

**DISEÑO DE UN SOFTWARE PARA REALIZAR ESTUDIOS  
ESTADÍSTICOS EN TIEMPOS DISCRETOS, UTILIZANDO CADENAS DE  
MARKOV**

**EDGAR FERNANDO VÉLEZ ZAMUDIO**

**UNIVERSIDAD PEDAGÓGICA NACIONAL**

**Facultad de ciencia y tecnología**

**Departamento de matemáticas**

**2015**

**DISEÑO DE UN SOFTWARE PARA REALIZAR ESTUDIOS  
ESTADÍSTICOS EN TIEMPOS DISCRETOS, UTILIZANDO CADENAS DE  
MARKOV**

*Edgar Fernando Vélez Zamudio*

*Cod: 2004240056*

*CC 80'449.526*

**TRABAJO DE GRADO**

*Modalidad: Interés personal*

**ASESOR**

*Jorge Edgar Páez Ortega*

**UNIVERSIDAD PEDAGÓGICA NACIONAL**

**Facultad de ciencia y tecnología**

**Departamento de matemáticas**

**Bogotá 2015**

**Nota de aceptación**

---

---

---

---

**Firma Jurado**

---

**Firma Jurado**

---

**Firma Asesor**

Bogotá, marzo de 2015

## RESUMEN ANALÍTICO EN EDUCACIÓN - RAE

1. Información General	
<b>Tipo de documento</b>	Trabajo de grado.
<b>Acceso al documento</b>	Universidad Pedagógica Nacional. Biblioteca Central
<b>Título del documento</b>	Diseño de un software para realizar estudios estadísticos en tiempos discretos, utilizando cadenas de Márkov.
<b>Autor(es)</b>	Edgar Fernando Vélez Zamudio.
<b>Director</b>	Jorge Edgar Páez Ortigón
<b>Publicación</b>	Bogotá, Universidad Pedagógica Nacional, 2015. 69 paginas.
<b>Unidad Patrocinante</b>	Universidad Pedagógica Nacional
<b>Palabras Claves</b>	Procesos estocásticos, cadenas de Márkov, software.

2. Descripción
<p>Este trabajo pretende mostrar el diseño de un software en power builder 9.0, el cual realice los cálculos de una cadena de Márkov en tiempos discretos, y obtenga como resultado la matriz de transición.</p> <p>Primero el usuario ingresa la cantidad de estados y sus nombres, luego ingresa en orden los estados por completo y el software calcula la matriz de transición, en seguida realiza la predicción a futuro de encontrarse en cada estado.</p> <p>En el documento escrito se encuentra: El código que se utilizó para programar en power builder 9.0, el manual de instrucciones de la herramienta, la descripción del proceso, los elementos que se tuvieron en cuenta para el diseño y elaboración del software y los resultados obtenidos en la prueba y aplicación del software.</p>

3. Fuentes
<p>Para la elaboración del software se consultaron tres fuentes principalmente:</p> <p>La primera relacionada con el código de programación y la sintaxis que utiliza power builder 9.0 para poder lograr que el software realizara los procesos y cálculos que se requerían.</p> <p>Gomez. W. (2000) powerbuilder, manual del programador. SYBASE, en línea <a href="http://es.scribd.com/doc/136839733/58470932-Manual-Power-Builder-pdf">http://es.scribd.com/doc/136839733/58470932-Manual-Power-Builder-pdf</a></p>

García, O., Nieto, F., (2004) Software educativo: Cadenas de Márkov en tiempo discreto. Tesis de grado para recibir el título de Ingeniero industrial en la Pontificia Universidad Javeriana.

La segunda fuente corresponde al trabajo realizado por las cadenas de Márkov en tiempos discretos, principalmente en el tema procesos estocásticos.

Guyon. X. (1999) Métodos numéricos por cadenas de Márkov. Université Paris 1Y Universidad de Los Andes, Mérida, Venezuela [http://samos.univ-paris1.fr/archives/ftp/cours/guyon/Meth\\_Num-Ch.Markov-Espanol.pdf](http://samos.univ-paris1.fr/archives/ftp/cours/guyon/Meth_Num-Ch.Markov-Espanol.pdf)

Mesa. G. (2007) Cadenas de Márkov, una sencilla aplicación. Revista memorias. Vol.5 No 9 en línea <http://www.revistamemorias.com/articulos9/cadenasdemarkov.pdf>

Aznar. E. (2007) Andrei Andreyevich Márkov. Universidad de Granada, facultad de ciencias, departamento de álgebra. En línea <http://www.ugr.es/~eaznar/markov.htm>

La tercera fuente sobre la deserción en la Universidad Pedagógica Nacional, para poder realizar la aplicación de la herramienta tecnológica en un ejemplo real.

Hernández, M. & Hernández, Carmenza. (1999). Factores determinantes de la deserción estudiantil en la Universidad Pedagógica Nacional. Comparación entre desertores y no desertores. Universidad Pedagógica Nacional, Vicerrectoría Académica, Sub proyecto 3.1.1.3 Inscripción, Admisión y Seguimiento a estudiantes, Bogotá.

UPN (2006) La deserción estudiantil en la UPN. Universidad Pedagógica Nacional, Vicerrectoría Académica, Centro de orientación y acompañamiento a estudiantes (COAE), En línea: [http://www.mineducacion.gov.co/1621/articles-85600\\_Archivo\\_pdf2.pdf](http://www.mineducacion.gov.co/1621/articles-85600_Archivo_pdf2.pdf)

#### 4. Contenidos

Este documento se encuentra estructurado en cinco apartados, en la primera parte se presenta la formulación del proyecto, los antecedentes que se tuvieron en cuenta, la justificación y los objetivos que se plantearon para el diseño del software.

En la segunda parte se presenta el marco referencial, centrado en la teoría principal de las cadenas de Márkov, una definición de la terminología empleada y las leyes de probabilidad que se utilizan. También la información sobre el lenguaje de programación

powerbuilder 9.0 (con el cual se diseñó el software), una descripción general de su funcionamiento y las utilidades como herramienta tecnológica.

En la tercera parte se encuentra el proceso de elaboración del software CMTD10, incluyendo el manual de instalación y uso de la herramienta, el código que se empleó para programar dicho software y una descripción general del proceso de elaboración del software.

En la cuarta parte se muestra el software diseñado, una evaluación en un evento real de la Universidad Pedagógica Nacional, verificando así la aplicación que puede tener la herramienta y ofreciendo una predicción a futuro con cierto porcentaje de probabilidad.

Por último, en la quinta parte se presentan algunas conclusiones personales y del trabajo de grado, teniendo en cuenta los objetivos planteados para el diseño y elaboración del software, el proceso y el resultado final.

### 5. Metodología

La metodología utilizada para el diseño del software se basa en las siguientes etapas: primero, identificación de la necesidad del software para simplificar cálculos y obtener los resultados necesarios ágil y eficazmente. Segundo planeación y diseño del software, tercero aplicación en una situación real del software y cuarto elaboración del documento final, el cual reúne el manual final del usuario y el código empleado.

### 6. Conclusiones

Al utilizar una herramienta tecnológica para simplificar cálculos, se obtiene un gran provecho del uso de las tecnologías actuales, además de lograr resultados confiables e inmediatos.

El código empleado en la programación se puede simplificar haciendo un debido y acertado uso de los ciclos y arreglos para organizar la información, además de utilizar un ambiente grafico que sea amigable con el usuario y facilite el uso del software.

Diseñar un software para una función específica pone en juego los conocimientos adquiridos a lo largo de la carrera, no solo en las áreas afines a la computación, si no a las demás asignaturas, teniendo en cuenta todos los casos posibles e inconvenientes que se puedan presentar debido a indeterminaciones y demás.

<b>Elaborado por:</b>	Edgar Fernando Vélez Zamudio
<b>Revisado por:</b>	Jorge Edgar Páez Ortegón

<b>Fecha de elaboración del Resumen:</b>	10	01	2015
--	----	----	------

# CONTENIDO

	Pag.
INTRODUCCIÓN .....	8
1. FORMULACIÓN DEL PROYECTO .....	10
1.1. JUSTIFICACIÓN. ....	10
1.2. OBJETIVOS .....	11
1.2.1. OBJETIVO GENERAL .....	11
1.2.2. OBJETIVOS ESPECÍFICOS. ....	11
2. MARCO REFERENCIAL.....	13
2.1. CADENAS DE MÁRKOV .....	13
2.1.1. ANDREIANDREYEVICH MÁRKOV .....	13
2.1.2. TERMINOLOGÍA .....	16
2.1.3. CADENAS DE MÁRKOV .....	17
2.1.4. APLICACIONES DE LAS CADENAS DE MÁRKOV .....	20
2.2. POWERBUILDER 9.0 .....	22
3. DISEÑO.....	24
3.1. ANTECEDENTES.....	24
3.2. PROCESO DE ELABORACIÓN.....	25
3.3. MANUAL DE INSTALACIÓN.....	40
3.3.1. INSTRUCCIONES DE USO .....	43
3.4. CÓDIGO POWERBUILDER 9.0.....	50
4. EVALUACIÓN DE LA HERRAMIENTA .....	56
4.1. ANÁLISIS.....	60
5. CONCLUSIONES.....	67
BIBLIOGRAFÍA.....	68

## INTRODUCCIÓN

El estado del tiempo de un día cualquiera, estar en mora o no con una institución bancaria y la página de internet que más se visita, son algunas de las situaciones en las cuales podemos utilizar un modelo estadístico para determinar la probabilidad que ocurra algún evento. Las cadenas de Márkov se utilizan para predecir la evolución y el comportamiento de determinados sistemas que cambian a través del tiempo aleatoriamente, a esto se le denomina un proceso estocástico.

Las cadenas de Márkov nos ofrecen la probabilidad de pasar de un estado a otro o mantenerse en el mismo estado, por ejemplo en el estado del tiempo, se podría determinar la probabilidad que al siguiente día llueva o haga sol, todo depende de los estados del tiempo anteriores y principalmente del estado del tiempo actual; Entre más datos se tengan mayor va a ser la confiabilidad de la probabilidad, como en la mayoría de estudios estadísticos es solo un indicador de probabilidad, mas no una predicción exacta y precisa de lo que puede suceder en el futuro; como se trata de tiempos discretos se maneja una cantidad finita de estados, los cuales pueden ser cualitativos o cuantitativos.

A continuación se presenta el diseño de un software, el cual realiza los cálculos propios de una cadena de Márkov, desde que ingresa los estados en orden, hasta que muestra los resultados organizados en una matriz de transición, la cual permite al usuario tener acceso a todos los casos posibles y la probabilidad de encontrarse en un estado.

Este documento se encuentra estructurado en cinco apartados, en la primera parte se presenta la formulación del proyecto, los antecedentes que se tuvieron en cuenta, la justificación y los objetivos que se plantearon para el diseño del software.

En la segunda parte se presenta el marco referencial, centrado en la teoría principal de las cadenas de Márkov, una definición de la terminología empleada y las leyes de probabilidad que se utilizan. También la información sobre el lenguaje de programación powerbuilder 9.0 (con el cual se diseñó el software), una descripción general de su funcionamiento y las utilidades como herramienta tecnológica.

En la tercera parte se encuentra el proceso de elaboración del software CMTD10, incluyendo el manual de instalación y uso de la herramienta, el código que se empleó para programar dicho software y una descripción general del proceso de elaboración del software.

En la cuarta parte se muestra el software diseñado, una evaluación en un evento real de la Universidad Pedagógica Nacional, verificando así la aplicación que puede tener la herramienta y ofreciendo una predicción a futuro con cierto porcentaje de probabilidad.

Por último, en la quinta parte se presentan algunas conclusiones personales y del trabajo de grado, teniendo en cuenta los objetivos planteados para el diseño y elaboración del software, el proceso y el resultado final.

# **1. FORMULACIÓN DEL PROYECTO**

El proyecto se formuló bajo la sugerencia del profesor Benjamín Sarmiento, docente de la Universidad Pedagógica Nacional en el área de estadística y probabilidad, luego se realizó una verificación de antecedentes en cuanto a trabajos realizados y se procedió a formular el proyecto CMTD10.

## **1.1. JUSTIFICACIÓN.**

En la actualidad es usual realizar estudios con estadística básica, sin tener en cuenta la probabilidad de que un evento suceda o no, las cadenas de Márkov realizan un análisis más completo, teniendo en cuenta dichas probabilidades y ofreciendo una predicción con la probabilidad de encontrarse en cada estado, es por esto que se programó un software que permita realizar un análisis utilizando un proceso estocástico.

Se diseñó un software que realice los cálculos de una cadena de Márkov para que los estudiantes de la Licenciatura en matemáticas o cualquier otra licenciatura de la Universidad Pedagógica Nacional, tengan la posibilidad de acceder a él y sacar un mejor provecho de la herramienta tecnológica, la cual agiliza y simplifica procedimientos, utilizando un entorno grafico de fácil manipulación y navegación.

Los procesos estocásticos se utilizan para modelar matemáticamente una gran variedad de situaciones problema en diferentes ramas del conocimiento, es necesario contar con un aplicativo tecnológico que agilice los cálculos y permita una mejor comprensión del tema.

## **1.2. OBJETIVOS**

Para el diseño del software CMTD10, se plantearon los siguientes objetivos, los cuales pretenden orientar el trabajo de grado y servir de herramienta de evaluación al final del proceso, analizando el cumplimiento parcial, total o nulo de cada uno de ellos.

### **1.2.1.OBJETIVO GENERAL**

Diseñar un software que realice los cálculos de una cadena de Márkov y organice los resultados en una matriz de transición, ofreciendo una predicción en un periodo de tiempo específico, utilizando un ambiente gráfico fácil de utilizar y mostrando su utilidad mediante un ejemplo real, describiendo el proceso de recopilación y elaboración.

### **1.2.2.OBJETIVOS ESPECÍFICOS.**

- ✓ Diseñar un software que realice los cálculos propios de una cadena de Márkov, y ofrezca al usuario una matriz de transición con un porcentaje de probabilidad específico.
  
- ✓ Realizar una aplicación del software en una situación específica de la Universidad Pedagógica Nacional, para evidenciar su utilidad y obtener una predicción de la probabilidad que suceda o no a futuro.

- ✓ Producir un documento que contenga la descripción del proceso de las cadenas de Márkov, el código de programación y el manual de instalación del software.

## **2. MARCO REFERENCIAL**

Se consultó la información necesaria para poder tener las herramientas suficientes para el diseño del software CMTD10, en cuanto a la teoría de las cadenas de Márkov y sus leyes, además de la sintaxis propia del lenguaje de programación PowerBuilder 9.0.

### **2.1. CADENAS DE MÁRKOV**

Se inicia este estudio desde el componente histórico, seguido de los preconceptos necesarios para conocer ésta temática, la fundamentación desde los procesos estocásticos y las aplicaciones.

#### **2.1.1. ANDREI ANDREYEVICH MÁRKOV**

Nació el 14 de junio de 1856 en Ryazan, Rusia y murió el 20 de julio de 1922 en Petrogrado, ahora San Petersburgo. En su infancia tuvo poca salud, teniendo que andar con muletas hasta los 10 años. Su educación secundaria fue en el Gymnasium San Petersburgo, donde mostró su talento matemático. En esta época escribe un primer trabajo sobre integración de ecuaciones diferenciales ordinarias, aunque no presenta nada nuevo.

En 1874, entra en la facultad de Físicas y Matemáticas de San Petersburgo. Asiste a las clases de Korkin, Zolotarev y sobre todo de Chebyshev, que era el jefe del departamento de matemáticas.

Después de graduarse, comienza a enseñar en la Universidad de San Petersburgo como ayudante. Consigue su doctorado en 1884, con una disertación Sobre ciertas aplicaciones de las fracciones continuas.

