor Véliz Linares
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON LAS TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 11 de octubre de 2016.

German Antonio Oliva Muralles


Guatemala, 14 de mayo de 2017

Señor Ing. Carlos Azurdia
Carrera de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Respetable Ing. Azurdia:

El motivo de la presente es para informarle que como asesor del estudiante German Antonio Oliva Muralles identificado con el CUI 2459 43706 0101 y carné 201213189, he procedido a revisar el trabajo de graduación titulado "SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON LAS TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES" y que de acuerdo a mi criterio el mismo se encuentra concluido. He tenido comunicación periódica con el estudiante y luego de haber revisado cuidadosamente el trabajo, considero que cumple con los requisitos de calidad y profesionalismo que deben caracterizar a un futuro profesional de la informática.

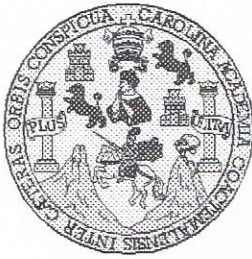
Sin otro particular me suscribo de usted.


Ariel Jesús Hernández Monterroso
Colegiado No. 13676
Ingeniero en Ciencias y Sistemas

Atentamente,

Ing. Ariel Jesús Hernández Monterroso

Asesor



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 24 de Mayo de 2017


Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **GERMAN ANTONIO OLIVA MURALLES** con carné 201213189 y CUI 2459 43706 0101, titulado “**SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON LAS TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES**”, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
E
L
A

D
E

I
N
G
E
N
I
E
R
Í
A

E
N

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24188000 Ext. 1534

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON LAS TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES”**, realizado por el estudiante **GERMAN ANTONIO OLIVA MURALLES** aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Ing. Marlon Antonio Pérez Türk
Director

Escuela de Ingeniería en Ciencias y Sistemas

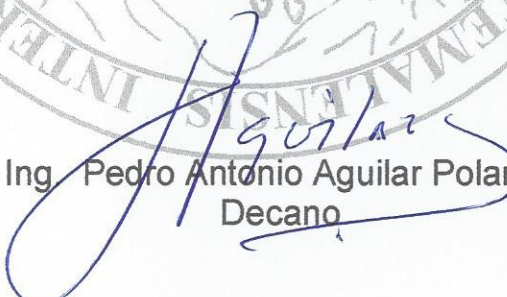


Guatemala, 09 de noviembre de 2017



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **SISTEMA INTERACTIVO PARA LA TOMA DE DECISIONES RELACIONADAS CON LAS TÉCNICAS DE LA INVESTIGACIÓN DE OPERACIONES**, presentado por el estudiante universitario: **German Antonio Oliva Muralles**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.


Ing. Pedro Antonio Aguilar Polanco
Decano



Guatemala, noviembre de 2017

/cc

ACTO QUE DEDICO A:

Dios	Por su amor, bondad y gracia. De ti me viene todo y a ti debe volver.
Mis padres	Jorge Oliva y Margarita Muralles, su amor será siempre mi inspiración.
Mi hermana	Margarita Oliva, por su cariño y apoyo en cada etapa de mi vida.
Mi novia	Cecilia Cobar, por ser una importante influencia en mi carrera, entre tantas cosas.
Mi familia	Por siempre creer en mí y brindarme su afecto en todo momento.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser mi casa de estudio y una importante influencia en mi carrera.
Facultad de Ingeniería	Por brindarme experiencia, conocimiento y amistades para crecer profesionalmente.
Mis amigos de la facultad	Sergio Herrera, Susana González, Jorge Gutiérrez, Brayan Yaquián, Kevin Pinto, Roselin del Cid, Carlos Rodríguez, Fernando Samayoa y René Ardón.
Ing. Vera Marroquín	Por ser una gran amiga y consejera en los últimos años de mi carrera.
Ing. José González	Por ser gran catedrático y un amigo a lo largo de mi carrera.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO	VII
RESUMEN.....	IX
OBJETIVOS.....	XI
INTRODUCCIÓN	XIII
1. TEORÍAS DE LA INFORMACIÓN	1
1.1. Teoría de la complejidad	1
1.1.1. Complejidad en el espacio.....	2
1.1.2. Complejidad en el tiempo	2
1.1.3. Notación O-Grande.....	3
1.2. Modelado de redes.....	3
1.2.1. CPM/PERT	4
1.3. Cadenas de Markov	8
1.3.1. Cadenas de Markov homogéneas	8
1.4. Teoría de colas.....	9
1.4.1. Proceso de entrada o llegada	9
1.4.2. Proceso de salida o de servicio	10
1.4.3. Líneas de espera o colas.....	11
1.4.4. Proceso de nacimiento y muerte	11
1.4.4.1. Modelo M/M/s	12
1.4.4.2. Modelo M/M/s/K.....	12
1.4.5. Otros modelos	12
1.4.5.1. Modelo M/G/1	12
1.4.5.2. Modelo M/D/s.....	13

	1.4.5.3.	Modelo M/Ek/s.....	13
1.5.		Teoría de inventarios.....	13
	1.5.1.	Tipos de costo	14
		1.5.1.1. Costo de pedido y organización	14
		1.5.1.2. Costo de compra unitario	14
		1.5.1.3. Costo de retención o posesión	15
		1.5.1.4. Costo de escasez o por faltantes	15
	1.5.2.	Modelo de lote económico.....	16
2.		IDENTIFICACIÓN DEL PROBLEMA	17
	2.1.	Antecedentes	17
	2.2.	Mercado objetivo.....	18
	2.3.	<i>Benchmarking</i>	18
3.		DISEÑO PRELIMINAR	21
	3.1.	Estructura general del sistema.....	21
	3.2.	Estructura del módulo de cadenas markovianas.....	23
	3.3.	Estructura del módulo de redes	25
	3.4.	Estructura del módulo de teoría de inventarios	31
	3.5.	Estructura del módulo de teoría de colas	33
4.		DOCUMENTACIÓN.....	35
	4.1.	Herramientas.....	35
	4.2.	Hardware.....	35
	4.3.	Software	35
	4.4.	Manual de desarrollo.....	36
		4.4.1. Herramientas.....	36
		4.4.2. Configuración del entorno de desarrollo.....	37
	4.5.	Secuencia de la solución.....	38

CONCLUSIONES	41
RECOMENDACIONES	43
BIBLIOGRAFÍA.....	45
APÉNDICE.....	47

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Búsqueda de la ruta crítica.....	5
2.	Estructura general del sistema.....	22
3.	Formulario del módulo de cadenas markovianas.....	23
4.	Reporte del módulo de cadenas markovianas.....	24
5.	Delimitación del problema del módulo de redes.....	26
6.	Formulario para el modelo de redes utilizando CPM.....	27
7.	Formulario para el modelo de redes utilizando PERT.....	28
8.	Reporte del módulo de redes.....	29
9.	Reporte detallado del módulo de redes.....	30
10.	Formulario del módulo de teoría de inventarios.....	31
11.	Reporte del módulo de teoría de inventarios.....	32
12.	Formulario del módulo de teoría de colas.....	33
13.	Reporte del módulo de teoría de colas.....	34
14.	Comandos de configuración y despliegue.....	37
15.	Menú principal.....	38
16.	Funcionamiento del módulo de Markov.....	39
17.	Funcionamiento del módulo de teoría de inventarios.....	40

GLOSARIO

Cordova	Es un <i>framework</i> de trabajo para el desarrollo de páginas web y aplicaciones móviles de forma integrada.
CPM	Método de la ruta crítica; el cual consiste en una metodología empleada para la optimización de problemas de redes.
CSS	Siglas en inglés de hojas de estilo en cascada. Consiste en un lenguaje, con el cual se define el formato general para diferentes tipos de fuentes.
<i>Framework</i>	En este contexto, es una herramienta informática que facilita el trabajo sobre cierto lenguaje de programación; abstrae y empaqueta diferentes elementos en tareas más grandes.
HTML	Son las siglas de un lenguaje de programación para páginas web, el cual significa lenguaje de marcas de hipertexto.
Ionic	Es un <i>framework</i> de trabajo para el desarrollo de páginas web y aplicaciones móviles de forma integrada. Este <i>framework</i> corre sobre el Cordova y Angular.

PERT	Técnica de revisión y evaluación de proyectos. Consiste en una técnica que se utiliza para la optimización de sistemas de modelado de redes.
Sistema operativo	Es un conjunto de programas informáticos que permite la interacción con los recursos que conforman una computadora.
Software	Es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.
TIC	Tecnología de la información y comunicación.

RESUMEN

La investigación de operaciones, conocida también como teoría de la toma de decisiones o programación matemática, consiste en un conjunto de técnicas basadas en la matemática, estadística y algoritmia, para la resolución óptima en la toma de decisiones. Su función principal es analizar sistemas reales y encontrar un punto óptimo para su funcionamiento; toman en cuenta, también, factores externos como la escasez de recursos.

La solución propuesta consiste en un sistema asequible para el usuario promedio en cuanto a la situación, la economía y las dependencias de hardware para que la herramienta esté en cualquier momento analizando la toma de decisiones. El sistema posee las principales técnicas de la investigación de operaciones para la toma de decisiones, las cuales se utilizan a través de una interfaz amigable.

OBJETIVOS

General

Implementar una solución de software de utilidad para toda aquella persona que necesite utilizar las técnicas más comunes relacionadas con la investigación de operaciones.

Específicos

1. Analizar los principales requerimientos que deben constituir la solución a desarrollar, con base en un mercado objetivo y otras soluciones previamente planteadas.
2. Considerar la teoría fundamental de los sistemas de información para el diseño del sistema a desarrollar.
3. Implementar en una solución móvil cuatro técnicas fundamentales de la investigación de operaciones para implementarlas dentro del sistema a desarrollar.
4. Desarrollar una solución de software que cumpla con las expectativas del mercado objetivo planteado.

INTRODUCCIÓN

El desarrollo exponencial en la tecnología en los últimos años crea un cambio constante en las TIC que se usan habitualmente, que mejoran conforme crece la tecnología en general. Hace tres décadas, un sistema informático podía manejarse solamente a través de ejecutables dentro de cada ordenador; luego, internet vino a crear un nuevo paradigma de sistema informático; algo similar a lo que la tecnología móvil vino a realizar hace una década atrás. Es por ello, que el uso de un sistema de ordenador para una tarea cotidiana hoy en día resulta poco pragmático que genera una oportunidad potencial para el emprendimiento de un nuevo sistema.

La solución propuesta consiste en un sistema asequible para el usuario promedio en cuanto a la situación, la economía y las dependencias de hardware para que la herramienta esté en cualquier momento analizando la toma de decisiones. El sistema posee las principales técnicas de la investigación de operaciones para la toma de decisiones, las cuales se utilizan a través de una interfaz amigable.

1. TEORÍAS DE LA INFORMACIÓN

1.1. Teoría de la complejidad

El análisis de algoritmos es una rama de las ciencias de la computación que estudia la eficiencia de los algoritmos. Por lo general, se utilizan métodos matemáticos para la cuantificación de su eficiencia. La medición de la eficiencia de un algoritmo debe realizarse independientemente del computador donde se ejecute.

El costo de un programa de computadora se mide desde dos perspectivas fundamentales: la primera perspectiva se basa en el esfuerzo y el tiempo humano que representa desarrollar, mantener y utilizar un programa; la segunda perspectiva se basa en un análisis de complejidad, el cual busca evaluar la eficiencia de un programa. La eficiencia de un programa se mide a través de técnicas matemáticas que cuantifican la cantidad de recursos que se requerirán para la ejecución del programa a partir de ciertos factores a tomar en cuenta.

Los factores a tomar en cuenta en esta perspectiva son la cantidad de tiempo y el espacio en memoria dentro de la computadora que requiere el programa en su ejecución. La complejidad en el espacio es la cantidad de memoria que necesita un programa para ejecutarse hasta su terminación. Por otro lado, la complejidad en el tiempo es la cantidad de tiempo que requiere un programa para ejecutarse hasta su terminación. Por lo general, este tipo de complejidad se mide en unidades de ejecución a nivel de procesador, ya que las unidades de tiempo convencionales variarán dependiendo del ordenador.

1.1.1. Complejidad en el espacio

De una forma semejante se analiza el algoritmo lógico de un programa para medir la cantidad de registros en memoria que utiliza a lo largo de toda su ejecución. El espacio real utilizado puede variar debido a factores externos al algoritmo: la arquitectura de la computadora utilizada, el compilador o intérprete del programa, el motor de ejecución, entre otros.

1.1.2. Complejidad en el tiempo

La tasa de crecimiento es una descripción de cuánto tiempo requiere un programa específico para ejecutarse por completo. La tasa de crecimiento depende del tamaño de la información que reciba como entrada al programa a la hora de iniciarse. Según esta cantidad de información a procesar varía la cantidad de sentencias a ejecutarse dentro del programa y, por tanto, el tiempo requerido para que el programa se ejecute por completo.

Se utiliza función algebraica para definir la tasa de crecimiento en función al tamaño de la entrada de información a procesar. Esta entrada se denota por lo general con la letra n y puede representar el tamaño de un arreglo de variables, número de nodos en una lista enlazada, número de registros en un archivo, el grado de un polinomio, entre otros.

Para el análisis de complejidad en el tiempo suelen manejarse unidades de procesamiento básicas, en lugar de unidades de tiempo convencionales. Esto se debe a que cada computador ejecuta tareas a una tasa distinta. Las unidades de procesamiento básicas son sentencias que requieren una llamada a la unidad lógica de procesamiento dentro del computador para ejecutarse. Esto brinda una descripción de la eficiencia de un programa independiente al equipo.

1.1.3. Notación O-Grande

La notación O es una forma de expresar el tiempo de ejecución de un programa. No es una medición de tiempo exacta sino una función que denota un comportamiento en términos de una expresión algebraica. Esta notación expresa factores constantes como el número medio de instrucciones máquina que genera un compilador determinado y el número medio de instrucciones máquina por segundo que ejecuta una computadora específica.

Con la notación O se expresa una aproximación general de la relación entre el tamaño del problema a resolver por el programa y el tiempo de ejecución requerido para resolverlo, representado en unidades de ejecución. Al manejar una representación aproximada genera un margen de incerteza, el cual, para su finalidad, resulta despreciable. Sin embargo, la función O cumple la condición de acotar superiormente a la función general dentro del dominio útil de la función.

En palabras de Joyanes y Zahonero “Si $f(n)$ es una función definida sobre enteros no negativos, se dice « $T(n)$ es $O(f(n))$ » si $f(n)$ acota superiormente a $T(n)$. $T(n)$ es $O(f(n))$ si existe un entero n_0 y una constante $c > 0$ tal que para todos los enteros $n \geq n_0$ se tendrá que $T(n) \leq c f(n)$ ¹”.

1.2. Modelado de redes

El modelado de redes es un área de la investigación de operaciones enfocada en la optimización de sistemas de redes. Algunas aplicaciones son las redes de computadoras, redes de transporte, redes eléctricas, oleoductos de gas natural o petróleo, planeación de proyectos, etc.

¹ JOYANES AGUILAR, Luis; ZAHONER, Ignacio. *Algoritmos y estructuras de datos*. p. 36.

Una red es una composición de un conjunto de puntos y un conjunto de líneas que unen ciertos pares de puntos. Los puntos se llaman nodos o vértices y las líneas se llaman arcos. Un arco es un par ordenado de vértices. Si un arco permite el movimiento de un vértice a otro en una sola dirección se le llama arco dirigido y a la red se le llama grafo dirigido. Por otro lado, si permiten el movimiento de un vértice a otro y de regreso se le llama arco no dirigido y a la red grafo no dirigido.

Estos sistemas de redes son una aplicación de los grafos ponderados dirigidos con los cuales se modela el flujo de un bien que va de una fuente a un destino dado. Se interpreta el peso de una arista en el grafo dirigido como una capacidad que impone un límite superior de consumo. Este límite superior se define por una función, la cual sirve de análisis para el proceso de optimización.

Se define un ejemplo sobre la representación de sistemas de redes a través de grafos: “si V denota el conjunto de pueblos y A el conjunto de carreteras, podemos definir una relación R sobre V como aRb si podemos viajar de a a b usando solamente las carreteras de A ”². En otras palabras, se posee un conjunto de nodos y un conjunto de caminos, los cuales poseen un nodo inicial y un nodo final. Esta representación matemática es el punto de partida para el análisis de sistemas de redes.

1.2.1. CPM/PERT

CPM, *critical path method* o método de la ruta crítica, surge para la optimización de procesos industriales en Reino Unido, encontrando la secuencia de procesos que más tiempo requiere. Por otro lado PERT, *program evaluation and review technique* o técnica de evaluación de revisión de

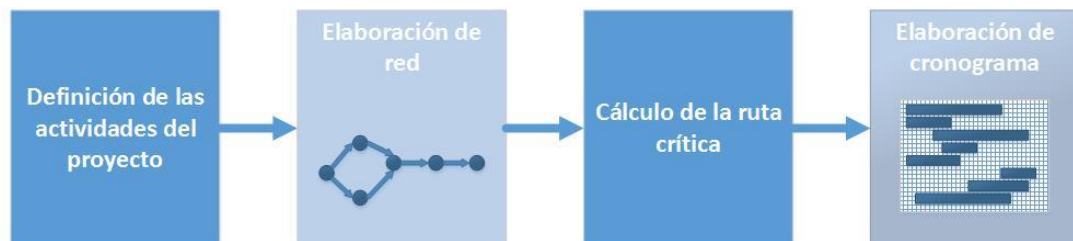
² GRIMALDI, Ralph P. *Matemática discreta y combinatoria*. p. 148.

programas, surge en un entorno militar, en el programa Polaris en Estados Unidos.