Desde 1886 fue profesor contratado de la Universidad de San Petersburgo y en 1893 se convierte en profesor ordinario. En 1886, Chebyshev propone a Markov para la Academia Rusa de ciencias, siendo aceptado como miembro ordinario desde 1896. Formalmente se retiró en 1905, aunque continuó enseñando hasta el final de su vida.

Los primeros trabajos de Márkov fueron sobre teoría de números y análisis, fracciones continuas, límites de integrales, teoría de aproximación y convergencia de series. Después de 1900, Márkov aplica los métodos de fracciones continuas, que había comenzado su maestro Pafnuty Chebyshev, a la Teoría de Probabilidades.

Según Aznar. E (2007), Márkov fue el más elegante portavoz y continuador de las ideas de Chebyshev. Destaca su aportación al teorema de Jacob Bernoulli conocido como la Ley de los Grandes Números, a dos teoremas fundamentales de probabilidad debidos a Chebyshev, y al método de los mínimos cuadrados.

Estudió sucesiones de variables mutuamente dependientes, con la esperanza de establecer las leyes límite de probabilidad en su forma más general. Probó el teorema Central del Límite bajo ciertas condiciones generales.

Sin embargo Markov es recordado por su estudio de las llamadas *cadena de Markov*, sucesiones de variables aleatorias en las cuales la siguiente variable está determinada por la actual variable pero es independiente de las anteriores, este trabajo lo publicó en 1907 y con esto surge una nueva rama de la teoría de probabilidades y comienza la teoría de los *procesos estocásticos*.

En 1923, Norbert Wiener introduce rigurosamente los procesos de Markov continuos. Sin embargo, la teoría general fue estudiada, en la década de 1930, por Andrei Kolmogorov.

El curso de Markov sobre cálculo de probabilidades y sus memorias originales, modelo de precisión y claridad, contribuyen a la transformación de la teoría de probabilidades, contribuyendo al desarrollo de las ideas Chebyshev con su análisis de las dependencias entre variables aleatorias.

Markov estuvo también interesado en la poesía y realizó estudios de estilos poéticos, afición que compartió con Kolmogorov. Aunque Markov desarrolló su teoría de cadenas, desde un punto de vista totalmente teórico, también aplicó estas ideas a cadenas de dos estados, vocales y consonantes, en los textos literarios.

Tras el comienzo de la revolución Rusa en 1917 y con la subida de los alimentos básicos, en septiembre de este año, Markov pidió a la Academia que le enviara a una ciudad de la Rusia interior y fue trasladado a Zarskoye Selo, una pequeña ciudad, donde enseñó matemáticas en la escuela secundaria sin recibir ningún pago.

Cuando volvió a San Petersburgo en 1921, su salud estaba deteriorada y tenía un ojo enfermo y aunque se encontraba en mal estado de salud, seguía dando clases de probabilidad en la Universidad. Su muerte se produjo en julio de 1922, después de varios meses de agonía.

Markov tuvo un hijo con su mismo nombre, nació el 9 de septiembre de 1903 y años después siguió sus huellas y se convirtió en un renombrado matemático, hizo grandes contribuciones a las matemáticas constructivas y la teoría de funciones recursivas. Conocido particularmente por el principio de Markov y la regla de

Markov. Fue quien dio continuidad al trabajo realizado por su padre y se encargó de hacerlo público para que fuera reconocido y divulgado.

## **2.1.2.TERMINOLOGÍA**

Para poder comprender mejor la terminología utilizada en el proceso de las cadenas de Márkov, a continuación se presenta una descripción de cada término utilizado y su significado en el contexto de las cadenas de Márkov en tiempos discretos.

### **✓ PROCESO ESTOCÁSTICO.**

Este término se utiliza en estadística y concretamente en la probabilidad, para determinar un proceso o fenómeno aleatorio que cambia o evoluciona con el tiempo.

### **✓ ESTADO**

Es la caracterización de la condición en la que se encuentra un sistema en cierto instante de tiempo, los valores del estado pueden ser cualitativos o cuantitativos según la situación, la cantidad de estados en tiempos discretos es finita.

### **✓ TRANSICIÓN**

En una cadena de Márkov, cuando el sistema pasa de un estado a otro o se mantiene en el mismo, después de cierto tiempo, se llama transición de estado.

### **✓ PROBABILIDAD CONDICIONAL**

Como es un proceso estocástico, no se puede saber con certeza el estado del sistema en un instante de tiempo futuro, pero si la probabilidad que se asocia a cada uno de los estados, dependiendo estrictamente del estado actual, pero teniendo en cuenta la transición de los estados anteriores.

### 2.1.3.CADENAS DE MÁRKOV

Guardiet (2002) describe los *estados* ( $E_{(t)}$ ) como una caracterización de la situación en la que se encuentra el sistema en un instante de tiempo ( $t$ ) dado, puede ser cualitativo o cuantitativo. Se determinan como una variable, que cambia de valor en el tiempo, a este cambio se le denomina *transición*. Por ser un sistema estocástico, no se conocerá con certeza el estado del sistema en un determinado instante, pero si la *probabilidad* asociada a cada estado y se define así:

$$p\{E_t = j | E_{t-1} = i, E_{t-2} = e_{t-2}, E_{t-3} = e_{t-3}, \dots, E_0 = e_0\} = p\{E_t = j | E_{t-1} = i\} = p_{ij}$$

Donde  $i, j$  y  $e$  son estados posibles del sistema y  $p_{ij}$  es la probabilidad de transición del estado  $i$  al estado  $j$ , además, el estado del sistema en el futuro  $j$  solo depende del estado presente  $i$

Un ejemplo que se podría plantear para poder comprender mejor cómo funciona el procedimiento de una cadena de Márkov es el siguiente:

*Se tiene el estado del tiempo de 10 días seguidos en una ciudad específica, se sabe que 5 días fueron soleados y 5 días fueron nublados.*

Con solo ésta información se podría afirmar que la probabilidad que al siguiente día sea soleado o nublado es del 50% puesto que de los diez días, la mitad fueron soleados y la otra mitad nublados.

Pero si tenemos en cuenta el orden en que se dieron estos días y realizamos un análisis con las cadenas de Márkov, se podría predecir con un porcentaje la probabilidad que el siguiente día sea soleado o nublado, todo dependiendo principalmente del estado del tiempo actual, es decir el del décimo día.

El orden del estado del tiempo en esos días fue el siguiente:

**Tabla N°1.**  $S=Soleado$   $N= Nublado$

Día	1	2	3	4	5	6	7	8	9	10
Estado del tiempo	S	S	S	S	N	N	N	S	N	N

Teniendo en cuenta la información de la tabla N°1, para este ejemplo contamos con 2 estados (soleado y nublado) y un periodo de tiempo de 10 días, siguiendo con la definición tendríamos lo siguiente:

$$p\{E_{11} = S | E_{10} = N, E_9 = N, E_8 = S, \dots, E_1 = S\} = p\{E_{11} = S | E_{10} = N\} = p_{NS}$$

La ley de *probabilidad* condicional es de la forma:

$$p\{E_{t+s} = j | E_{t+s-1} = 1\} = p\{E_t = j | E_{t-1} = 1\}$$

Donde  $s$  son los periodos de tiempo anteriores que se van a tener en cuenta.

La información se organiza en una matriz de probabilidad de transición, la cual es cuadrada y tiene tantas filas como estados del sistema, además la suma de las probabilidades de cada fila debe ser igual a uno y de estas matrices parte el trabajo de las cadenas de Márkov.

Para el ejemplo que estábamos trabajando la matriz de transición es la siguiente:

$$\begin{array}{cc} & \text{S} & \text{N} \\ \begin{array}{c} \text{S} \\ \text{N} \end{array} & \begin{pmatrix} 0,6 & 0,4 \\ 0,25 & 0,75 \end{pmatrix} \end{array}$$

Nos indica que como en el décimo día fue nublado, hay una probabilidad del 60% que al siguiente día sea nublado y de un 40% que sea soleado, esto se puede observar en la primera fila que corresponde a un día nublado, estado del tiempo del día diez.

Más adelante mostrare con ayuda del software CMTD10, como se hace para obtener estos resultados.

## **2.1.4.APLICACIONES DE LAS CADENAS DE MÁRKOV**

Las cadenas de Márkov se pueden utilizar para realizar estudios estadísticos en diferentes ramas del conocimiento, a continuación menciono algunas ramas específicas y un ejemplo de aplicación.

### **✓ FÍSICA**

El uso de las cadenas de Márkov en el área de física se puede observar principalmente en la termodinámica, donde el ejemplo más claro son las cadenas de Ehrenfest o el modelo de difusión de Laplace.

### **✓ ECONOMÍA Y FINANZAS**

Las cadenas de Márkov en la economía y finanzas son utilizadas para determinar la oportunidad de arbitraje, además se puede predecir a corto o a largo plazo el colapso de una bolsa de valores como medida preventiva, son utilizadas también para determinar la volatilidad de precios en el mercado.

### **✓ NEGOCIOS**

Se utilizan las cadenas de Márkov para determinar la capacidad de pago de un deudor moroso, también lo hacen en entidades de riesgo como data crédito determinando la probabilidad de pago después de ciertas transiciones en el sistema, el porcentaje de confianza aumenta a medida que el tiempo pasa y el usuario se mantiene al día en sus pagos.

### **✓ JUEGOS DE AZAR**

La mayoría de juegos de azar se pueden modelar mediante una cadena de Márkov, la teoría de la ruina del jugador es ratificada, cada vez la probabilidad de que el

jugador pierda es mayor y con el paso del tiempo la ruina es inevitable, hasta que el jugador queda sin dinero.

#### ✓ INTERNET

EL rango de páginas utilizado por los motores de búsqueda, está organizado por una cadena de Márkov, el cual tiene en cuenta la página en que se encuentra el usuario y la pagina a la cual cambia, esto le otorga una probabilidad cada vez mayor a las páginas más visitadas y las ubica en una mejor posición del motor de búsqueda.

#### ✓ EPIDEMIOLOGIA

El proceso Galton-Watson es un claro ejemplo de la aplicación de las cadenas de Márkov en esta rama, este proceso consiste en ramificaciones que se puede usar, entre otras cosas, para modelar el desarrollo de una epidemia, modelando matemáticamente la probabilidad de contagio.

#### ✓ METEOROLOGÍA

Este es el uso más frecuente y sencillo de una cadena de Márkov, si se realiza un estudio, día a día en una zona específica, se tendrá que el clima del día siguiente va a depender exclusivamente del clima actual, el porcentaje de probabilidad de cada estado se afianzara con el paso del tiempo y esto permitirá determinar las épocas de lluvia o sol con una mayor precisión.

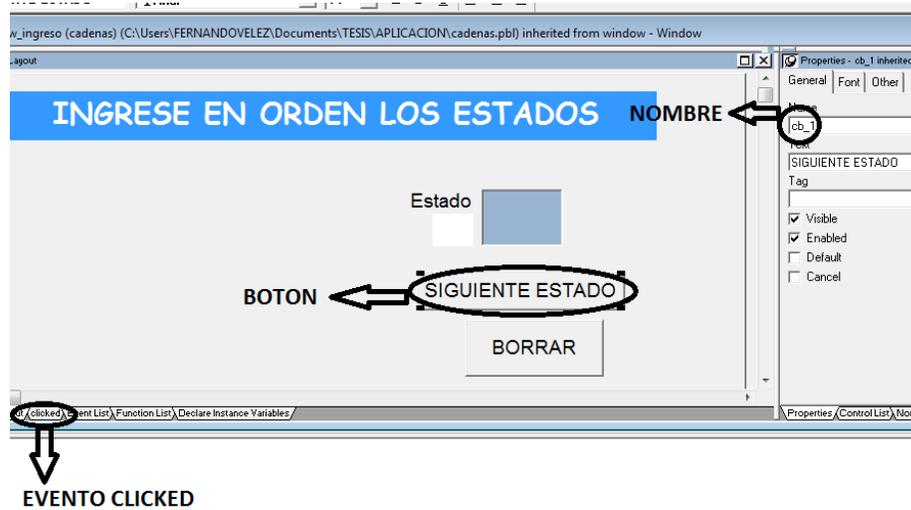
## **2.2. POWERBUILDER 9.0**

PowerBuilder es un medio ambiente gráfico de desarrollo de aplicaciones cliente/servidor. Usando PowerBuilder, se puede desarrollar fácilmente poderosas aplicaciones gráficas orientadas a objetos, que tienen acceso a bases de datos locales o en el servidor, haciendo uso de una interfaz atractiva para el usuario, el cual le proporciona las herramientas necesarias para crear una gran variedad de aplicaciones.

La interfaz de usuario de una aplicación PowerBuilder consiste de menús y ventanas con las cuales interactúa el usuario. Las aplicaciones pueden incluir todos los controles windows standard, tales como: buttons, checkboxes, dropdown listboxes y edit boxes, así como controles especiales PowerBuilder que permiten que las aplicaciones sean fáciles de desarrollar y usar.

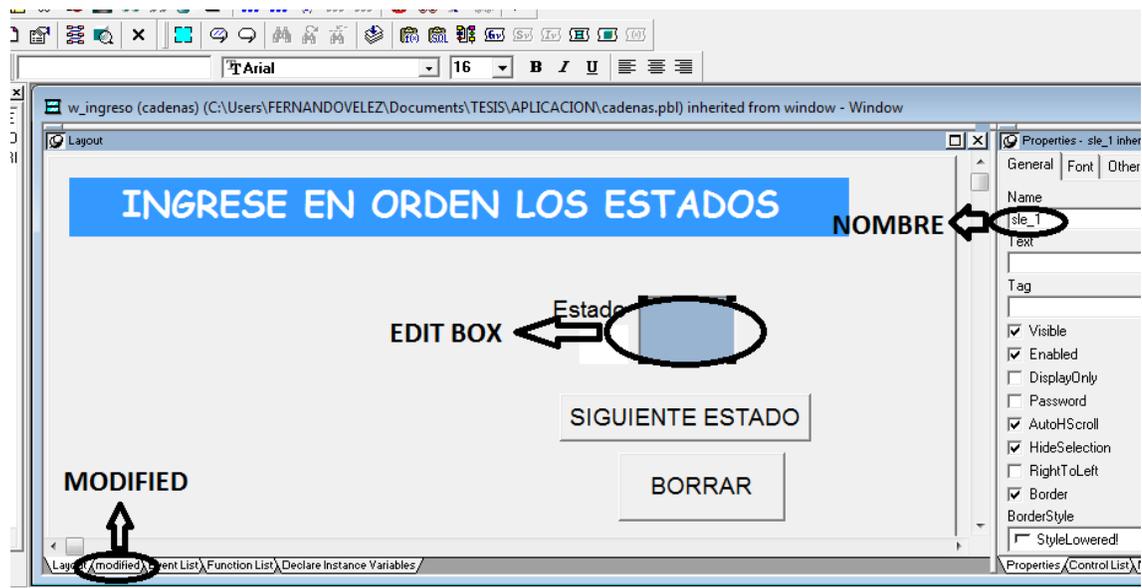
En una aplicación, el usuario controla lo que sucede a través de las acciones que toma. Por ejemplo, cuando un usuario le da click a un botón, selecciona un ítem del menú, o ingresa datos en un edit box, se ejecutan uno o más eventos de acuerdo a como esté programada dicha aplicación.

El usuario digita las instrucciones que especifican el proceso que debe ejecutarse cuando los eventos se ejecutan, por ejemplo, uno de los eventos de los botones se llama 'Clicked'. El usuario escribe un script para el evento Clicked, el cual especifica que sucederá cuando le dé un click a ese botón, ver figura 1.



**Figura 1.** Descripción de los elementos básicos de Power Builder 9.0

De modo similar, las edit boxes o cajas de edición, tienen un evento llamado 'Modified', el cual se ejecuta cada vez que el usuario cambia un valor en la caja. (Ver Figura 2.)



**Figura 2.** Descripción elementos modificables Power Builder 9.0

### 3. DISEÑO

A continuación se presenta una descripción del proceso de diseño del software CMTD10, desde el planteamiento del proyecto, hasta su culminación.

#### 3.1. ANTECEDENTES

Los trabajos realizados sobre cadenas de Márkov, se encuentran en universidades donde se ofrecen carreras de ingeniería, tal es el caso de un trabajo realizado por German Riaño<sup>1</sup>, profesor de la Universidad de los Andes quien hizo público un software en la red, el cual permite guardar, leer, generar y cambiar las condiciones iniciales de la matriz de transición, pero en un ambiente visual de difícil comprensión.

También el realizado por Olga García y Felipe Nieto<sup>2</sup> estudiantes de Ingeniería Industrial de la Universidad Javeriana en el año 2004, los cuales realizaron una tesis, diseñando un software educativo para ser utilizado por los estudiantes de la misma carrera, que presentaban dificultades en el espacio académico procesos estocásticos. En la actualidad en el Departamento de Matemáticas de la Universidad Pedagógica Nacional, no se encuentran trabajos sobre el tema.

Dada la dificultad de plantear las matrices de transición y realizar el análisis utilizando las cadenas de Márkov, es necesario hacer uso de una herramienta tecnológica que simplifique y agilice los cálculos, se espera ofrecer un software de libre consulta y uso.

---

<sup>1</sup> Riaño., G. (sf) Cadenas de Márkov, Universidad de los Andes, en línea <http://copa.uniandes.edu.co/software/Markov/markov.html>

<sup>2</sup> Garcia., O., Nieto., F., (2004) Software educativo: Cadenas de Márkov en tiempo discreto. Tesis de grado para recibir el título de Ingeniero industrial en la Pontificia Universidad Javeriana.

## 3.2. PROCESO DE ELABORACIÓN

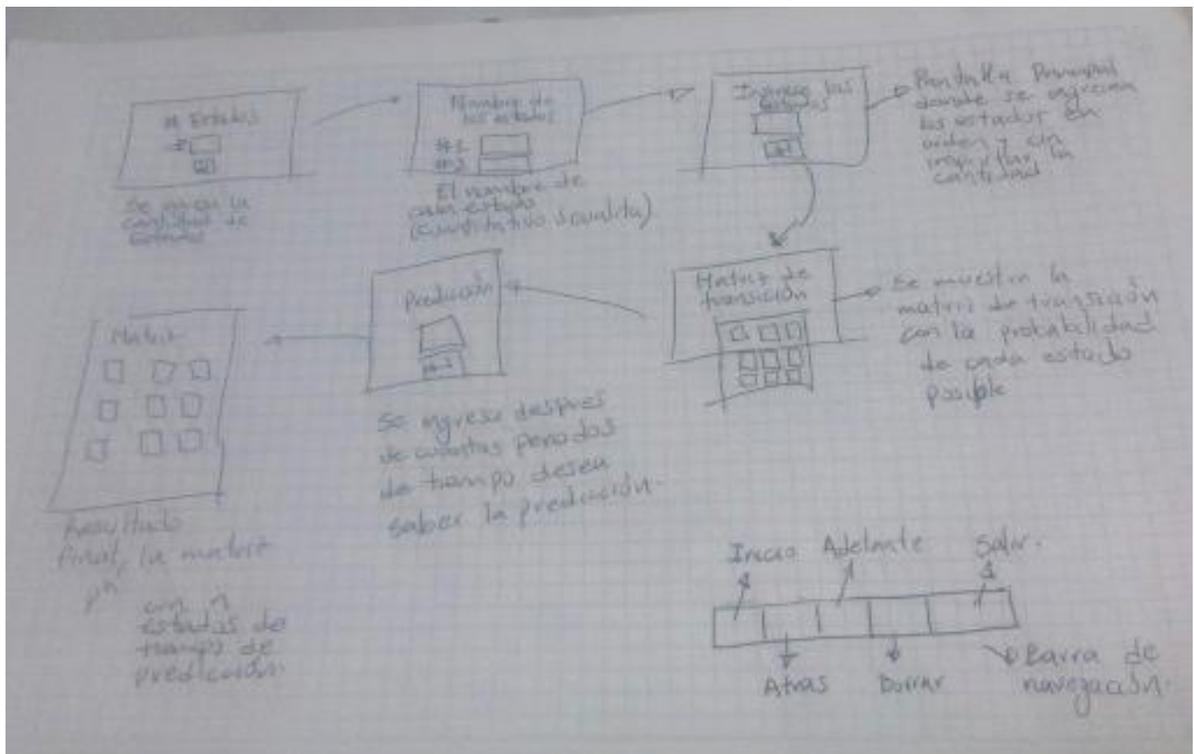
Durante la elaboración se pasaron por diferentes etapas, desde el bosquejo de las pantallas hasta el resultado final, a continuación se describen algunos aspectos positivos y diferentes complicaciones que se presentaron a lo largo del proceso de elaboración.

- 1- **Describir el software:** Se realizó una descripción del software que se quería desarrollar, esta fue la base para poder comenzar a diseñar las diferentes ventanas y el estilo de la herramienta, teniendo en cuenta el alcance que debía tener y la función específica que iba a desempeñar.

Las ideas que se tuvieron en cuenta fueron:

- a) Un software que recogiera los datos, que el usuario los ingresara en una pantalla.
- b) Que el usuario ingresara el nombre de cada estado y la cantidad, para que al final el software utilizara estos nombres para el análisis final.
- c) El software a diseñar debía arrojar la matriz de transición después que el usuario ingresara los estados en su totalidad, sin tener un límite en la cantidad ingresada.
- d) Que el programa realizara los cálculos de multiplicación entre matrices para poder obtener una predicción de la probabilidad de un estado futuro.

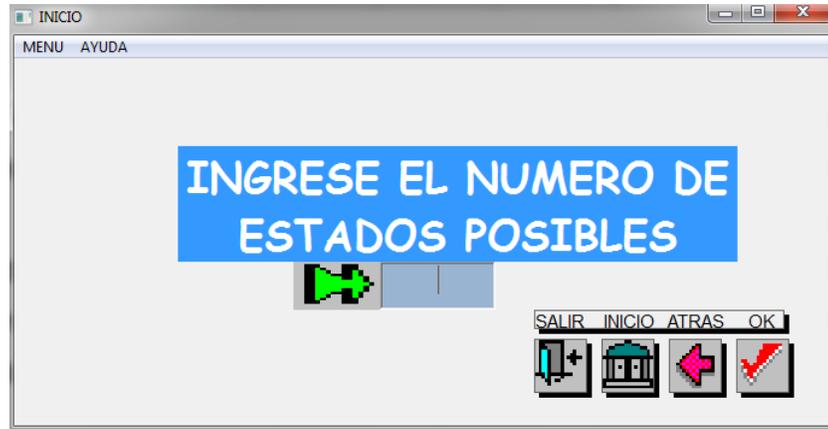
2- **Bosquejo o borrador:** Se dibujó cada pantalla y se describió la función de cada una, además del diseño y los botones que debía tener para la navegación, aunque con las modificaciones que se fueron haciendo, el diseño de dichas ventanas fue modificado en algunas ventanas, hubieron otras que no hubo necesidad de programar.



**Figura 3.** Bosquejo del diseño de las ventanas y sus respectivas funciones.

3- **Programación:** De acuerdo a lo planeado y la descripción del software, se procedió a programar cada una de las ventanas, en un principio se pensaba realizar una ventana por cada número de estados, pero era demasiada información repetida, entonces se utilizó una sola aprovechando una función de ocultar los botones o cuadros de comando que no se utilizarían. Al final solo se utilizaron 5 ventanas.

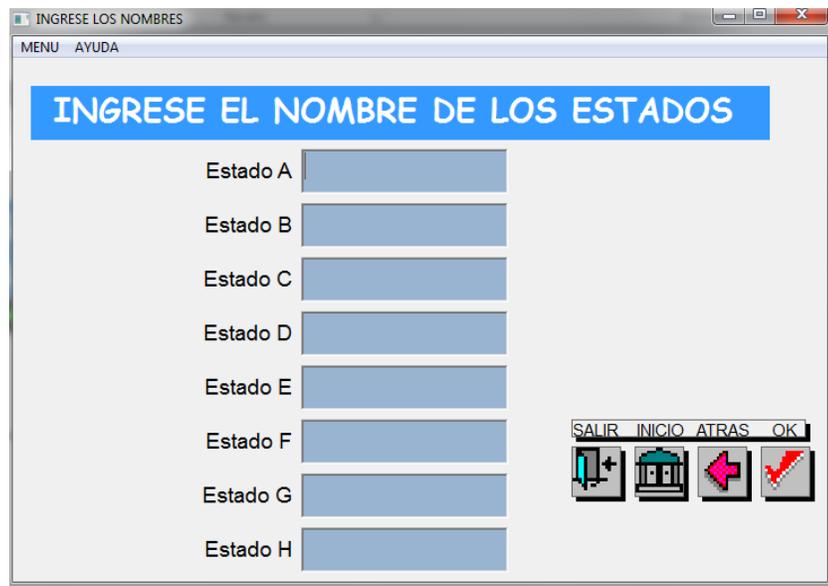
## Pantalla 1



**Figura 4.**

En esta primera pantalla se ingresa la cantidad de estados posibles del estudio que se vaya a realizar, en el cuadro gris se debe ingresar un número natural entre 2 y 8 (Figura 4.)

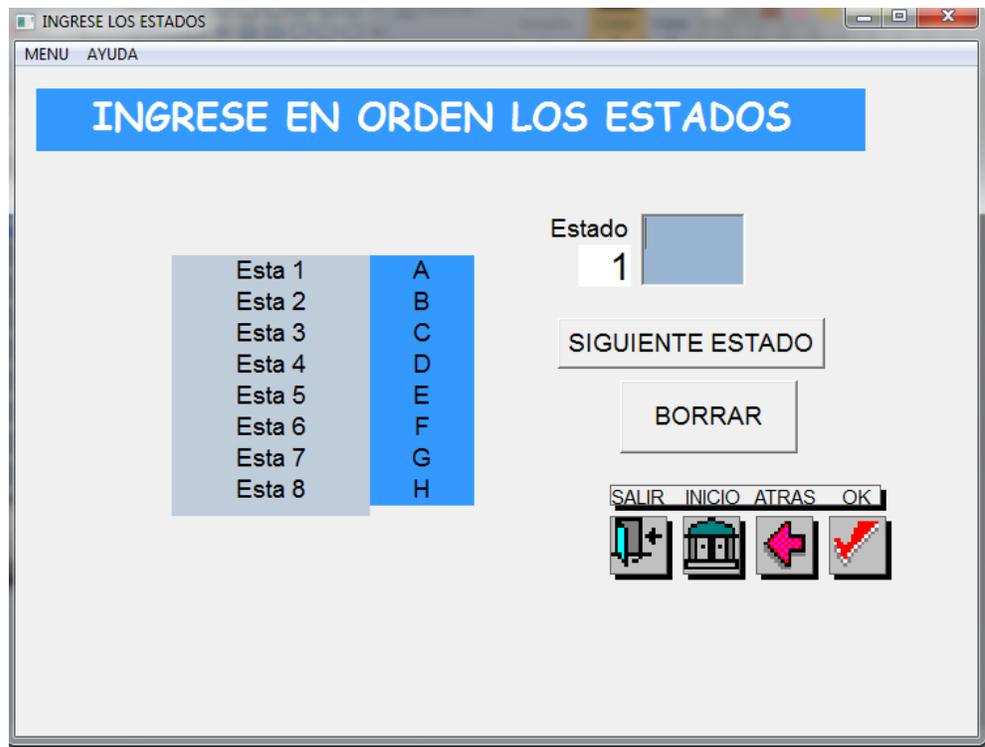
## Pantalla 2



**Figura 5.**

Se deben ingresar el nombre de los estados para poder identificar las probabilidades en la matriz de transición final y la predicción a futuro que se vaya a realizar. (Figura 5.)

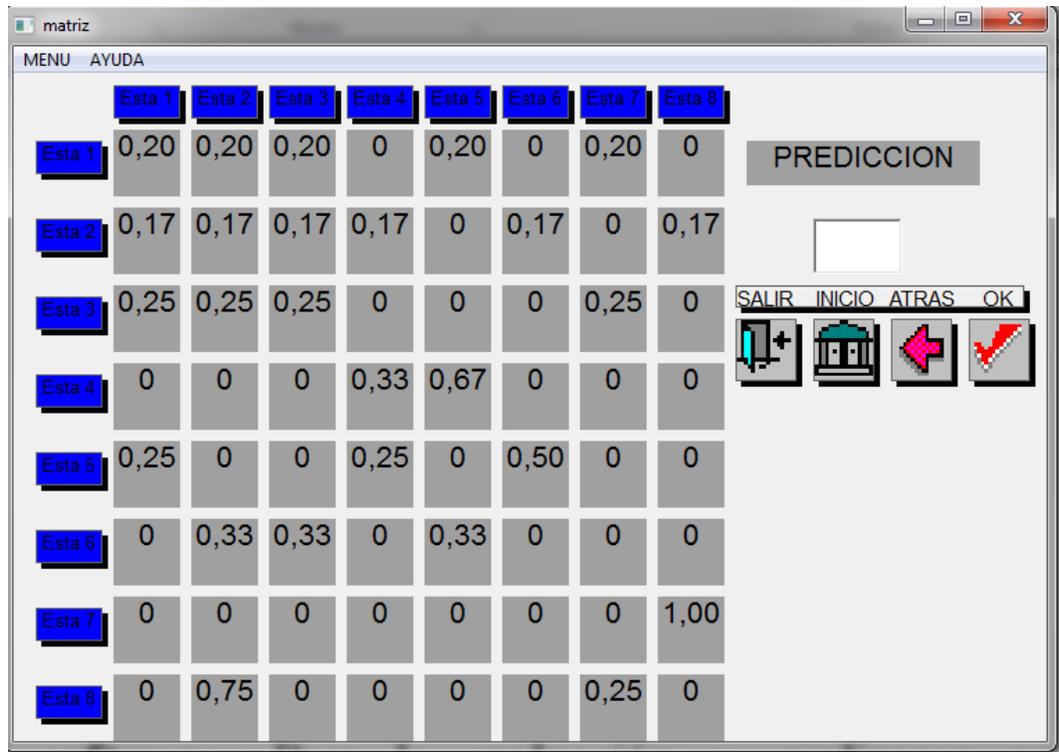
### Pantalla 3



**Figura 6.**

En esta pantalla se deben ingresar en orden los estados del estudio a realizar, cada vez que se ingresa un estado en el recuadro gris se debe dar click en el botón siguiente estado, al terminar de ingresar los estados en su totalidad, se debe dar click en el chulito rojo de ok, en caso de equivocarse al ingresar algún dato, la única opción es presionar el botón BORRAR para que vuelva a comenzar desde el primer estado a ingresarlos. (Figura 6)