El objetivo de las técnicas PERT y CPM es el cálculo de la ruta crítica de un proceso. A pesar que los lugares y el entorno de desarrollo de las técnicas CPM y PERT son totalmente distintas, se desarrollaron con el mismo propósito. PERT supone que el tiempo para realizar cada una de las actividades es una variable aleatoria descrita por una distribución de probabilidad. CPM, por otra parte, infiere que el costo de las actividades se conoce en forma determinística.

PERT y CPM métodos aportaron los elementos necesarios para consolidar un solo método para la búsqueda de la ruta crítica basado en el diagrama de red. Actualmente, se utilizan como un solo método, difiriendo únicamente en la naturaleza de las variables a manejar.

Figura 1. **Búsqueda de la ruta crítica**



Fuente: elaboración propia.

Las fases que constituyen este proceso se ilustran en la figura 1. La primera fase consiste en la definición de actividades y por cada actividad una lista de actividades predecesoras y una ponderación de la actividad. Para concluir un proceso, deben llevarse a cabo todas las actividades definidas en

esta fase. Las actividades predecesoras son una lista de actividades que deben concluirse en su totalidad para iniciar la actividad respectiva.

De acuerdo con la figura 1, la siguiente fase es el trazo del grafo correspondiente a la red. Cada actividad se representa con las aristas del grafo. Cada actividad está representada por solo un arco y debe estar identificada por dos nodos terminales distintos. Los nodos del grafo constituyen las relaciones de precedencia entre las diferentes actividades.

La siguiente etapa consiste en el cálculo del costo total del proceso y la clasificación de las actividades en críticas o no críticas. Una actividad crítica es aquella cuya holgura es cero, lo cual significa que sus tiempos permanecen fijos. Por el contrario, una actividad no crítica es aquella que puede ser programada en un intervalo de tiempo mayor que su duración.

El cálculo de la ruta crítica implica la determinación del paso adelantado y del paso retrasado. El paso adelantado o tiempo de evento inicial indica el tiempo de ocurrencia más temprano de una actividad y se calcula recorriendo el grafo del nodo inicial al nodo final. El paso retrasado o tiempo de evento tardío indica el tiempo de ocurrencia más tardío y se calcula recorriendo el grafo del nodo final al nodo inicial. El nodo inicial es aquel en donde las actividades no poseen dependencias, mientras que el nodo final es aquel donde convergen todas las actividades de culminación.

El tiempo de evento inicial es el primer momento posible en el que puede ocurrir el evento correspondiente a un nodo de la red, es decir, una actividad. El tiempo de evento inicial de un nodo es el máximo de los tiempos de evento inicial de los nodos predecesores sumado a la ponderación de dicho nodo.

Debido a su dependencia es que se calcula desde los nodos iniciales, sin predecesores, hasta los nodos finales de la red.

El tiempo de evento final es el último momento posible cuando puede ocurrir el evento correspondiente a un nodo de la red sin retrasar la terminación del proyecto. El tiempo de evento final de un nodo es el mínimo de los tiempos de evento final de los nodos sucesores sustrayendo la ponderación dada a dicho nodo. El tiempo de evento final del último nodo de la red es igual a su tiempo de evento inicial. Se parte de este último nodo hasta el nodo de inicio de la red para realizar este cálculo.

Al finalizar este proceso pueden determinarse las actividades críticas. Una actividad crítica es aquella cuyo tiempo de evento inicial y tiempo de evento final son iguales. Las actividades críticas crean una secuencia de actividades que constituyen la ruta crítica. Cualquier retraso en una actividad crítica retardará la terminación del proyecto; de aquí surge la importancia de la determinación y el análisis de estas actividades para la optimización del proceso en general.

De no ser iguales los tiempos de evento de una actividad o nodo, la diferencia entre estos crea un tiempo de holgura o tiempo libre, lo cual indica que esa actividad puede adelantarse o retrasarse, según sea conveniente. Teniendo en cuenta la ruta crítica y las holguras de las actividades se puede proceder al desarrollo del cronograma, dependiendo del análisis del operador y la conveniencia a factores externos al del cálculo realizado.

1.3. Cadenas de Markov

Entre las diferentes clases de procesos estocásticos se destacan los procesos markovianos. Estos procesos se enfocan en el análisis del comportamiento y la evolución futura de un proceso a partir de su estado actual.

Al igual que para cualquier proceso, se distingue entre cadenas de Markov de tiempo discreto y aquellas de tiempo continuo. Las cadenas de Markov son procesos markovianos de espacio de estado discreto.

Las cadenas de Markov se describen como una secuencia de variables discretas que establecen un estado en el tiempo y cuya probabilidad de cambiar de estado tras un evento discreto, depende tan solo de los estados actuales, no de los estados pasados. Son procesos de Markov discretos tanto en el espacio de estados como en el dominio del tiempo. El hecho de que el dominio del tiempo sea discreto, permite definir un mecanismo de transición estocástico entre el presente del sistema y el futuro inmediato.

1.3.1. Cadenas de Markov homogéneas

Las cadenas de Markov homogéneas son cadenas de Markov que presentan un mecanismo de transición independiente del tiempo calendario. Cuando las probabilidades condicionales de pasaje de un estado al otro son también independientes de la etapa n decimos que la cadena correspondiente es homogénea en el tiempo; es decir, las variables aleatorias del proceso conservan su distribución en una traslación en el tiempo.

1.4. Teoría de colas

“La teoría de colas es el estudio de la espera en las distintas modalidades, representando los tipos de sistemas de líneas de espera a través de los modelos de colas. Un sistema de colas está básicamente conformado por una línea de espera, un conjunto de servidores y un flujo de clientes que entran y salen del sistema”³.

1.4.1. Proceso de entrada o llegada

Se tiene un proceso de entradas al sistema de elementos denominados clientes. Según Wayne L. Winston, en “los modelos típicos se supone que no más de una llegada ocurre en un instante dado y que una llegada al sistema no afecta el proceso de llegada”⁴.

Una variable de la cual depende el proceso de llegada es el tamaño o población de entrada. La población de entrada puede ser un número finito o adquirir un límite infinito. Esto se refleja en el proceso de entrada, dado que el caso en el que se tiene un proceso de entrada finito, el número de clientes en el sistema afecta al proceso de llegada, por lo que requiere un análisis más complejo de analizar.

Una variable aleatoria rige el momento en que un cliente llega al sistema. Según Wayne L. Winston “la suposición de que cada tiempo entre llegadas está regido por la misma variable aleatoria, implica que la distribución de llegadas es independiente del momento de tiempo en que se genere. Esto requiere que la

³ HILLIER, Frederick S.; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. p. 215.

⁴ WINSTON, Wayne L. *Investigación de operaciones*. p. 589.

tasa de llegada posea la propiedad de carencia de memoria, propiedad típica de la distribución exponencial”⁵.

1.4.2. Proceso de salida o de servicio

Hillier y Lieberman dicen que “el proceso de servicio consiste en una o más estaciones de servicio, cada una de ellas con uno o más canales de servicio paralelos, llamado servidores”⁶. Un cliente llega al sistema para ser atendido en uno de estos canales de servicio por un servidor.

Wayne L. Winston clasifican el proceso de servicio en “servidores en paralelo y servidores en serie. Los servidores en paralelo brindan el mismo servicio y los clientes solo requieren pasar por uno de ellos para salir del sistema”⁷. Los servidores en serie en cambio, forman una secuencia de servidores que el cliente debe transcurrir para salir del sistema.

El proceso de servicio se modela a partir de un tiempo de servicio. Hillier y Lieberman definen el tiempo de servicio como “el periodo de tiempo que el cliente pasa en un canal de servicio hasta terminar de ser atendido”⁸. Se modela a partir de una variable aleatoria con una distribución de probabilidad específica. Entre las distribuciones que se utilizan para modelar esta variable están la exponencial, distribución degenerada o de tiempos de servicio constantes, la distribución Erlang o gamma y la distribución de Poisson.

⁵ WINSTON, Wayne L. *Investigación de operaciones*. p. 589.

⁶ HILLIER, Frederick S.; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. p. 223.

⁷ Op. Cit. p. 599.

⁸ Op. Cit. p. 235.

1.4.3. Líneas de espera o colas

Una cola es donde los clientes esperan una vez dentro del sistema por recibir un servicio. Una cola puede tener un comportamiento finito o infinito, dependiendo si tiene o no un límite de clientes dentro de ella. Las colas infinitas se utilizan en la mayoría de modelos, pero existen ciertos casos, en donde el límite de fila es lo suficientemente pequeño como para alcanzarse de forma frecuente, por lo que es necesario modelarlo.

Una cola posee también una disciplina, que es la forma en la que se elige al cliente que será atendido. Existe la disciplina del primer cliente en llegar es el primer cliente atendido, el cual es el caso más usado. Otras disciplinas son el último cliente en llegar es el primer cliente atendido, existen colas con prioridad o colas de servicio aleatorio.

1.4.4. Proceso de nacimiento y muerte

El proceso de nacimiento y muerte es una teoría con diversas aplicaciones en la estadística. En este caso, describe en términos probabilísticos como cambian los intervalos de tiempo entre cada nacimiento y entre cada muerte a través del tiempo. Esto vuelve a los modelos de colas un caso específico de cadena de Markov de tiempo continuo. Este enfoque permite la derivación de varios modelos con tasa de llegada con una distribución de Poisson y tiempos de servicio exponenciales.

1.4.4.1. Modelo M/M/s

Hillier y Lieberman describen este modelo como: “una simplificación del proceso de nacimiento y muerte”⁹. Los tiempos entre llegadas y los tiempos de servicio se asumen como independientes e idénticamente distribuidos de acuerdo con una distribución exponencial. El caso más simple de este modelo es el caso en donde existe solamente un servidor ($S=1$).

1.4.4.2. Modelo M/M/s/K

Hillier y Lieberman describen este modelo como: “una variante al modelo M/M/s, en el cual se tiene una cola finita”¹⁰. Esto implica que el número de clientes en el sistema no puede exceder un límite K definido.

1.4.5. Otros modelos

Hillier y Lieberman (2010, p.737) dan una noción simplificada de modelos que no siguen una distribución exponencial. Estos modelos, requieren un análisis matemático superior al del proceso de nacimiento y muerte.

1.4.5.1. Modelo M/G/1

Para Hillier y Lieberman este modelo asume que: “el sistema de colas tienen un servidor, un proceso de entradas de Poisson, con un tiempo entre llegadas exponenciales, por lo que se maneja una tasa media de llegadas fija”¹¹. Se supone que los clientes tienen tiempos de servicio independientes,

⁹ HILLIER, Frederick S.; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. p. 241.

¹⁰ *Ibíd.*

¹¹ *Ibíd.*

pero no se imponen restricciones sobre cuál debe ser la distribución de servicio. Para la estructuración del modelo se requiere conocer la media y la varianza de la distribución de servicio.

1.4.5.2. Modelo M/D/s

Para Hillier y Lieberman este modelo asume que: “el sistema de colas tienen un servidor un proceso de entradas de Poisson, con un tiempo entre llegadas exponenciales, por lo que se maneja una tasa media de llegadas fija”¹². Se supone que los clientes tienen tiempos de servicio independientes, pero con una distribución degenerada o de tiempos de servicio constantes para los servidores. Esta distribución asume que no existe variación entre un tiempo de servicio y otro.

1.4.5.3. Modelo M/Ek/s

Para Hillier y Lieberman este modelo es: “un caso intermedio entre la distribución exponencial y la distribución M/D/s. Una asume una variación máxima entre los tiempos de servicio, mientras la otra asume una variación de cero respectivamente”¹³. La distribución de Erlang describe una función de densidad de probabilidad a partir de una varianza definida entre los tiempos de llegada.

1.5. Teoría de inventarios

Wayne L. Winston describe que “la teoría de inventarios se utiliza para minimizar los costos involucrados en una empresa por el mantenimiento de

¹² HILLIER, Frederick S.; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. p. 243.

¹³ *Ibíd.*

mercancía que se espera vender. Los modelos de inventarios definen el periodo de tiempo en el que se debe hacer un pedido de producto y la cantidad de producto a pedir”¹⁴.

1.5.1. Tipos de costo

La optimización de costos en los modelos de inventario se hace con base en diferentes costos que surgen en el ciclo de vida de un inventario. Los costos tomados en cuenta en los principales procesos se muestran a continuación.

1.5.1.1. Costo de pedido y organización

Consiste en el costo por realizar un pedido, independientemente de la cantidad de producto pedido. Este costo toma en cuenta gastos administrativos o de trabajadores dentro de la empresa en el momento de realizar un pedido.

1.5.1.2. Costo de compra unitario

Consiste en el costo que implica solicitar una unidad de producto. Según Wayne L. Winston, este costo “incluye el costo de mano de obra variable, el costo fijo variable y el costo de materia prima al producirse el producto”¹⁵. Al adquirirse por un servicio de terceros el costo debe incluir, además del precio de venta, gastos involucrados en el transporte del producto, como pago de impuestos y costo de envío.

¹⁴ WINSTON, Wayne L. *Investigación de operaciones*. p. 599.

¹⁵ *Ibíd.*

1.5.1.3. Costo de retención o posesión

Wayne L. Winston describe este costo como: “el costo de mantener una unidad de inventario durante un periodo de tiempo”¹⁶. Según Hillier y Lieberman este costo suele llamarse: “costo de almacenar e incluye el costo del capital invertido, espacio, seguros, protección e impuestos relacionados con el almacenamiento del producto en inventario”¹⁷.

Según Hillier y Lieberman este costo “debe tomar en cuenta un valor de rescate. Este valor representa el valor de desecho de un artículo sobrante, cuando no se requiere más inventario. Al capital perdido por estos sobrantes se le llama costo de recuperación”¹⁸.

1.5.1.4. Costo de escasez o por faltantes

Según Wayne L. Winston, existen dos escenarios donde se desenvuelve este costo en el caso en el que no se pueda satisfacer la demanda de un cliente por escasez de productos: “el primer escenario es que los pedidos se puedan posponer, en este caso los pedidos pasan a una lista de pendientes y se completan en el siguiente periodo de abastecimiento de productos; el segundo escenario es cuando se pierden las ventas que no son completadas, la pérdida de una venta implica una pérdida de capital, además, puede ocasionar otros gastos a la empresa por el atraso de producción.”¹⁹

¹⁶ WINSTON, Wayne L. *Investigación de operaciones*. p. 599.

¹⁷ HILLIER, Frederick S.; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. p. 243.

¹⁸ *Ibíd.*

¹⁹ WINSTON, Wayne L. *Investigación de operaciones*. p. 611.

1.5.2. Modelo de lote económico

Según Hillier y Lieberman, el modelo de lote económico modela la forma en que los niveles de inventarios se reducen a lo largo del tiempo y se reabastece en periodos constantes de tiempo²⁰.

El modelo de lote económico toma en cuenta un costo de preparación para ordenar un lote, un costo unitario por producto y un costo unitario de almacenamiento del producto en inventario. El modelo busca calcular con qué frecuencia y en qué cantidad se debe reabastecer el inventario de manera que minimice el costo total del modelo.

En este modelo básico no se permiten faltantes, por lo que se asume que el reabastecimiento llega inmediatamente después de que el inventario llega a cero. Además, se maneja una tasa constante de demanda de los clientes por el producto.

²⁰ HILLIER, Frederick S.; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. p. 243.

2. IDENTIFICACIÓN DEL PROBLEMA

2.1. Antecedentes

Los sistemas de software se construyen con el objetivo de facilitar las labores del ser humano e incrementar su eficiencia y eficacia. Conforme se desarrollan nuevas tecnologías de software, incrementa también su alcance para el desarrollo de tareas cotidianas que normalmente realizaría el ser humano.

En el campo de la investigación de operaciones por los estudiantes y profesionales de la Universidad de San Carlos de Guatemala, no se utilizan herramientas de software para la implementación de las técnicas de este campo. La única excepción son sistemas sin soporte, actualmente, incluso incompatibles con los sistemas operativos actuales.

Los profesionales de la Universidad de San Carlos de Guatemala que toman decisiones en áreas donde son aplicables técnicas de la investigación de operaciones prefieren utilizar un análisis superficial al problema. Esto se debe a que estas técnicas requieren la inversión de un tiempo que muchas veces no disponen para implementarlo.

Se tiene, entonces, la necesidad de una herramienta que sea capaz de apoyar al profesional y al estudiante a resolver problemas complejos en un tiempo óptimo. Se requiere que sea también asequible tanto económicamente como en cuanto a su facilidad de uso.

2.2. Mercado objetivo

En primer plano para la observación del problema a trabajar, se determinaron a los estudiantes de ingeniería de la Universidad de San Carlos de Guatemala como usuarios en potencia. Los estudiantes en los cursos relacionados con el tema de investigación de operaciones carecen de una herramienta de software que simplifique el uso de las técnicas de una forma práctica. En un espacio pedagógico, una herramienta de este tipo resulta necesaria, al adentrarse en problemas complejos, que requieran un mayor nivel de análisis, en vez de un dominio básico de cada técnica. Como posibles usuarios futuros, se encuentran los profesionales que utilizan estas técnicas y buscan una solución asequible y fácil de utilizar para facilitar sus labores cotidianas.