## Pantalla 4

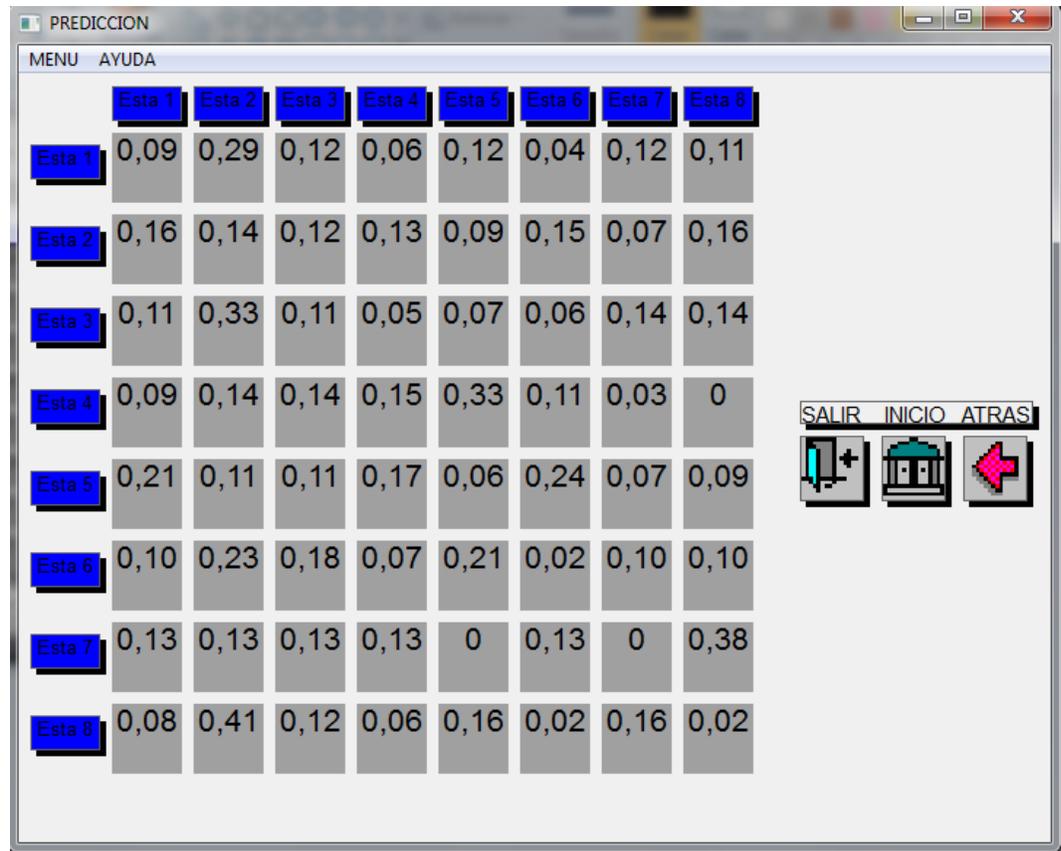


**Figura 7.**

Se muestra la matriz de transición después de haber ingresado en su totalidad todos los estados, se observa que los nombres se encuentran en azul oscuro en cada fila para poder determinar cada probabilidad, como por ejemplo la probabilidad de encontrarse en el estado 4 y pasar al estado 5 es del 67% y la probabilidad de encontrarse en el estado 3 y permanecer en ese mismo estado es del 25%, pero la probabilidad de encontrarse en el estado 5 y pasar al estado 3 es del 0%. (Figura 7)

Al lado derecho se encuentra un recuadro para ingresar la cantidad de periodos de tiempo posteriores del cual se desea tener la predicción, se debe ingresar la cantidad y dar click en ok.

## Pantalla 5



	Esta 1	Esta 2	Esta 3	Esta 4	Esta 5	Esta 6	Esta 7	Esta 8
Esta 1	0,09	0,29	0,12	0,06	0,12	0,04	0,12	0,11
Esta 2	0,16	0,14	0,12	0,13	0,09	0,15	0,07	0,16
Esta 3	0,11	0,33	0,11	0,05	0,07	0,06	0,14	0,14
Esta 4	0,09	0,14	0,14	0,15	0,33	0,11	0,03	0
Esta 5	0,21	0,11	0,11	0,17	0,06	0,24	0,07	0,09
Esta 6	0,10	0,23	0,18	0,07	0,21	0,02	0,10	0,10
Esta 7	0,13	0,13	0,13	0,13	0	0,13	0	0,38
Esta 8	0,08	0,41	0,12	0,06	0,16	0,02	0,16	0,02

**Figura8.**

En esta última pantalla se observa la matriz de predicción en un periodo de tiempo de 3 transiciones, aunque en la matriz de transición se observaba que la probabilidad de estar en el estado 5 y pasar al estado 3 era nula, en esta se observa una probabilidad del 11%, todo por cómo se ha comportado el sistema y las diferentes transiciones que ha sufrido con el paso del tiempo. (Figura 8)

## Barra de navegación



**Figura 9.** Barra de navegación

Esta barra de navegación se encuentra en todas las pantallas del software y cuenta con 4 o 3 botones dependiendo el caso, el botón salir permite cerrar la ventana actual y cerrar el programa, al oprimir este botón se borrarán todos los datos que tengamos almacenados, ya sean los nombres de los estados, la cantidad, los estados o los datos de los arreglos de las matrices. (Figura 9)

El botón inicio envía al usuario a la pantalla principal del programa y permite volver a realizar todo el procedimiento desde el comienzo, ingresando la cantidad de estados y demás.

El botón atrás se debe utilizar en caso que quiera volver a la página anterior para realizar algún cambio o para verificar algún dato y por último el botón ok, permite avanzar a la siguiente pantalla, en caso de no cumplir con todos los elementos suficientes y necesarios, emergerá una ventana de advertencia para que los campos obligatorios sean diligenciados completa y adecuadamente.

**4- Evaluación del software:** Para la evaluación se realizaron pruebas con ejercicios hechos a mano y se verificaban los resultados en el software, además como una prueba de la utilidad del software, se realizó una prueba en una situación de la vida real en la Universidad Pedagógica Nacional. Este análisis se muestra en el numeral 4.

**Ejemplo:**

Un vendedor de computadoras de un almacén de cadena, lleva un registro de la cantidad de computadores vendidos cada día, los datos de los últimos 30 días los consigna en la siguiente tabla (Tabla 2).

**Tabla 2.** *Cantidad de computadoras vendidas diariamente.*

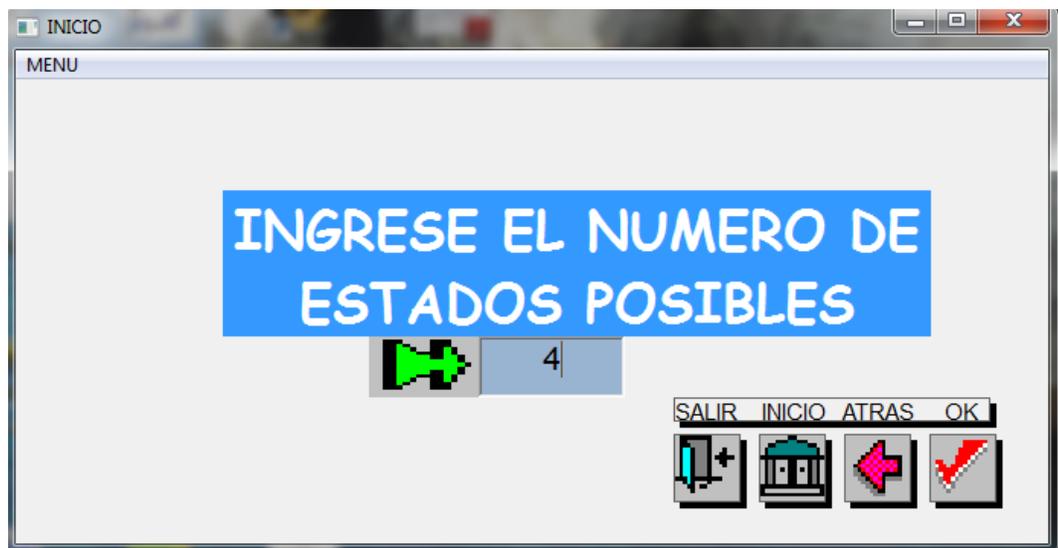
<b>DÍA</b>	<b>CANTIDAD</b>	<b>DÍA</b>	<b>CANTIDAD</b>	<b>DÍA</b>	<b>CANTIDAD</b>
1	CERO	11	CERO	21	UNO
2	UNO	12	CERO	22	CERO
3	UNO	13	UNO	23	TRES
4	DOS	14	DOS	24	UNO
5	TRES	15	DOS	25	DOS
6	TRES	16	UNO	26	DOS
7	CERO	17	TRES	27	DOS
8	UNO	18	DOS	28	TRES
9	UNO	19	UNO	29	UNO
10	DOS	20	UNO	30	CERO

En este caso se tiene un estado *cuantitativo*, se observa además que ningún día logro vender cuatro computadoras o más, esto nos indica que los estados para realizar el análisis con las

cadenas de Márkov son la cantidad de computadoras vendidas cada día (cero, una, dos o tres).

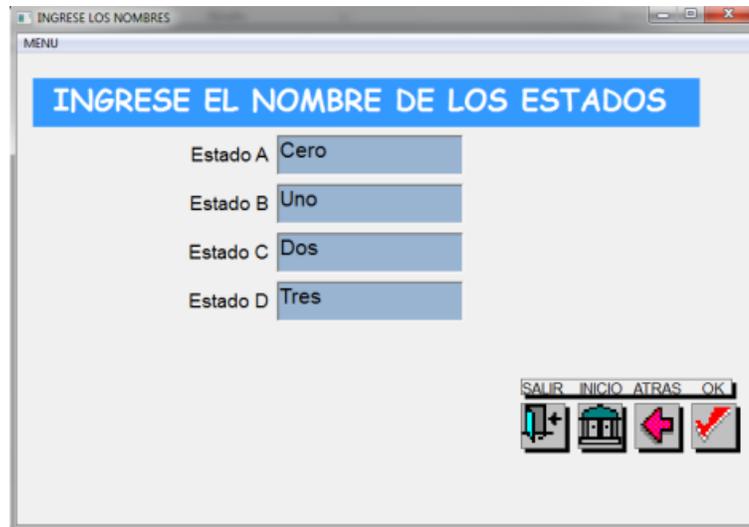
A continuación se describe el paso a paso para ingresar los datos en el software CMTD10 y los procedimientos que realiza para obtener la matriz de transición y la predicción a futuro.

1. Se ingresa la cantidad de estados, luego click en “OK”



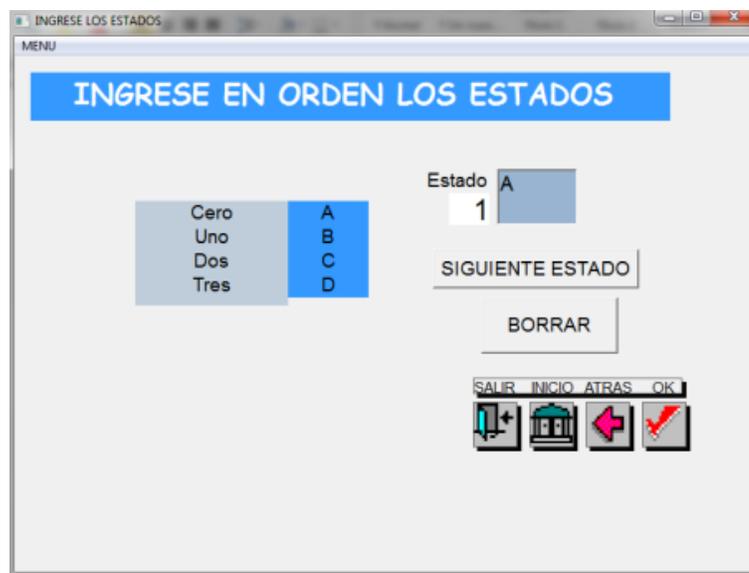
**Figura 10.**

2. Se ingresa el nombre de cada estado, luego click en “OK”.



**Figura 11.**

3. Se deben ingresar en orden los estados comenzando con el número cero, al cual le corresponde la letra A, luego click en “SIGUIENTE ESTADO” y se repite el procedimiento para cada estado ingresado.



**Figura 12.**

Al ingresar cada estado, en software cuenta las transiciones que se realizan entre cada par de estados y los organiza en una matriz cuadrada de 4x4, dependiendo el estado actual y el estado al cual hace la transición el sistema, cada vez un contador por casilla va aumentando. Es decir si se el primer día vendió cero computadores y el segundo día vendió una (Paso de vender cero a vender una), la matriz en la primera transición es:

$$\begin{array}{c}
 \text{Cero} \\
 \text{Uno} \\
 \text{Dos} \\
 \text{Tres}
 \end{array}
 \begin{array}{cccc}
 \text{Cero} & \text{Uno} & \text{Dos} & \text{Tres} \\
 \left( \begin{array}{cccc}
 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0
 \end{array} \right)
 \end{array}$$

Como al tercer día vendió una computadora y el segundo día también, entonces la matriz que genera el sistema es

$$\begin{array}{c}
 \text{Cero} \\
 \text{Uno} \\
 \text{Dos} \\
 \text{Tres}
 \end{array}
 \begin{array}{cccc}
 \text{Cero} & \text{Uno} & \text{Dos} & \text{Tres} \\
 \left( \begin{array}{cccc}
 0 & 1 & 0 & 0 \\
 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0
 \end{array} \right)
 \end{array}$$

El cuarto día paso de vender una computadora a vender 2, la matriz generada es:

$$\begin{array}{c}
 \text{Cero} \\
 \text{Uno} \\
 \text{Dos} \\
 \text{Tres}
 \end{array}
 \begin{array}{cccc}
 \text{Cero} & \text{Uno} & \text{Dos} & \text{Tres} \\
 \left( \begin{array}{cccc}
 0 & 1 & 0 & 0 \\
 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0
 \end{array} \right)
 \end{array}$$

4. Se continúan ingresando los estados hasta el último y después de haberlo guardado, click en “OK”



**Figura 13.**

Al finalizar el ingreso de los estados se obtiene la matriz

$$\begin{array}{cccc}
 & \text{Cero} & \text{Uno} & \text{Dos} & \text{Tres} \\
 \text{Cero} & \begin{pmatrix} 1 & 3 & 0 & 1 \end{pmatrix} \\
 \text{Uno} & \begin{pmatrix} 2 & 4 & 4 & 1 \end{pmatrix} \\
 \text{Dos} & \begin{pmatrix} 1 & 2 & 3 & 2 \end{pmatrix} \\
 \text{Tres} & \begin{pmatrix} 1 & 2 & 1 & 1 \end{pmatrix}
 \end{array}$$

Se observa que en la casilla (Cero, Dos) de la matriz hay un cero, esto indica que el vendedor nunca vendió dos computadoras después de no haber vendido ninguna.

Luego se divide cada fila por la sumatoria de transiciones de cada estado, entonces la primera fila se debe dividir por 5, por qué el sistema tuvo 5 transiciones cuando no se vendieron computadoras y así con las demás filas de la siguiente manera:

$$\begin{array}{c}
 \text{Cero} \\
 \text{Uno} \\
 \text{Dos} \\
 \text{Tres}
 \end{array}
 \begin{pmatrix}
 \text{Cero} & \text{Uno} & \text{Dos} & \text{Tres} \\
 \frac{1}{5} & \frac{3}{5} & \frac{0}{5} & \frac{1}{5} \\
 \frac{2}{11} & \frac{4}{11} & \frac{4}{11} & \frac{1}{11} \\
 \frac{1}{8} & \frac{2}{8} & \frac{3}{8} & \frac{2}{8} \\
 \frac{1}{5} & \frac{2}{5} & \frac{1}{5} & \frac{1}{5}
 \end{pmatrix}$$

Al realizar las divisiones, se obtiene la siguiente matriz.

$$\begin{array}{c}
 \text{Cero} \\
 \text{Uno} \\
 \text{Dos} \\
 \text{Tres}
 \end{array}
 \begin{pmatrix}
 \text{Cero} & \text{Uno} & \text{Dos} & \text{Tres} \\
 0,2 & 0,6 & 0 & 0,2 \\
 0,18 & 0,36 & 0,36 & 0,9 \\
 0,13 & 0,25 & 0,38 & 0,25 \\
 0,2 & 0,4 & 0,2 & 0,2
 \end{pmatrix}$$

Ésta es la denominada matriz de transición de las cadenas de Márkov, donde la sumatoria de cada fila es uno y muestra todas las probabilidades de transición del sistema, todo dependiendo del estado actual.