En forma general, la solución a realizar está enfocada en usuarios que busquen una herramienta de software libre. Los principales usuarios son estudiantes de una entidad pública, por lo que el nivel económico de los usuarios puede ser muy diverso y se desea abarcar la mayor cantidad de usuarios posibles.

2.3. *Benchmarking*

Con base en el mercado objetivo de la solución a plantear, se ha encontrado que la principal competencia es el software WinQSB. WinQSB es un sistema que permite la solución de una gran cantidad de problemas administrativos, de producción, de recursos humanos, de dirección de proyectos, entre otros. Se enfoca a grandes rasgos en la solución de problemas de investigación de operaciones.

WinQSB comprende diversos módulos funcionales: para la solución de problemas de programación lineal, programación cuadrática, programación no lineal, modelado de redes, métodos PERT/CPM, análisis de colas, sistema de simulación de colas, teoría y sistemas de inventarios, procesos de Markov, análisis de decisiones, predicciones, control de calidad, análisis de aceptación de muestras, programación de flujos de trabajo, planeamiento agregado, facilidades de localización y planeación de requerimiento de materiales.

Su principal limitación consiste en la falta de seguimiento de esta herramienta. Este programa fue desarrollado para correr sobre una plataforma Windows. Sin embargo, esta no se actualizó y corre hasta Windows XP. Además, esta corre solo en arquitecturas de 32 bits; lo cual lo hace en una herramienta obsoleta y con una interfaz poco amigable para el usuario; aunque su funcionalidad consiste en un gran punto de partida para la implementación de un sistema actualizado.

3. DISEÑO PRELIMINAR

A continuación, se presenta un diseño preliminar de la solución a implementar, con base en los requerimientos planteados para el sistema. Se muestra la forma como el usuario puede hacer uso de la aplicación, la funcionalidad general de los módulos a desarrollar y las características que se desea brindar a la solución.

3.1. Estructura general del sistema

Como se puede observar en la figura 2, el sistema en general consta de dos áreas principales: la primera es un panel con los diferentes módulos del sistema, los módulos a comprender son cadenas de Markov, modelado de redes, teoría de colas y teoría de inventarios, con solo seleccionar una de las opciones se podrá cambiar de área de trabajo sin perder los datos entre un módulo y otro; la segunda área es el área de trabajo, en donde se despliegan las diferentes etapas de trabajo de cada módulo y se realiza la resolución de problemas de forma automatizada.

Cada módulo está compuesto por una interfaz inicial para el ingreso de datos iniciales, en donde se definirán parámetros que restrinjan el problema. El ingreso de estos datos permitirá mostrar, en algunos casos, una segunda interfaz de ingreso personalizada según los parámetros iniciales ingresados. En esta interfaz se procede a ingresar los valores necesarios para la resolución del problema seleccionado. Finalmente, al terminar el ingreso de datos, se mostrará al usuario una interfaz con el reporte de todos los valores calculados.

Las etapas de formularios y de reportes pueden contener parámetros que requieran más espacio que el que se tiene en pantalla. También, puede resultar útil al usuario trabajar con el teléfono de forma vertical u horizontal con respecto a la pantalla. Para ello, la interfaz debe adaptarse a las necesidades del usuario o del problema. Esta interfaz se redimensiona en tamaño, según el tamaño de la pantalla, la posición del positivo móvil y la cantidad de parámetros de entrada o de salida. Al ser más grande la interfaz que la pantalla física, basta con hacer un deslizamiento sobre la pantalla para desplazarse por la interfaz lógica.

Figura 2. **Estructura general del sistema**



Fuente: elaboración propia.

3.2. Estructura del módulo de cadenas markovianas

El primer módulo a definir es el módulo de cadenas markovianas. Esta técnica consiste en el cálculo probabilístico de un cambio de estado luego de cierto tiempo transcurrido. Para ello se requiere conocer una matriz de transición con cada una de las probabilidades de transitar de un estado a otro, la condición presente de la que se desea partir y el tiempo a futuro que se desea proyectar.

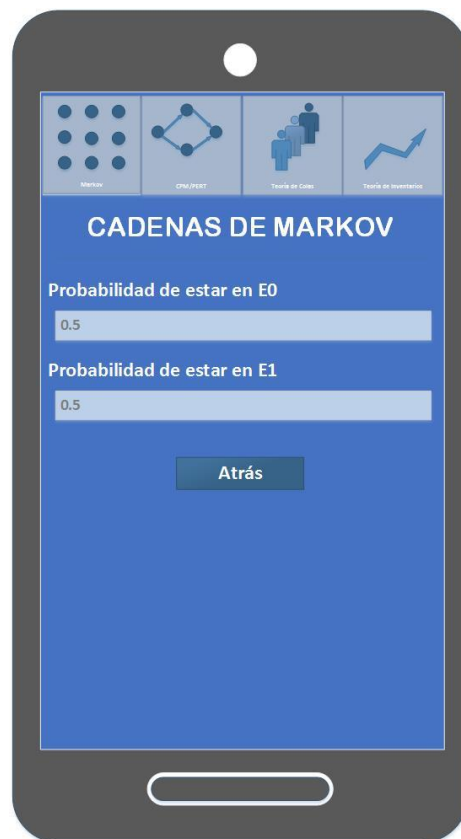
Figura 3. Formulario del módulo de cadenas markovianas

The image shows a mobile application interface for calculating Markov chains. At the top, there is a navigation bar with four icons: a grid of dots (labeled 'Inicio'), a network diagram (labeled 'CMU/ITST'), a group of people (labeled 'Trabajo en Clase'), and a line graph (labeled 'Trabajo de Investigación'). Below the navigation bar, the main title is 'CADENAS DE MARKOV'. The form consists of several sections: 'Número de estados' with a dropdown menu showing '2'; 'Matriz de transición' with four dropdown menus labeled 'De E0 a E0', 'De E0 a E1', 'De E1 a E0', and 'De E1 a E1'; 'Vector inicial' with two dropdown menus labeled 'E0' and 'E1'; and 'Tiempo' with a dropdown menu. At the bottom of the form is a 'Calcular' button.

Fuente: elaboración propia.

En la figura 3 se observa la estructura de la etapa inicial del módulo de cadenas de Markov. Esta interfaz solicita al usuario los datos requeridos para la resolución del problema. Primero, se requiere conocer el grado del problema; es decir, la cantidad de estados estocásticos que se desea manejar. La variación de este parámetro hace cambiar la interfaz en sí, ya que dependiendo del grado del problema varía el grado de la matriz de transición y el grado del vector inicial. Este cambio se realiza de forma automática y redimensiona la pantalla de ser necesario.

Figura 4. **Reporte del módulo de cadenas markovianas**



Fuente: elaboración propia.

La operación de ingreso de datos de este módulo termina en cuanto se presiona el botón 'calcular'. Luego, se procede a mostrar un reporte similar al de la figura 4. La cantidad de respuestas dependerá del grado del problema a trabajar, ya que hay una probabilidad resultante por cada estado existente. En el caso que se tenga un grado lo suficientemente grande como para abarcar más espacio que el que se tiene en el dispositivo. Al terminar de visualizar el reporte pueden editarse los datos ingresados presionando 'atrás', cambiar de módulo o cerrar el programa.

3.3. Estructura del módulo de redes

En el módulo de redes se tiene primero una etapa de delimitación del problema. En esta se restringen los parámetros generales para adaptar de forma dinámica una etapa subsiguiente de ingreso de datos. Esta etapa cuenta con dos parámetros de entrada a definir como se observa en la figura 5.

Uno de los parámetros de entrada es la cantidad de actividades dentro del problema, la cual es un número entero positivo. Dependiendo de este número se genera una estructura de datos para representar el modelo de red. En esta se generan identificadores únicos por cada actividad utilizando una notación a partir de letras para que sea más fácil para el usuario diferenciar entre el identificador que acaba de conocer y los valores que debe ingresar.

La segunda variable es el método o técnica a utilizar. El algoritmo a trabajar es exactamente el mismo, la diferencia radica en la personalización del ingreso de datos dependiendo de la naturaleza de la variable. La división de la etapa de ingreso de datos se debe a la complejidad temporal cuadrática que requiere la generación de la estructura de datos y de los identificadores, por lo que no es aceptable realizar estos procesos de forma dinámica y continua.

Figura 5. **Delimitación del problema del módulo de redes**



Fuente: elaboración propia.

Al presionar el botón 'calcular' de la interfaz de la figura 5 se conciben dos distintos escenarios. El primero es la elección del método de la ruta crítica. En este caso la etapa de ingreso de datos posee una estructura como la de la figura 6. El costo de cada actividad es una variable determinística, por lo que se puede solicitar al usuario un valor exacto, el cual puede ser un número real mayor que cero. Este costo puede corresponder a un tiempo de transición, un costo, una distancia, una ponderación de esfuerzo o una cantidad de cierto recurso que se requiere para culminar la actividad.