5. Obtenemos la matriz de transición.

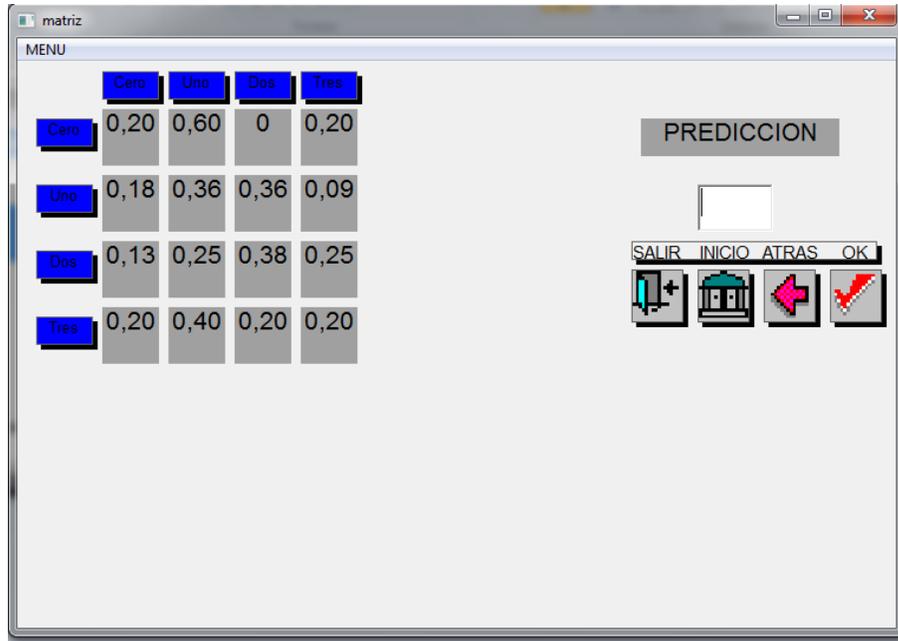


Figura 14.

Si se quiere saber una predicción del porcentaje de venta del siguiente día se debe elevar la matriz de transición a la segunda potencia, es decir multiplicarla por sí misma una vez así:

$$\begin{pmatrix} 0,2 & 0,6 & 0 & 0,2 \\ 0,18 & 0,36 & 0,36 & 0,9 \\ 0,13 & 0,25 & 0,38 & 0,25 \\ 0,2 & 0,4 & 0,2 & 0,2 \end{pmatrix} \times \begin{pmatrix} 0,2 & 0,6 & 0 & 0,2 \\ 0,18 & 0,36 & 0,36 & 0,9 \\ 0,13 & 0,25 & 0,38 & 0,25 \\ 0,2 & 0,4 & 0,2 & 0,2 \end{pmatrix} =$$

Al realizar el producto de las matrices se obtiene

$$\begin{pmatrix} 0,19 & 0,42 & 0,26 & 0,13 \\ 0,17 & 0,37 & 0,29 & 0,18 \\ 0,18 & 0,37 & 0,28 & 0,2 \\ 0,18 & 0,39 & 0,26 & 0,17 \end{pmatrix}$$

6. Al ingresar un periodo de tiempo igual a dos, obtenemos la siguiente predicción.

	Cero	Uno	Dos	Tres
Cero	0,19	0,42	0,26	0,13
Uno	0,17	0,37	0,29	0,18
Dos	0,18	0,37	0,28	0,20
Tres	0,18	0,39	0,26	0,17

**Figura 15.**

Al vendedor le hacen falta 2 computadores más para alcanzar la meta propuesta para una bonificación y quiere saber qué tan probable es que al día siguiente venda los dos computadores. Como el día treinta no vendió ningún computador, el estado del ultimo día fue cero, entonces al observar la primera fila de la matriz predicción, se determina que hay un 19% de probabilidad que al día siguiente no venda ninguna computadora; una probabilidad del 42% de vender solo una; una probabilidad del 26% de vender las dos computadoras que necesita y una probabilidad del 13% de vender tres computadoras.

Al hacer un análisis global el vendedor, determina que hay una probabilidad del 39% de cumplir con la meta y una probabilidad del 61% de no lograrlo, esto lo deduce después sumar las probabilidades de vender dos computadoras con la probabilidad de vender 3, puesto que en ambos casos lograría cumplir con la meta establecida.

### **3.3. MANUAL DE INSTALACIÓN.**

Las aplicaciones desarrolladas con PowerBuilder se ejecutan exclusivamente en el sistema operativo Microsoft Windows, los requisitos mínimos del sistema para instalar Powerbuilder 9.0 y ejecutar aplicaciones en la computadora son los siguientes:

#### **Requisitos de software (Una de las siguientes versiones de Microsoft Windows)**

- ✓ Microsoft Windows 2000 SP4.
- ✓ Windows XP SP3.
- ✓ Windows Vista SP2.
- ✓ Windows 7.
- ✓ Windows 8.

#### **Requisitos de hardware**

- ✓ Un procesador de clase Pentium funcionando a 1 GHz o más rápido
- ✓ Por lo menos 512 MB de RAM

## Pasos a seguir

- ✓ En la carpeta SOFTWARE se encuentra el archivo CMTD10.rar (Figura 16.)

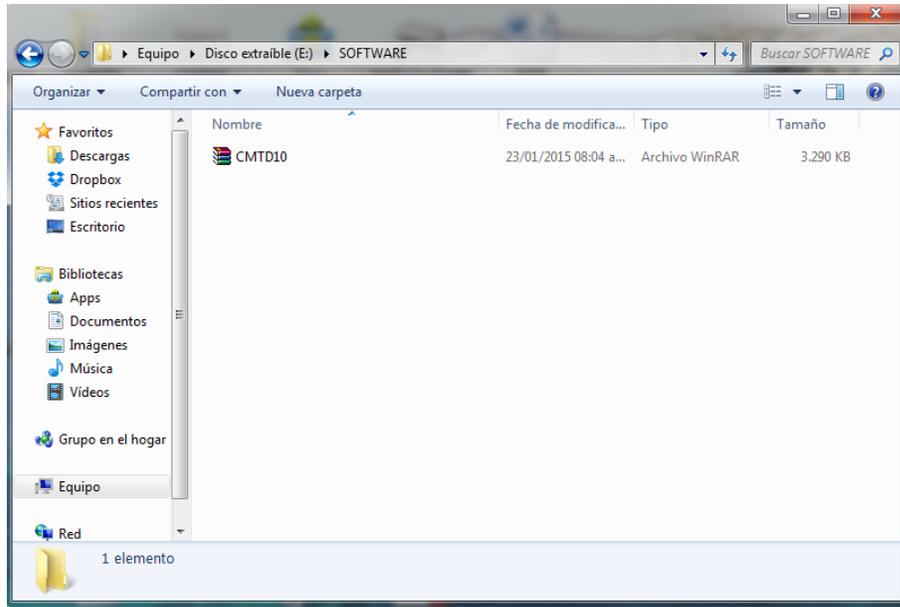


Figura 16.

- ✓ Se debe descomprimir la carpeta CMTD10 en la misma carpeta, click derecho y “Extraer aquí” (figura 17)

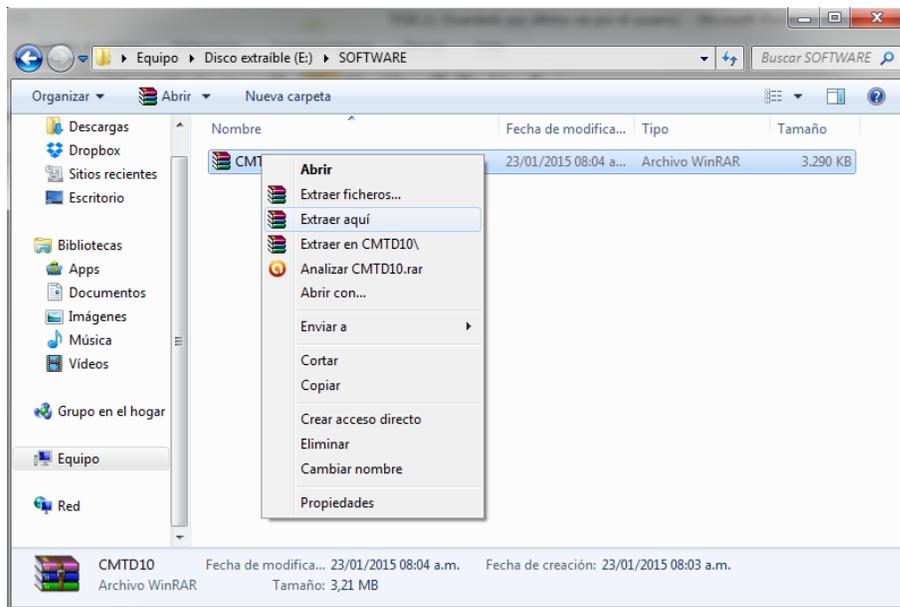
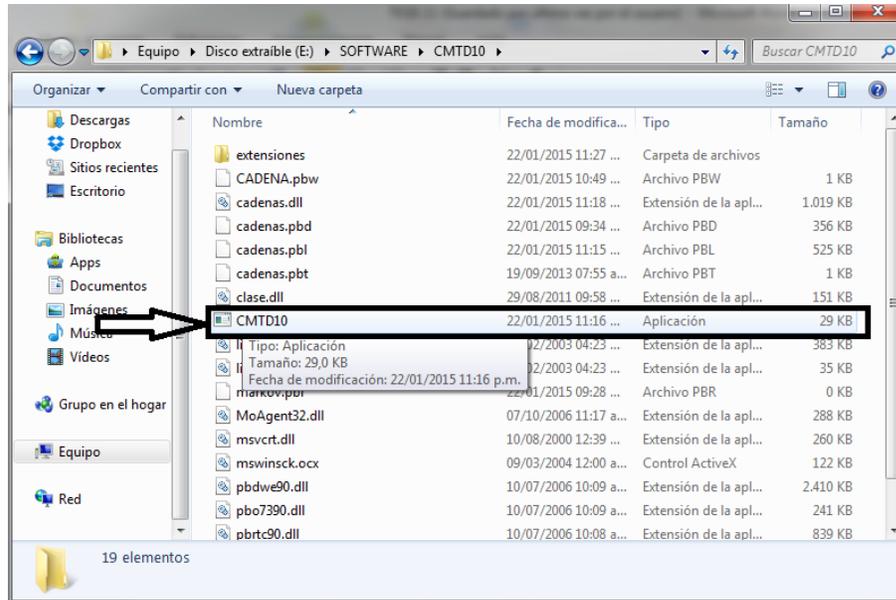


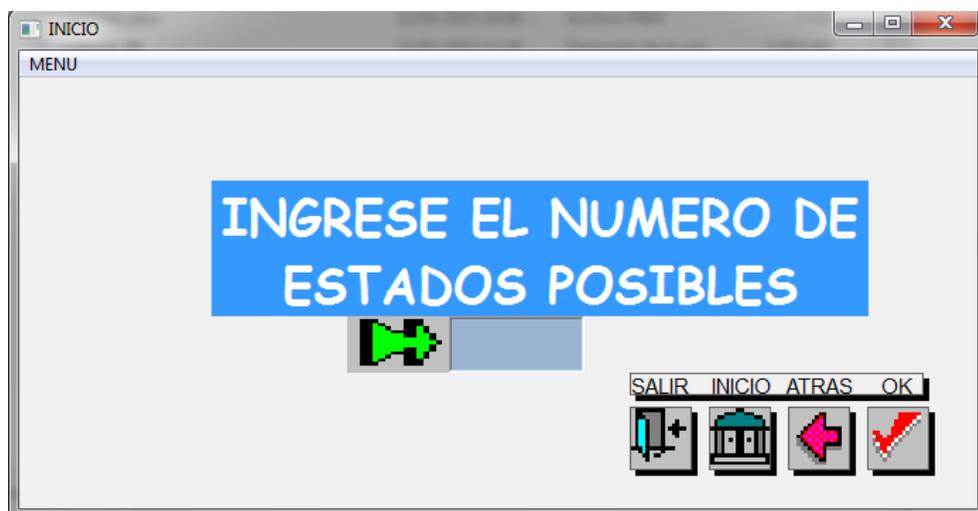
Figura 17.

- ✓ En la carpeta se encuentra el archivo “CMTD10.exe”, (doble click). (Ver Figura 18.)



**Figura 18.**

- ✓ Emergerá la primera pantalla (Figura 19.)



**Figura 19.**

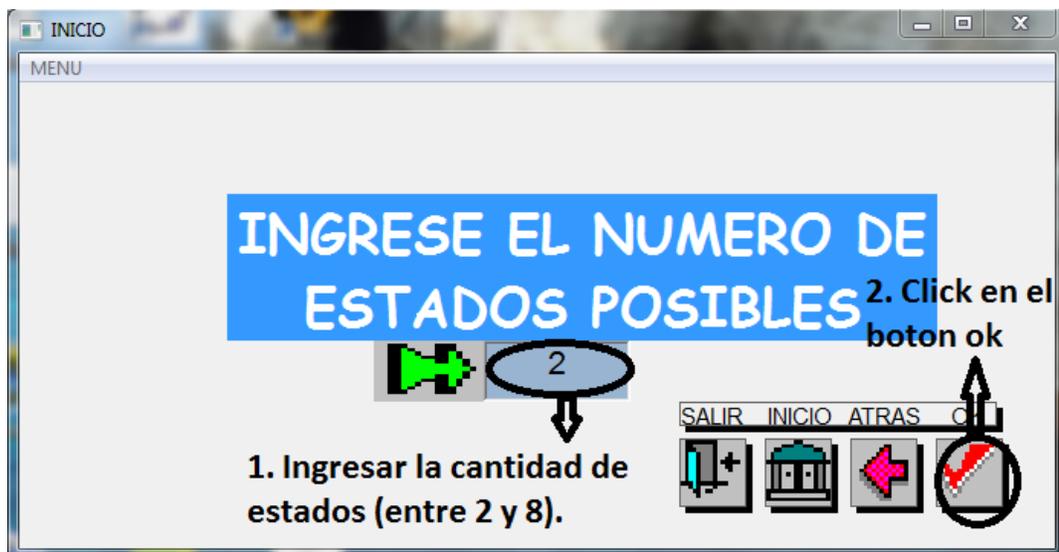
### 3.3.1. INSTRUCCIONES DE USO

Para el debido uso de la herramienta CMTD10, a continuación se presenta paso a paso como se deben ingresar los datos y los botones que debemos presionar para poder obtener los resultados adecuadamente, lo realizaré con el ejemplo de los estados del tiempo del numeral 2.1.3. Y se observa en la tabla 1.

**Tabla1.** *S=Soleado N= Nublado*

Día	1	2	3	4	5	6	7	8	9	10
Estado del tiempo	S	S	S	S	N	N	N	S	N	N

#### Pantalla #1



**Figura 20.**

Tal como se observa en la Figura 20., en esta pantalla se encuentra el espacio para ingresar la cantidad de estados con los cuales se va a realizar el trabajo con las cadenas de Márkov,

se ingresa el número y se da click en ok, en este caso se ingresa el número dos por que tenemos dos estados posibles (Soleado y Nublado).

En caso de ingresar un valor distinto, emergerá una ventana que advierte que el valor ingresado debe estar entre 2 y 8. (Ver Figura 21.)