Figura 6. **Formulario para el modelo de redes utilizando CPM**

Fuente: elaboración propia.

Además, cada actividad posee una lista de predecesores. Esta es una lista de actividades que deben terminarse para iniciar dicha actividad. Para referenciar a otra actividad, se escribe la letra o conjunto de letras que identifican a la actividad que se desea referenciar. Al haber más de una actividad predecesora se separan con comas. El identificador de cada variable se genera automáticamente por el sistema y corresponde a una notación común en el modelado de redes. Se utilizan identificadores autogenerados para evitar el ingreso de datos innecesarios al usuario.

El segundo escenario a tomar en cuenta es donde se elige la técnica de revisión y evaluación de proyectos para la resolución del problema. Este se observa en la figura 7. En este caso el costo de la actividad se maneja de forma probabilística. Esto depende de la forma como se mida el costo de cada actividad y de la naturaleza del costo que están midiendo. En este caso se requiere ingresar valores mínimo, máximo y medio del costo de cada actividad. De estos valores se obtiene un costo promedio y se procede a operar de la misma forma que en el método de la ruta crítica.

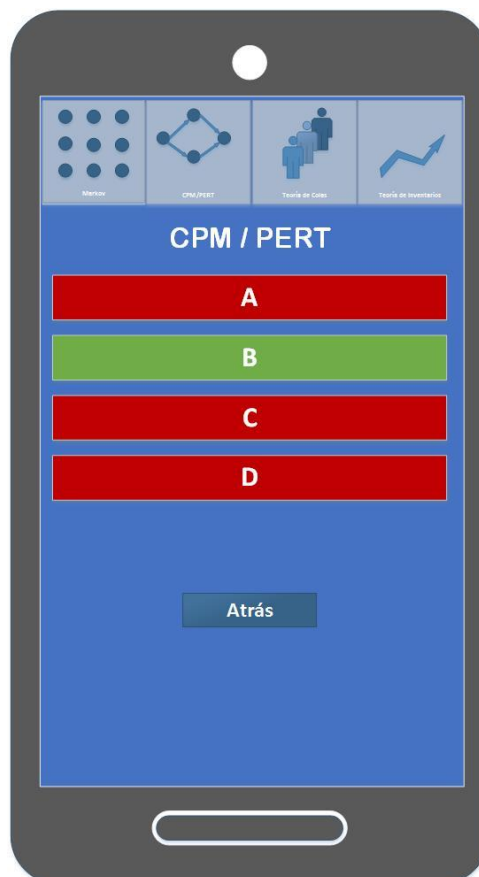
Figura 7. **Formulario para el modelo de redes utilizando PERT**

The image shows a smartphone screen with a blue-themed application interface. At the top, there is a navigation bar with four icons: a grid of dots (Menu), a network diagram (CNU/PERT), a group of people (Tabla de Costo), and a line graph (Tabla de Inventario). Below the navigation bar, the title "PERT" is centered. The main content area is divided into two sections, A and B. Each section contains four input fields: "Costo Mínimo", "Costo Medio", "Costo Máximo", and "Predecesores". Each of the first three fields has a small up/down arrow icon on its right side. At the bottom of the screen, there is a dark blue button labeled "Calcular".

Fuente: elaboración propia.

Una vez ingresados los parámetros de entrada, en cualquiera de los escenarios planteados, se procede a presionar el botón 'calcular'. Esto despliega la etapa del reporte del modelo de redes, como se observa en la figura 8. En esta etapa se observan en forma de lista las actividades definidas en las etapas anteriores. Esta posee una codificación de colores, en donde las actividades con un fondo rojo son parte de la ruta crítica, mientras las actividades con un fondo verde no.

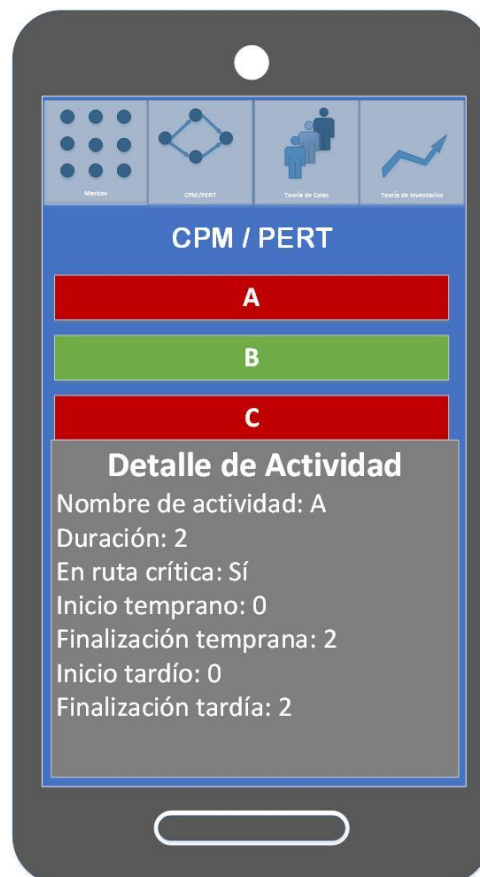
Figura 8. **Reporte del módulo de redes**



Fuente: elaboración propia.

La interfaz del reporte cuenta además, con un desglose detallado de información al presionar una actividad, como se muestra en la figura 9. Este informe incluye el nombre de la actividad, el costo de la actividad, si pertenece o no a la ruta crítica y el intervalo de tiempo cuando puede oscilar el inicio y el fin de cada actividad. Este reporte se muestra en una ventana emergente y desaparece al presionar fuera de la pantalla o transcurrido un periodo corto de tiempo.

Figura 9. **Reporte detallado del módulo de redes**



Fuente: elaboración propia.

3.4. Estructura del módulo de teoría de inventarios

El módulo de teoría de colas comprende una serie de casos específicos de aplicación. Cada caso posee un algoritmo de solución y cierta cantidad de entradas específicas. La etapa de ingreso de datos se muestra en la figura 10 donde se muestra una serie de opciones, con los casos de aplicación de la teoría de inventarios básicos. Dependiendo de la opción seleccionada cambian las variables definidas en la parte inferior de la interfaz.

Figura 10. **Formulario del módulo de teoría de inventarios**

The image shows a mobile application interface for inventory theory. At the top, there is a navigation bar with four icons: a grid of dots (Inicio), a network diagram (OPINIONES), a queue of people (Teoría de Colas), and a line graph (Teoría de Inventarios). Below the navigation bar, the main title is 'TEORÍA DE INVENTARIOS'. Underneath, there is a section 'Método a trabajar' with a radio button selected for 'Modelo de lote económico (EOQ)'. Other options include 'Modelo de lote económico con faltantes', 'Modelo de lote económico con producción y consumo simultaneo', and 'Modelo de lote económico con producción y consumo simultáneo con faltantes'. The 'Variables' section contains five input fields: 'Costo de almacenamiento' (h), 'Costo de emisión' (k), 'Costo de pedido' (c), 'Tasa de demanda' (d), and a 'Calcular' button at the bottom.

Fuente: elaboración propia.

Una vez definido el método e ingresados sus respectivos parámetros de entrada se presiona 'calcular'. El sistema realiza todos los cálculos posibles para el método y despliega un reporte con una estructura semejante a la de la figura 11. Se calculan los valores óptimos de cantidad de productos y tiempo de pedido de forma determinística a partir de los cuales se obtiene el costo mínimo de producción.

Figura 11. **Reporte del módulo de teoría de inventarios**



Fuente: elaboración propia.

3.5. Estructura del módulo de teoría de colas

El módulo de teoría de colas comprende una serie de casos específicos de aplicación. Cada caso posee un algoritmo de solución y cierta cantidad de entradas específicas. La etapa de ingreso de datos se muestra en la figura 12 donde se muestra una serie de opciones, con los casos de aplicación de la teoría de inventarios básicos. Dependiendo de la opción seleccionada cambian las variables definidas en la parte inferior de la interfaz.

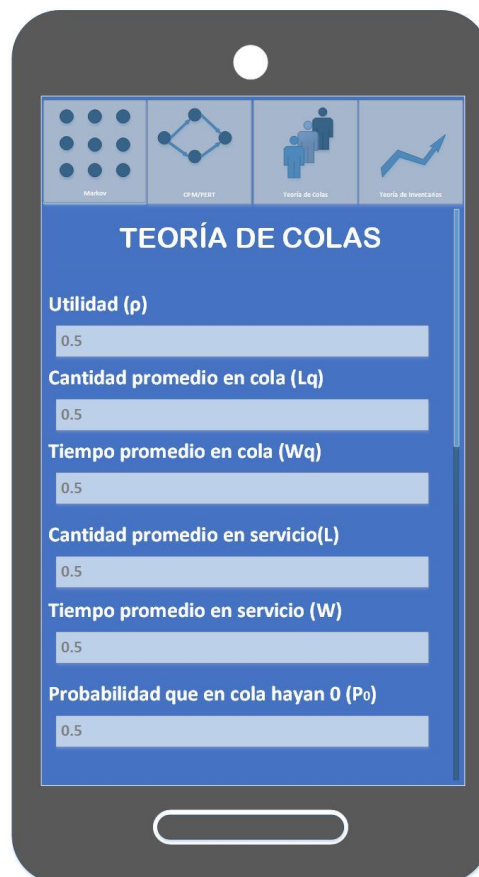
Figura 12. **Formulario del módulo de teoría de colas**

The image shows a mobile application interface for queue theory. At the top, there is a navigation bar with four icons: a grid of dots (Inicio), a network diagram (QMPERT), a queue of people (Teoría de Colas), and a line graph (Teoría de Inventarios). Below the navigation bar, the title 'TEORÍA DE COLAS' is displayed in a blue header. Underneath, the section 'Método a trabajar' contains four radio button options for different queue models: 'M / M / 1 / oo / oo / FIFO' (which is selected), 'M / M / C / oo / oo / FIFO', 'M / M / C / K / oo / FIFO', and 'M / M / oo / oo / oo / FIFO'. The 'Variables' section follows, with three input fields: 'λ Tasa de entrada', 'μ Tasa de servicio', and 'i Número de corridas de prueba'. Each input field has a dropdown arrow on the right. At the bottom of the form is a 'Calcular' button.

Fuente: elaboración propia.

Una vez definido el método e ingresados sus respectivos parámetros de entrada se presiona 'calcular'. El sistema realiza todos los cálculos posibles para el método y despliega un reporte con una estructura semejante a la de la figura 13. Entre los cálculos a realizar en cada uno de los casos están la utilidad del sistema actual, la cantidad promedio de personas en la cola y en el servicio, el tiempo promedio de una persona en la cola y en el servicio y una serie de probabilidades sobre la cantidad de personas en cola.

Figura 13. **Reporte del módulo de teoría de colas**



The image shows a smartphone screen with a dark grey background. At the top, there is a navigation bar with four icons: a 3x3 grid of dots, a network diagram, a queue of people, and a line graph. Below the navigation bar, the title "TEORÍA DE COLAS" is displayed in white text on a blue background. The main content area is a list of input fields, each with a label and a value of 0.5. The labels are: "Utilidad (ρ)", "Cantidad promedio en cola (L_q)", "Tiempo promedio en cola (W_q)", "Cantidad promedio en servicio (L)", "Tiempo promedio en servicio (W)", and "Probabilidad que en cola hayan 0 (P_0)".

Fuente: elaboración propia.

4. DOCUMENTACIÓN

4.1. Herramientas

La solución propuesta posee una arquitectura monolítica. Esta utiliza herramientas nativas de *Android* tras su despliegue desde el *framework* donde se desarrolló. Sin embargo, esta aplicación se basta a sí misma para ejecutar todas las técnicas y métodos que la comprenden.

4.2. Hardware

El sistema está diseñado con la finalidad de ejecutarse en dispositivos móviles. No posee ninguna dependencia de hardware con los dispositivos, ya que no hace uso de ningún módulo físico dentro de este. Sin embargo, por cuestión de costos de despliegue del sistema, tendrá restricciones de sistema operativo, lo que traerá limitaciones con los dispositivos móviles bajo los que se ejecutará.

4.3. Software

El sistema operativo bajo el que se ejecutará será *Android OS*, únicamente para versiones 4 o posteriores, debido a las dependencias con el *framework* Ionic y a los costos de publicación del producto. Esto se hace con el objetivo de minimizar costos y aprovechar la constante actualización de los dispositivos móviles por parte de los usuarios.

4.4. Manual de desarrollo

Se describen todos los requerimientos para el desarrollo de la aplicación. También, se mencionan los requerimientos para la implementación de la solución planteada como las herramientas utilizadas, las dependencias de software de terceros estas herramientas y sus requerimientos mínimos de hardware.

4.4.1. Herramientas

Para el desarrollo de la solución planteada se utilizó el *framework* Ionic, el cual provee las librerías y herramientas de desarrollo necesarias para compilar, desplegar y realizar pruebas a la aplicación. Como plataforma para el desarrollo de la solución se eligió Ionic. Este *framework* maneja el despliegue de soluciones híbridas, es decir, puede desplegar soluciones para múltiples plataformas. Esto facilita incrementar el alcance de la solución a un mayor mercado objetivo.

Ionic es una herramienta de código libre y de código abierto. Ionic ofrece una biblioteca de HTML optimizado para móviles, componentes CSS y Javascript, gestores y herramientas para crear aplicaciones altamente interactivas. Está basado en Angular.js y en Cordova. Angular.js brinda una lógica simplificada y optimizada basada en Node.js. Las herramientas que se utilizaron se describen a continuación:

- La última versión del kit de desarrollo de *Android*
- Node.js versión 6
- La última versión disponible de Cordova
- La última versión disponible de Ionic
- Kit de despliegue de Ionic a *Android*

4.4.2. Configuración del entorno de desarrollo

En la figura 14 se observa una secuencia de pasos que se siguieron para la instalación de todas las herramientas descritas en la sección anterior. La primera herramienta a instalar es el kit de desarrollo de *Android*. Esta herramienta es requerida para realizar el despliegue de la aplicación en la plataforma móvil.

Ionic es un *framework* que requiere de Cordova y Node.js para su correcto funcionamiento. Es por ello que se requiere su instalación, como se observa en la figura 14. Finalmente se instala Ionic, se crea un proyecto y se configura su plataforma de despliegue. Los últimos tres comandos se utilizan para el despliegue y pruebas de la solución.

Figura 14. Comandos de configuración y despliegue

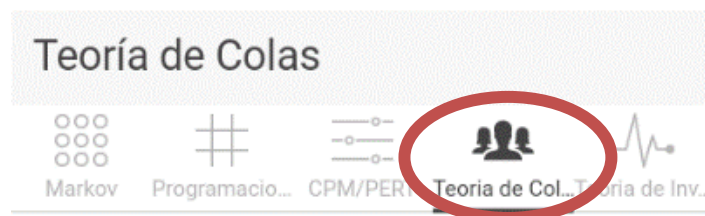
```
1 #1. Instalar SDK
2 $ sudo add-apt-repository ppa:ubuntu-sdk-team/ppa
3 $ sudo apt update && sudo apt install ubuntu-sdk
4 $ sudo apt update && sudo apt dist-upgrade
5
6 #2. Instalar Node.js
7 $ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
8 $ sudo apt-get install -y nodejs
9
10 #3. Instalar Cordova
11 $ sudo npm install -g cordova
12
13 #4. Instalar Ionic
14 $ sudo npm install -g ionic
15
16 #5. Crear Proyecto
17 $ ionic start Proyecto1 tabs
18 $ cd Proyecto1
19
20 #6. Configurar plataforma de despliegue
21 $ ionic platform add android
22
23 #7. Despliegue y pruebas
24 $ ionic serve
25 $ ionic build android
26 $ ionic emulate android
```

Fuente: elaboración propia.

4.5. Secuencia de la solución

El sistema comprende un conjunto de herramientas al cual se puede acceder desde un menú principal semejante al de la figura 15. Este menú se encuentra visible en todo momento desde la aplicación. Los módulos que contiene la aplicación son cadenas de Markov, modelado de redes o CPM/PERT, teoría de colas y teoría de inventarios. Puede navegar entre un módulo y otro sin perder los datos ya ingresados para facilitar la navegación entre pantallas.