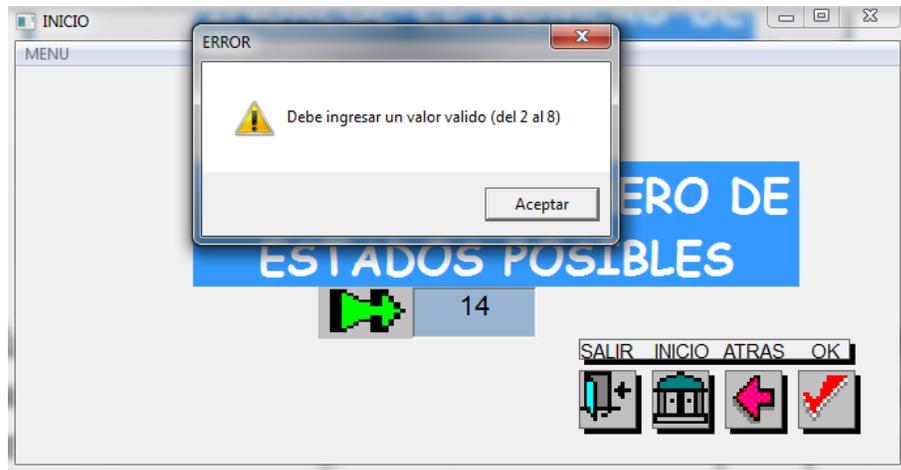


Figura 21.

## Pantalla #2

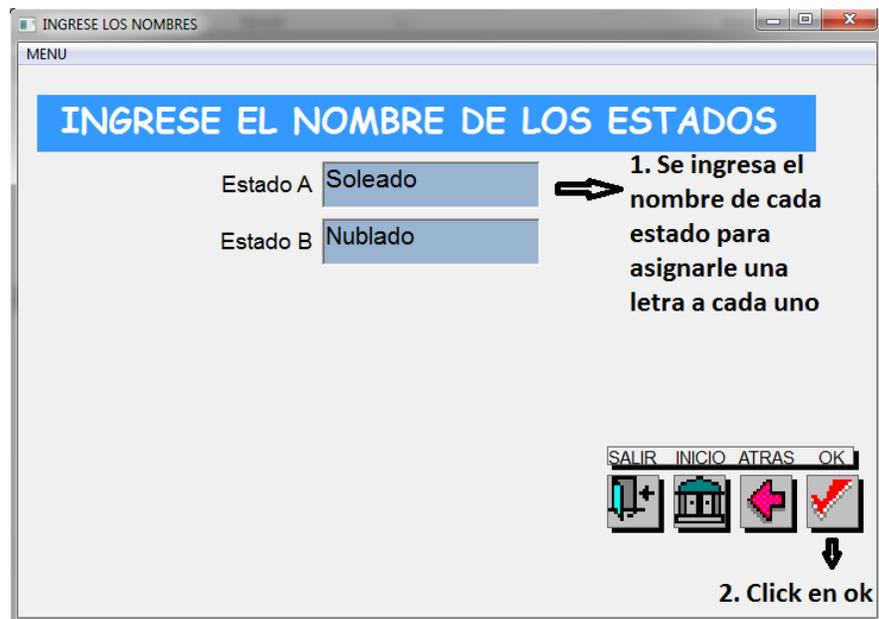


Figura 22.

En la Figura 22 se observa la segunda pantalla, en la cual se debe ingresar el nombre de cada estado, teniendo presente que la letra que se le asigna a cada estado es la que se debe ingresar más adelante, si no se ingresan los valores de todos los campos, no se podrá avanzar a la siguiente pantalla.



**Figura 23.**

### Pantalla #3

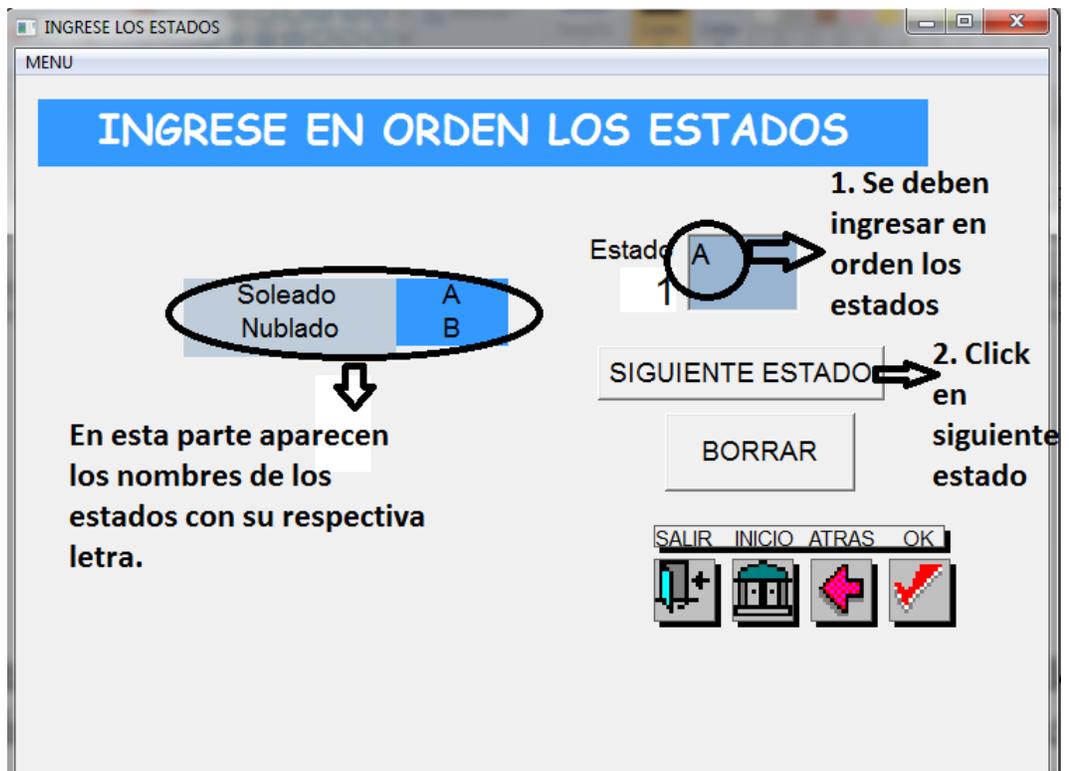


Figura 24.

Se debe ingresar la letra del estado en el recuadro señalado, después de haber ingresado el primer estado se da click en siguiente estado (Ver Figura 25.) y el programa ira guardando el orden de los estados en un arreglo unidimensional, para luego poder realizar los cálculos de una cadena de Márkov.

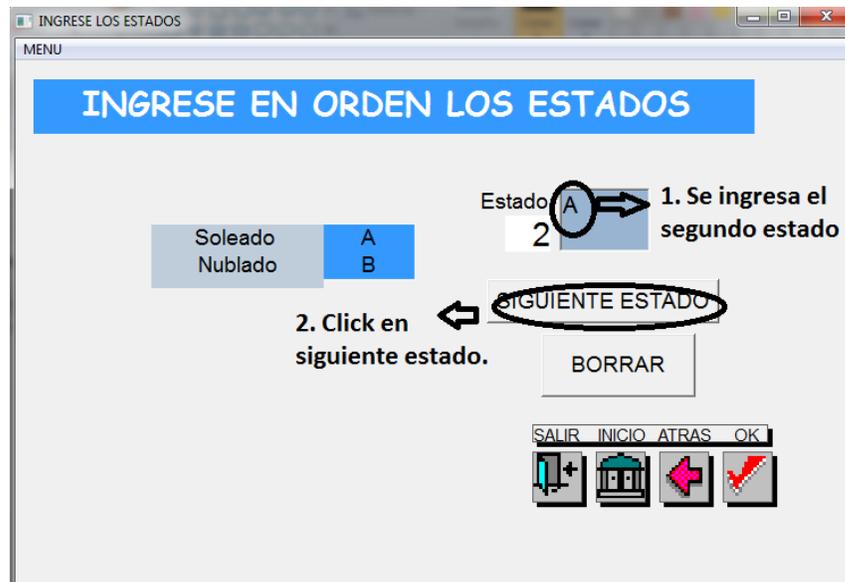


Figura 25.

Se continua ingresando los estados en orden, el usuario decide cuando parar de ingresar estados y dar click en ok, en caso de ingresar un valor distinto a los correspondientes de los estados, el programa no lo lee, es decir solo guarda valores correctos.

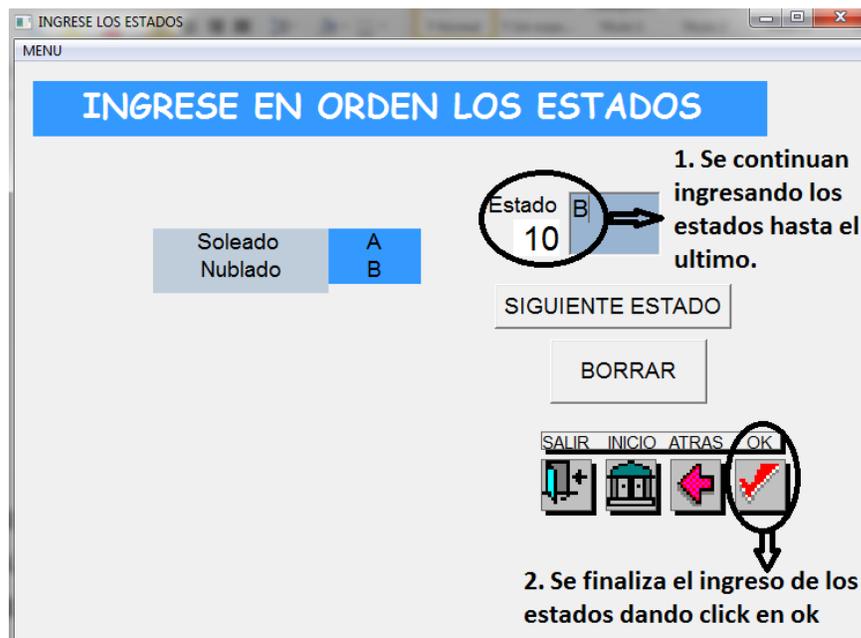


Figura 26.

## Pantalla #4

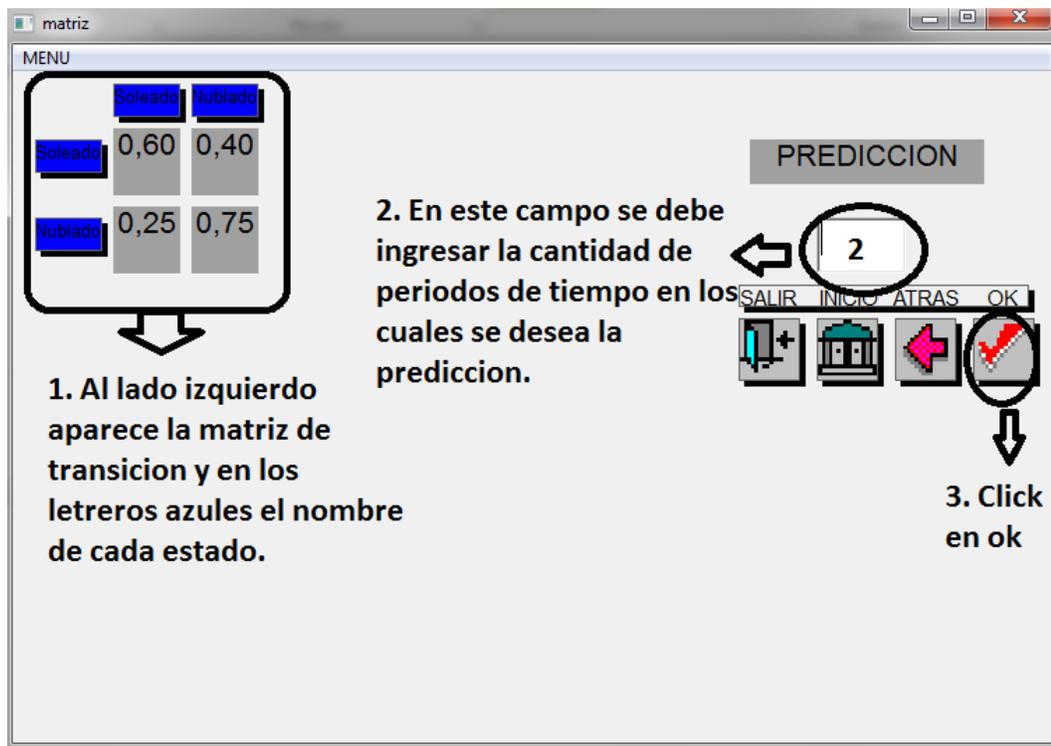
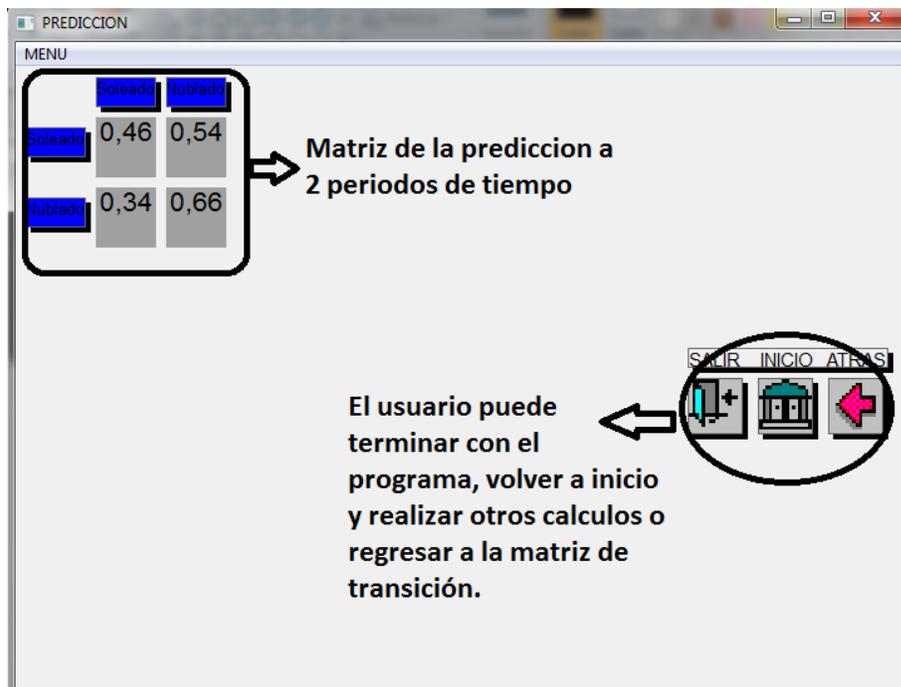


Figura 27.

Para efectos del ejemplo, se puede observar que al realizar el procedimiento de la cadena de Márkov, obtenemos una matriz de transición la cual indica que si el sistema se llegara a encontrar en el estado soleado, habría una probabilidad de un 60% que al siguiente día fuera de nuevo soleado y de un 40% que al siguiente día fuera nublado; pero si el sistema se encontrara en estado nublado, habría una probabilidad del 25% que el siguiente día fuera soleado y de un 75% que al día siguiente fuera nublado también.

## Pantalla #5



**Figura 28.**

En esta pantalla final (Figura 28.) se observa la matriz de predicción con sus respectivos porcentajes de probabilidad, el usuario tiene la opción de dar click en el botón inicio para volver al comienzo y realizar otros cálculos distintos, si ya termino y la información que arrojó el software es completa, debe dar click en salir, o también puede volver a la pantalla anterior para modificar el periodo de tiempo en el cual quiere hacer la predicción.

Cabe aclarar que si el programa regresa al inicio, todos los datos serán reiniciados, solo se almacenan los datos cuando regresamos a la pantalla anterior.

### 3.4. CÓDIGO POWERBUILDER 9.0

El programa CMTD10 realiza principalmente tres grupos de instrucciones, el primero es recoger la información necesaria, el segundo es el proceso y los cálculos que realiza con dicha información y el tercero es mostrar los resultados después de haber realizado todo el procedimiento. A continuación se presenta el código que se utiliza para cada una de estas instrucciones y la descripción ejemplificada de la sintaxis que utiliza *Power Builder 9.0*

✓ **Recolección de la información.**

<b>Instrucción</b>	<b>Sintaxis (código)</b>	<b>Descripción</b>
<b>Asignar un valor a una variable.</b>	<code>es=dec(em_1.text)</code>	Se asigna el valor ingresado en un “editmask 1” a la variable “es”  Se utiliza para saber con cuantos estados se va a trabajar, se puede modificar en las opciones para que solo admita valores numéricos o letras también.
<b>Declaración de variables.</b>	<code>dec s1, s2, s3</code>  <code>int h</code>  <code>dec{2} mostrar [1 to 8,1 to 8]</code>	Las variables pueden ser globales si se van a utilizar en todo el programa, para que guarden valores que pueden ser utilizados más adelante; también pueden ser locales si solo se van a utilizar en la pantalla actual.

	<pre>string esta1, esta2, esta3, esta4  integer ingreso_num [1 to 8, 1 to 8]</pre>	<p>Se puede establecer el tipo de variable de la siguiente manera:</p> <p>Dec Valores decimales</p> <p>Int Valores enteros</p> <p>dec{2}____[1 to 8, 1 to 8] Arreglo bidimensional con dos cifras decimales y admite valores del 1 al 8</p> <p>Si son varias variables del mismo tipo se separan con una coma.</p>
<b>Guardar un valor en un archivo</b>	<pre>esta1=sle_1.text</pre>	<p>Se guarda el texto ingresado en una “single line edit 1” en el archivo llamado “esta1”</p> <p>Se utilizó para guardar los nombres de los estados y mostrarlos al final.</p>
<b>Contar los valores ingresados</b>	<pre>if numes=3 then     if cont&gt;1 then         ingreso_num         [i,j]+=1     end if cont+=1 i=j end if</pre>	<p>Con un condicional “if” se verifica la cantidad de estados con los que se está trabajando “numes”, luego con el mismo condicional se confirma que se haya ingresado al menos un valor para comenzar a ubicar los datos que se ingresan en la matriz “ingreso_num”. La notación cont+=1 indica que se va sumando una unidad al contador cada vez que se digite un valor.</p>

✓ **Proceso de la información.**

<b>Instrucción</b>	<b>Sintaxis</b>	<b>Descripción</b>
<b>Contar los elementos de cada fila</b>	<pre> choose case I   case 1     conta=(ingreso_num [i,j]+conta)   case 2     contb=(ingreso_num [i,j]+contb)   case 3     contc=(ingreso_num [i,j]+contc) end choose </pre>	<p>El condicional “case” permite realizar una acción diferente para cada caso, se utilizó para contar los elementos de cada fila “i”, cada vez que se ingresaba un elemento en cada fila, se iban contabilizando para después poder ofrecer la probabilidad en cada fila. Se inicia con un “choose case i” y se cierra el condicional con un “end choose”</p>
<b>Cargar la información de una matriz en otra</b>	<pre> for i=1 to numes   for j=1 to numes     proa[i,j]=mostrar[i,j]   next next </pre>	<p>Con éste doble ciclo “for” se copia la información de la matriz “mostrar” a la matriz “proa”. La dimensión de la matriz es numes, se comienza el ciclo con un “for” y se cierra con un “next”.</p>
<b>Multiplicar dos matrices.</b>	<pre> for i=1 to numes   for j= 1 to numes     prob[i,j]=0     for k= 1 to numes       prob[i,j]=prob[i,j]+       proc[i,k]*proa[k,j]     next   next next </pre>	<p>Con este triple ciclo “for” encadenado se multiplica la matriz “proc” por la matriz “proa” y el resultado es guardado en la matriz “prob”, la dimensión de las tres matrices es “numes”. Se utilizo para obtener las matrices <math>P^n</math> de la predicción, siendo “P” la matriz de transición y “n” los periodos de tiempo para los que se va</p>

	next	a realizar dicha predicción.
<b>Calculo de cada probabilidad</b>	<pre> for i=1 to numes   for j=1 to numes     choose case i   case 1     if conta&gt;0 then       mostrar[i,j]=(ingreso _num[i,j]/conta)     else       mostrar[i,j]=ingreso _num[i,j]     end if   case 2     if contb&gt;0 then       mostrar[i,j]=(ingreso _num[i,j]/contb)     else       mostrar[i,j]=ingreso _num[i,j]     end if   end choose end for next next </pre>	<p>Con la combinación adecuada de dos ciclos “for”, un condicional “choose” y uno “if”, el programa calcula cada una de las probabilidades, dividiendo cada valor de la matriz “ingreso_num” por el total de valores ingresados en cada fila “conta” para la primera fila, “contb” para la segunda y así sucesivamente, dependiendo de la cantidad de estados y los datos que haya ingresado el usuario, toda esta información es almacenada en la matriz “mostrar” que es la que el sistema publica en la pantalla de matriz de transición.</p> <p>Cuando no hay datos en alguna casilla de la matriz y el contador sea cero, se utiliza un “else” que indica que si no se cumple la condición if, entonces realice otra acción y me permite asignar el valor sin dividirlo por el contador para evitar indeterminaciones y que el programa se detenga o se bloquee.</p>

✓ **Muestra de los resultados.**

<b>Instrucción</b>	<b>Sintaxis</b>	<b>Descripción</b>
<b>Mostrar en una caja de texto un valor Especifico</b>	<pre>st_1.text=string(mostrar[1,1]) st_2.text=string(mostrar[1,2]) st_3.text=string(mostrar[1,3])</pre>	<p>Con esta instrucción se muestra en la estatic text “st_1” la información de la matriz “mostrar” ubicado en la primera fila y primera columna, de igual manera con la “st_2” y la “st_3”</p> <p>Se utilizó para mostrar en la matriz de transición cada elemento almacenado después de haber realizado los cálculos correspondientes.</p>
<b>Mostrar u ocultar objetos</b>	<pre>st_3.visible=false st_5.visible=false</pre>	<p>Nos permite hacer visible algún objeto utilizando “=true” o invisible ingresando “=false” como se observa en el ejemplo, en este caso se están haciendo invisibles las estatic text “st” número tres y cinco porque según la cantidad de estados, solo se van a observar los elementos correspondientes para esta cantidad.</p>
<b>Mostrar en un botón de texto</b>	<pre>st_66.text=string(esta1)</pre>	<p>Con esta instrucción se muestra en la “estatictext_66” el nombre del primer estado que se había almacenado en el archivo “esta1”.</p> <p>Este comando se utilizó para mostrar la matriz de transición con sus respectivos estados y de igual manera con la matriz de predicción.</p>

<p><b>Mensajes emergentes</b></p>	<pre>messagebox('ERROR', 'Debe ingresar una letra valida', exclamation!) sle_1.setFocus( )</pre>	<p>Cuando el usuario ingresa un valor incorrecto o le hacen falta espacios por diligenciar, emerge un cuadro de mensaje “messagebox”, el cual le indica que debe hacer para que el programa avance a la siguiente ventana; el orden en el que deben ir las instrucciones debe ser el siguiente y deben estar separados por comas y encerrados en comillas simples como se observa en el ejemplo.</p> <p>La primera instrucción es el titulo del mensaje, la segunda es el mensaje que se quiere mostrar y la tercera indica el tipo de notificación que va a emerger.</p>
-----------------------------------	--	---

## 4. EVALUACIÓN DE LA HERRAMIENTA

La deserción estudiantil en la Universidad Pedagógica Nacional (UPN) ha sido objeto de estudio desde la década de los ochenta, para evaluar la herramienta que se diseñó, a continuación se presenta un pequeño estudio de la deserción en la Universidad.

Los aspectos a estudiar son de tipo cualitativo (razones por la cual no continua sus estudios en la UPN) y cuantitativo (Porcentaje de deserción por periodo académico); En sus 50 años, la Universidad Pedagógica Nacional realizó el estudio *“La deserción estudiantil: reto investigativo y estratégico asumido de forma integral por la UPN”* en el cual definen la deserción como:

*“El hecho de que un grupo de estudiantes matriculados no siga la trayectoria normal del programa académico, bien sea por retirarse de ella o por demorar más tiempo del previsto en finalizarla, por repetir cursos o por retiros temporales. El abandono o interrupción pueden ser voluntarios o forzados. También puede presentarse cambio de carrera dentro de la misma institución o cambio de institución donde puede continuar con la misma carrera o con otra”.*

El centro de orientación y acompañamiento a estudiantes (COAE) ha asumido el reto de hacer seguimiento y establecer algunas estrategias para evitar los índices de deserción en la UPN. En sus diferentes indagaciones (las cuales realiza personalmente para evitar información errada) para identificar las causas de la deserción, el COAE determinó que, las principales causas o

motivos por los cuales los estudiantes no culminan exitosamente sus estudios son las siguientes:

#### FAMILIARES

- ✓ Dificultades económicas
- ✓ Problemas en las relaciones familiares
- ✓ Enfermedad o deceso de un familiar
- ✓ Embarazo o atención a hijos

#### ACADÉMICAS

- ✓ Bajo Rendimiento académico
- ✓ Insatisfacción con los profesores
- ✓ La UPN no llenó sus expectativas
- ✓ Obtuvo cupo en otra universidad

#### SOCIAL CULTURAL

- ✓ La carrera no llenó sus expectativas
- ✓ Trabajo
- ✓ Cambio de domicilio
- ✓ Problemas de salud
- ✓ Dificultades de adaptación a la universidad

A partir del 2003 el Ministerio de Educación Nacional (MEN) implementa el programa AKADEMIA, el cual define que el índice de deserción se debe calcular de la siguiente manera: Se comparan los estudiantes matriculados de dos semestres consecutivos de cada uno de los programas de formación, se le restan los ingresos a primer semestre y los graduados y se le suman las

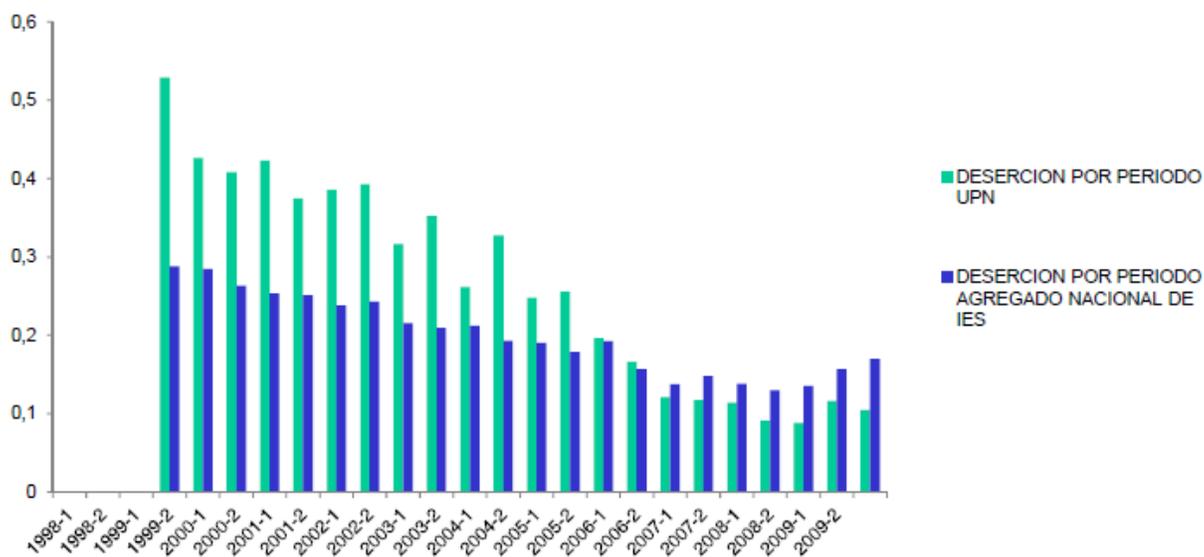
cancelaciones totales (o aplazamiento de semestre) del último período, de esta manera se identifican los desertores por periodo.

Se denominan desertores parciales a los que después de haber interrumpido su proceso académico en uno o más semestres, reingresan al programa y continúan sus estudios normalmente, y desertores a los que después de determinado momento no realizan registro de calificaciones.

A continuación se muestra el cuadro histórico de la deserción en la UPN (Figura 29), el cual fue presentado en la audiencia pública de rendición de cuentas a la ciudadanía, vigencias 2009 y i semestre de 2010 por el rector Oscar Armando Ibarra Russi.



### MARCO HISTÓRICO – CONTEXTUALIZACIÓN DE LA POBLACIÓN ESTUDIANTIL DISMINUCIÓN DE LOS ÍNDICES DE DESERCIÓN



**Figura 29.** Tomada de la Audiencia pública de rendición de cuentas a la ciudadanía Universidad Pedagógica Nacional.

Con base en estos datos se realizó la evaluación del software CMTD10, los datos a ingresar se muestran en la siguiente tabla (Tabla 3.).

**Tabla 3.** *Datos equivalentes al diagrama de la figura 29.*

<b>PERIODO</b>	<b>DESERCIÓN</b>	<b>PERIODO</b>	<b>DESERCIÓN</b>
		<b>1999-2</b>	53%
<b>2000-1</b>	43%	<b>2000-2</b>	41%
<b>2001-1</b>	43%	<b>2001-2</b>	37%
<b>2002-1</b>	38%	<b>2002-2</b>	39%
<b>2003-1</b>	32%	<b>2003-2</b>	36%
<b>2004-1</b>	26%	<b>2004-2</b>	32%
<b>2005-1</b>	25%	<b>2005-2</b>	26%
<b>2006-1</b>	20%	<b>2006-2</b>	17%
<b>2007-1</b>	13%	<b>2007-2</b>	13%
<b>2008-1</b>	12%	<b>2008-2</b>	9%
<b>2009-1</b>	8%	<b>2009-2</b>	12%
<b>2010-1</b>	11%		

Según el Ministerio de Educación Nacional (MEN), la deserción intersemestral en la educación superior general en Colombia es del 10%. Esta es la meta que se ha establecido el COAE y este es el punto de referencia para el análisis que se realizó con ayuda del software CMTD10.

## 4.1. ANÁLISIS.

A continuación se presenta el análisis realizado con las cadenas de Márkov y con ayuda del software CMTD10, el sistema realiza transiciones en sus estados cuantitativos (porcentaje de deserción) y el objetivo es determinar la probabilidad que el estado sea menor o igual al 10% en el siguiente periodo de tiempo (2010-2) y realizar una predicción a futuro. Se describe cada uno de los elementos a tener en cuenta.

### **Estado:**

Se cuenta con un estado cuantitativo y por la naturaleza de la situación se establecen 5 estados para el sistema en forma de intervalo semi-abierto de la siguiente manera:

**Tabla 4.** *Estados definidos en cada intervalo.*

<b>Número</b>	<b>Estado</b>
<b>1</b>	<b>[0% - 10%)</b>
<b>2</b>	<b>[10% - 20%)</b>
<b>3</b>	<b>[20% - 30%)</b>
<b>4</b>	<b>[30% - 40%)</b>
<b>5</b>	<b>[40% - 53%]<sup>3</sup></b>

### **Número de estados:**

Como el análisis se va a realizar entre los periodos de tiempo 1999-2 al 2010-1, se cuenta con 22 estados, el sistema realiza una transición (cambio de estado) cada seis meses.