Figura 15. Menú principal

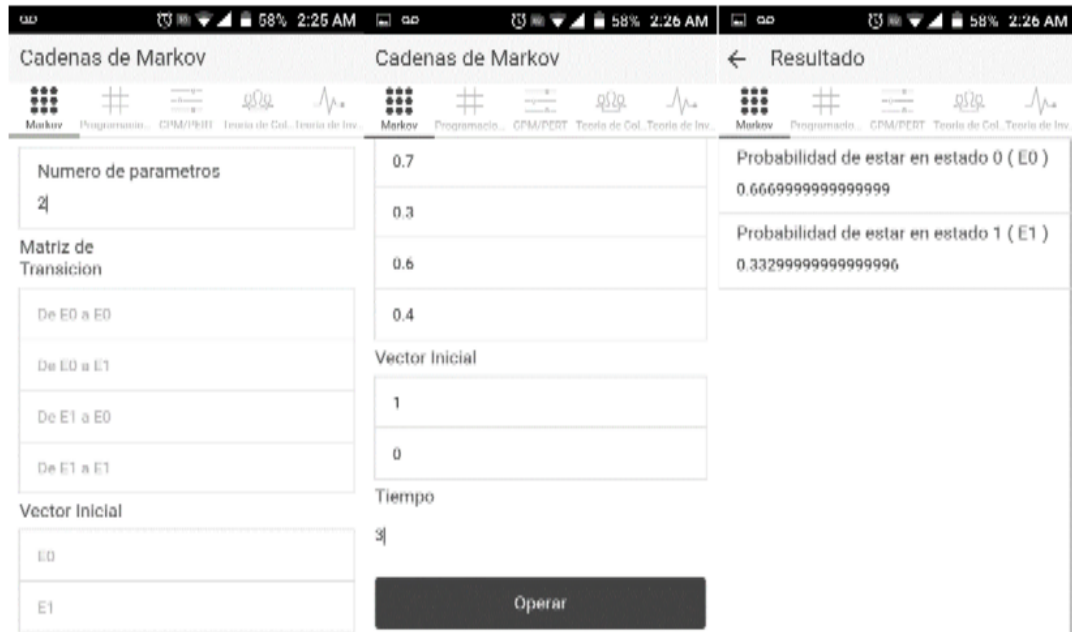


Fuente: elaboración propia.

En la figura 16 se muestra la secuencia básica de uno de los módulos; en este caso el módulo de cadenas markovianas. En este caso se unifica la etapa de delimitación del problema y de ingreso de datos. Cuando se ingresa el grado de estados estocásticos a manejar cambia dinámicamente el número de entradas de datos al sistema.

En la sección central de la figura 16 se observa un ejemplo de grado dos, el cual comprende de las probabilidades de la matriz de transición, las probabilidades del vector inicial y el número de transiciones a futuro a generar. En la sección derecha de la figura 16 se observa la proyección a futuro generada por el sistema.

Figura 16. Funcionamiento del módulo de Markov

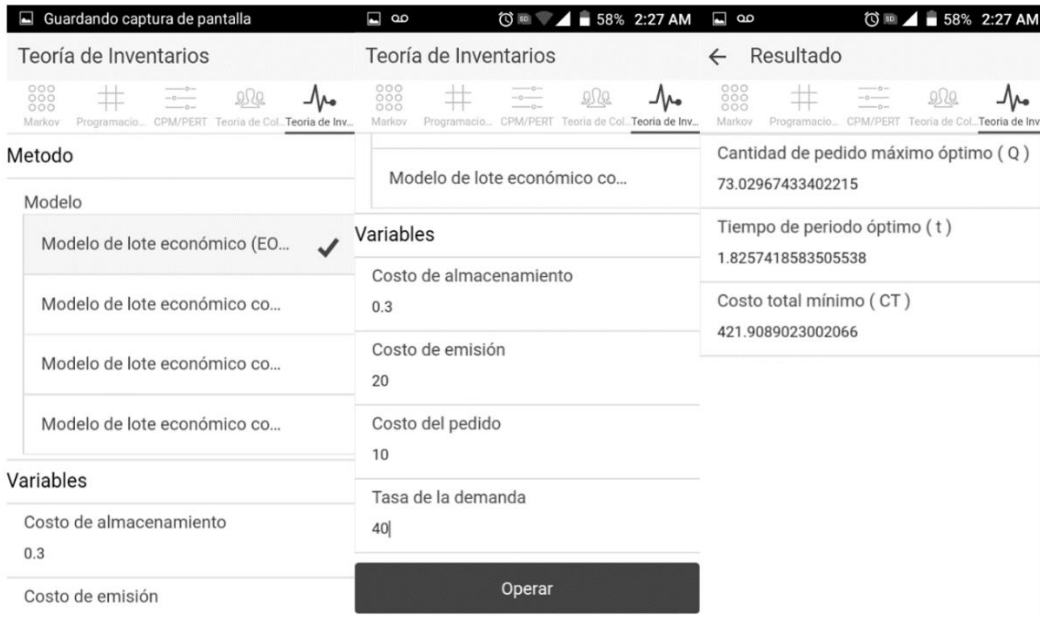


Fuente: elaboración propia.

En la figura 17 se muestra la secuencia básica del módulo de teoría de inventarios. En este caso se unifica la etapa de delimitación del problema y de ingreso de datos. Primero, se tiene una lista de modelos disponibles con los cuales trabajar como se muestra en la parte izquierda de la figura. En la parte central se muestra el formato de ingreso de datos, en donde se definen todos los costos a considerar dentro del modelo.

Una vez ingresados todos los parámetros de entrada se procede a presionar la opción 'calcular'. Esto despliega el reporte de la parte derecha de la figura 17. Este reporte muestra el tamaño del lote y el periodo de tiempo entre cada pedido óptimo y su costo.

Figura 17. Funcionamiento del módulo de teoría de inventarios



Fuente: elaboración propia.

CONCLUSIONES

1. El manejo del concepto de usabilidad dentro de una interfaz se torna más difícil mientras más elementos se manejan dentro de la misma.
2. La optimización de procesos genera un impacto considerable en dispositivos móviles, debido al límite de procesamiento de algunos dispositivos.
3. La eficiencia de una operación de software depende tanto del algoritmo planteado, como de las herramientas de software utilizadas.
4. El principal requerimiento de un usuario para adquirir una solución móvil es la asequibilidad y su apariencia.

RECOMENDACIONES

1. Al diseñar una interfaz en la que se deba presentar una gran cantidad de información, lo mejor es crear una estructuración ordenada e intuitiva, para que la usabilidad perdure a pesar de la complejidad del problema que se maneje.
2. Tomar el estudio de mercado objetivo y el *benchmarking* como un proceso para la fase de recopilación de requerimientos y diseño de la solución.
3. En el momento de implementar un algoritmo, técnica o metodología existente, se debe evaluar constantemente en qué medida la implementación es óptima, a través de la función de tiempo, el manejo de diferentes estructura de datos y el manejo de diferentes estructuras lógicas.
4. Debe conocerse a fondo cada aspecto de una técnica para el diseño e implementación de un sistema informático que lo ejecute.
5. Detallar la interfaz final, conociendo a fondo las necesidades del cliente y las herramientas que se utilizan para su implementación a la hora de realizar el diseño preliminar.

BIBLIOGRAFÍA

1. GRIMALDI, Ralph P. *Matemática discreta y combinatoria*. 3a ed. México: Addison-Wesley Iberoamericana, 1997. 148 p.
2. HILLIER, Frederick S; LIEBERMAN, Gerald J. *Introducción a la investigación de operaciones*. 9a ed. México: McGraw-Hill/Interamericana S.A., 2010. 978 p.
3. JOYANES AGUILAR, Luis; ZAHONER, Ignacio. *Algoritmos y estructuras de datos*. México: McGraw-Hill/Interamericana S.A., 2004. 217 p.
4. WINSTON, Wayne L. *Investigación de operaciones*. 4a ed. México: Cengage Learning Editores S.A., 2004. 1418 p.

APÉNDICE

Apéndice 1. Cadenas de Markov



**O.R.
ASSISTANT**



CADENAS DE MARKOV



CPM / PERT



TEORÍA DE COLAS



TEORÍA DE INVENTARIOS

Autor
German Oliva
german.gaom@gmail.com

CADENAS DE MARKOV

1. NÚMERO DE PARÁMETROS
Ingresar el número de estados estocásticos dentro del problema.

2. MATRIZ DE TRANSICIÓN
Llenar cada una de las entradas con la probabilidad de la descripción utilizando un número decimal menor que uno como se muestra en el ejemplo. Los estados se nombran con una letra E y un número de identificación.

3. VECTOR INICIAL
Ingresar por cada estado estocástico la probabilidad de estar en este en el tiempo presente, según el formato dado.

4. TIEMPO
Ingresar un número entero que represente el salto a futuro que se desea proyectar.

5. CALCULAR
Presionar el botón calcular para proceder a visualizar los resultados.

6. ATRÁS
Al estar en el reporte siempre puedes regresar y editar los datos ingresados para ver variantes de los datos actuales.



OTROS MÓDULOS

<p>1. CPM / PERT Ingresar la cantidad de nodos o actividades en la red y presionar siguiente.</p>	<p>1. DEMÁS MÓDULOS Elegir el método a trabajar.</p>
--	---

2. INGRESO VALORES
Ingresar cada uno de los valores que se les pide como parámetros de entrada. Cada entrada posee un nombre de variable y una descripción de la misma para su correcto uso.

5. CALCULAR
Presionar el botón calcular para proceder a visualizar los resultados.

6. ATRÁS
Al estar en el reporte siempre puedes regresar y editar los datos ingresados para ver variantes de los datos actuales.

 <p>EJEMPLO NÚMERO</p>	 <p>EJEMPLO PROBABILIDAD</p>
---	---



Continuación del apéndice 1.



CADENAS DE MARKOV



OTROS MÉTODOS



Fuente: elaboración propia.