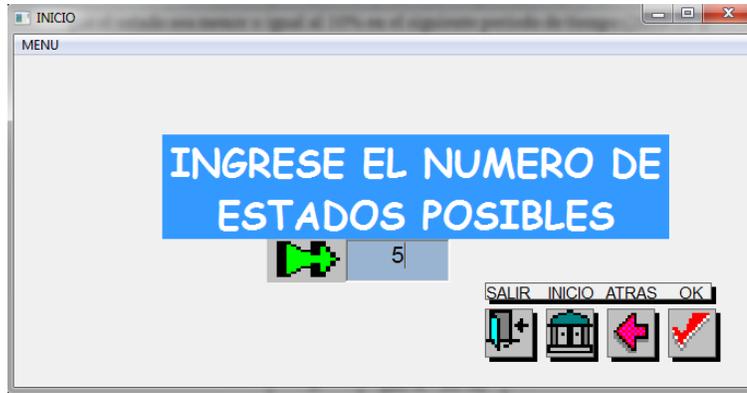
---

<sup>3</sup> El último intervalo se realiza hasta 53% por ser un dato atípico.

## INGRESO DE DATOS

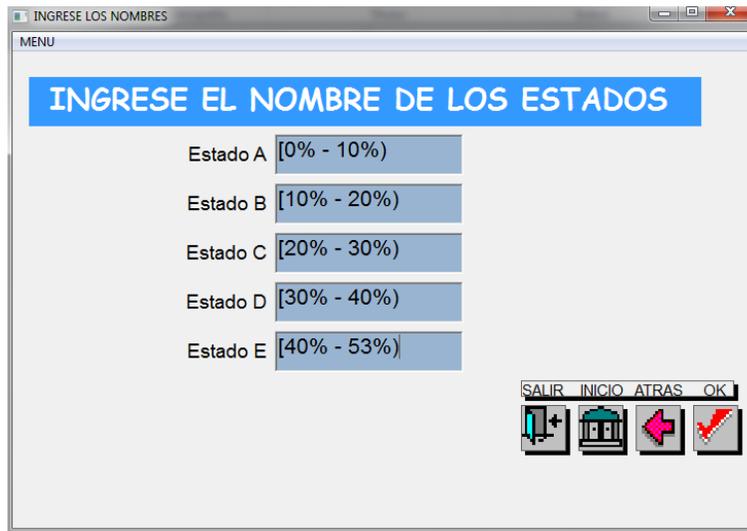
A continuación se presenta en el orden como se ingresaron los datos y al final la matriz de transición y una predicción a futuro.

### Ventana 1



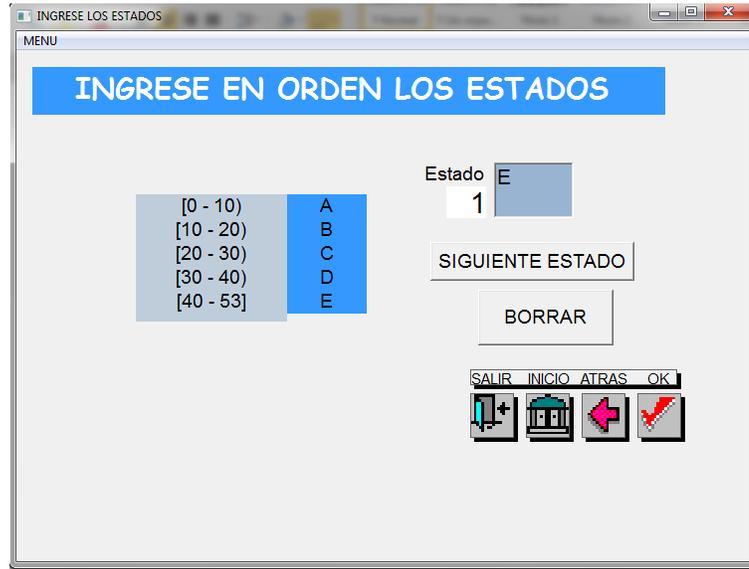
**Figura 30.** Ventana donde se ingresa el número de estados posibles, en este caso cinco.

### Ventana 2



**Figura 31.** Ventana donde se ingresa el nombre de cada estado, en este caso los intervalos a los cuales pertenece cada porcentaje de deserción.

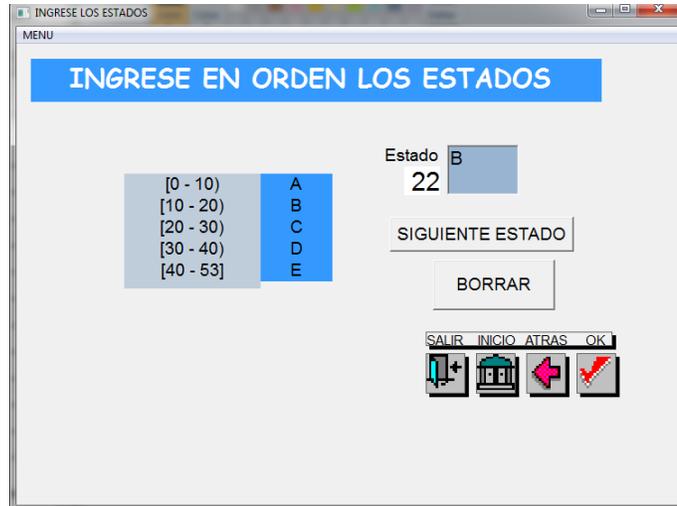
### Ventana 3



**Figura 32.** *Ventana para ingresar en orden los estados correspondientes a cada intervalo.*

Como en el primer periodo de tiempo (1999-2) la deserción fue del 53%, y este valor pertenece al intervalo [40 - 50], entonces se ingresa la letra E y así sucesivamente con los demás estados.

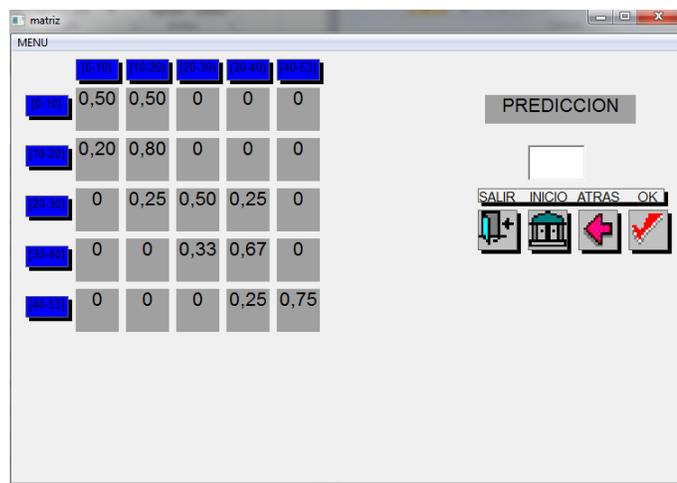
## Ventana 4



**Figura 33.** Ingreso del último estado.

El último estado del sistema es el intervalo B, puesto que en el 2010-1 se presentó una deserción del 11%.

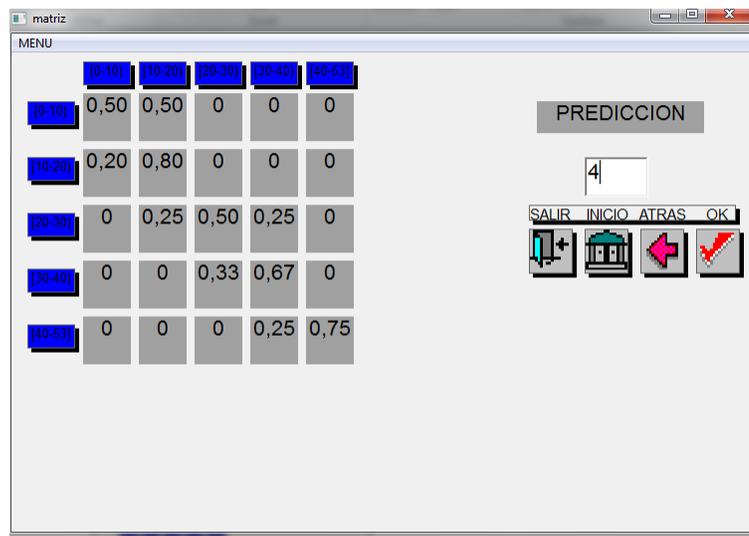
## Ventana 5



**Figura 34.** Matriz de transición.

Emerge la matriz de transición, de la cual se pueden deducir todos los casos posibles de acuerdo al estado actual, se puede observar por ejemplo que como en el último periodo de tiempo la deserción fue del 11%, el cual pertenece al intervalo [10-20), entonces existe una posibilidad del 80% que al siguiente semestre se presente una deserción entre 10% y 20%. También se puede observar que existe una posibilidad del 20% que en el siguiente periodo exista una deserción menor al 10%.

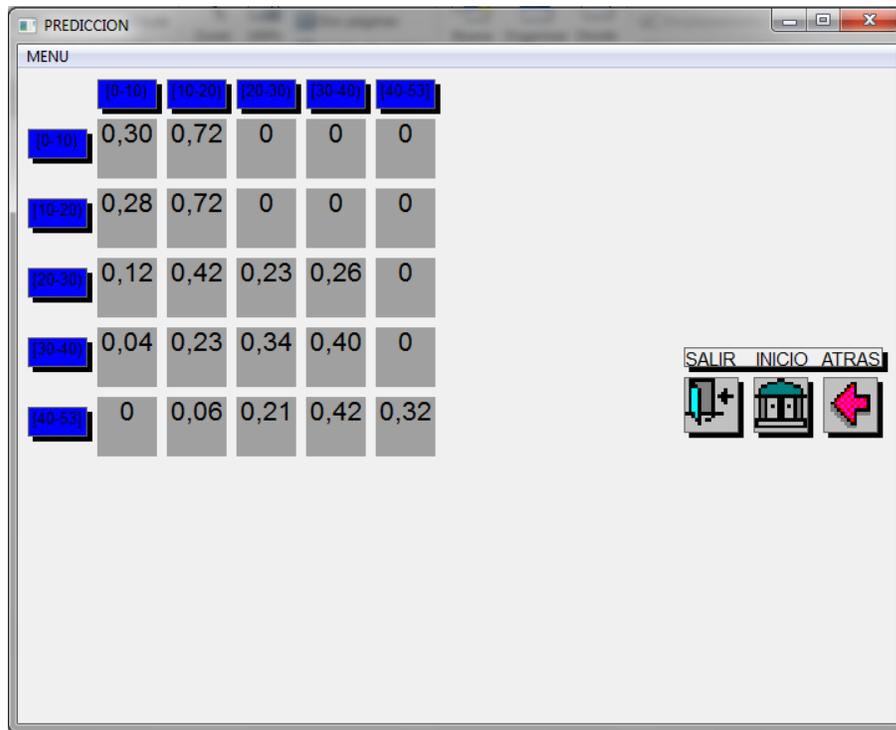
### Ventana 6



**Figura 35.** Ingreso periodos de tiempo para la predicción.

Se realiza una predicción a 4 periodos de tiempo, es decir la deserción para el 2012-1

## Ventana 7



**Figura 36.** Ventana matriz de predicción.

Según lo observado en la matriz de predicción (Ver Figura 36), se llega a las siguientes conclusiones puntuales:

1. Si el periodo anterior (2011-2) tuvo un porcentaje de deserción entre 0% y 10%, entonces existe un 30% de probabilidad que en el siguiente semestre (2012-1), haya una deserción en el intervalo [0% - 10%), y una probabilidad del 72% que en el siguiente semestre haya una deserción en el intervalo [10%-20), esto se puede observar en la primera fila de la matriz predicción, teniendo en cuenta que las probabilidades en los demás estados es nula.

2. Asumiendo que la deserción en el periodo 2011-2 tuvo un porcentaje de deserción entre el 10% y el 20%, entonces existe una probabilidad del 28% que la deserción este en el intervalo [0% - 10%) y una probabilidad del 72% que se encuentre en el intervalo [10% - 20%). Esta información se analiza al observar la segunda fila de la matriz predicción.

## 5. CONCLUSIONES

Programar una herramienta que realice algunos cálculos específicos, es un reto que asumí desde la propuesta en el anteproyecto de mi trabajo de grado y que día a día se fue consolidando, buscar un lenguaje de programación que se ajuste a las necesidades, estudiar su sintaxis y entorno de trabajo. Con todo esto no fue fácil poner en marcha el proyecto, pero después de estar diseñando cada ventana y la función que iba a cumplir, la convicción de poder sacar adelante este trabajo de grado se fue fortaleciendo, se siente una gran satisfacción poder poner en juego todo lo aprendido a lo largo de la Licenciatura en Matemáticas, no solo en conocimiento propio del área o de herramientas tecnológicas, sino también en diferentes aspectos que hoy me permiten culminar este proyecto.

Se entrega un documento escrito con mi trabajo de grado, en el cual se encuentra la descripción del proceso de elaboración, el código de programación, el manual de instalación e instrucciones de uso de la herramienta tecnológica y una aplicación del software en una situación de la Universidad Pedagógica Nacional, con esto concluyo satisfactoriamente los objetivos propuestos en el anteproyecto.

El material que se consigna en la biblioteca de la Universidad pedagógica es de libre uso y consulta, cedo todos los derechos intelectuales de mi trabajo para que estudiantes de diferentes carreras o universidades puedan tener acceso y utilizarlo cuando lo necesiten.

## BIBLIOGRAFÍA

[1] Guardiet. F., Sall. J., Suñé. A., (2002) *Métodos cuantitativos de organización industrial II*, Capítulo 3 Cadenas de Márkov, Ediciones UPC, 2002. En línea <http://es.scribd.com/doc/92621758/Cadenas-de-Markov>

[2] Guyon. X. (1999) *Métodos numéricos por cadenas de Márkov*. Université Paris 1Y Universidad de Los Andes, Mérida, Venezuela [http://samos.univ-paris1.fr/archives/ftp/cours/guyon/Meth\\_Num-Ch.Markov-Espanol.pdf](http://samos.univ-paris1.fr/archives/ftp/cours/guyon/Meth_Num-Ch.Markov-Espanol.pdf)

[3] Mesa. G. (2007) *Cadenas de Márkov, una sencilla aplicación*. Revista memorias. Vol.5 No 9 en línea <http://www.revistamemorias.com/articulos9/cadenasdemarkov.pdf>

[4] Riaño., G. (sf) *Cadenas de Márkov*, Universidad de los Andes, en línea <http://copa.uniandes.edu.co/software/Markov/markov.html>

[5] Garcia., O., Nieto., F., (2004) *Software educativo: Cadenas de Márkov en tiempo discreto*. Tesis de grado para recibir el título de Ingeniero industrial en la Pontificia Universidad Javeriana.

[6] Aznar. E. (2007) *Andrei Andreyevich Márkov*. Universidad de Granada, facultad de ciencias, departamento de algebra. En línea <http://www.ugr.es/~eaznar/markov.htm>

[7] Gomez. W. (2000) *powerbuilder, manual del programados*. SYBASE, en línea <http://es.scribd.com/doc/136839733/58470932-Manual-Power-Builder-pdf>

[8] Hernández, M. & Hernández, Carmenza. (1999). *Factores determinantes de la deserción estudiantil en la Universidad Pedagógica Nacional. Comparación entre desertores y no desertores*. Universidad Pedagógica Nacional, Vicerrectoría Académica, Sub proyecto 3.1.1.3 Inscripción, Admisión y Seguimiento a estudiantes, Bogotá.

[9] UPN (2006) *La deserción estudiantil en la UPN*. Universidad Pedagógica Nacional, Vicerrectoría Académica, Centro de orientación y acompañamiento a estudiantes (COAE), En inea: [http://www.mineducacion.gov.co/1621/articles-85600\\_Archivo\\_pdf2.pdf](http://www.mineducacion.gov.co/1621/articles-85600_Archivo_pdf2.pdf)

[10] UPN (2010) *Audiencia pública de rendición de cuentas a la ciudadanía-vigencias 2009 y I semestre de 2010*, Universidad Pedagógica Nacional, Rectoría, Bogotá.