

LA **PROGRAMACIÓN** NO TIENE LÍMITES, CON UN POCO DE **FE**
Developeralta



Markov

Versión 1.0

Manual Técnico
Software de aplicación

Crear una cadena de Markov con los datos de un archivo

ÍNDICE

1. DESCRIPCIÓN DEL PROBLEMA	2
1.1 REQUERIMIENTOS BÁSICOS.....	3
RESTRICCIONES BÁSICAS	4
2. PROPUESTA DE SOLUCIÓN	4
2.1 PLANTEAMIENTO DE LA SOLUCIÓN.....	4
2.2 DEFINICIÓN DE ENTRADAS Y SALIDAS	5
2.3 LIMITANTES	5
3. REPRESENTACIÓN DE LA SOLUCIÓN (Pseudocódigo)	5
4. PRUEBAS DE SOFTWARE	10
4.1 PRUEBAS UNITARIAS	10
4.2 PRUEBAS DE INTREGACIÓN.....	10
4.3 PLAN DE PRUEBAS	10
5. REQUERIMIENTOS MINIMOS DE HARDWARE Y SOFTWARE PARA EL DESARROLLO Y MANTENIMIENTO	12
5.1 HARDWARE	12
5.2 SOFTWARE	12
6. IMPLEMENTACIÓN (Codificación Comentada)	12
7. RECOMENDACIONES	16
8. BIBLIOGRAFÍA	16

1. DESCRIPCIÓN DEL PROBLEMA

Es común ver en el espacio público un porcentaje de personas que fuman y otras que no lo hacen. Cabe la posibilidad de que aquellos que no fuman más adelante sí lo hagan y los que actualmente fuman dejen el cigarrillo. Una sucesión de observaciones con determinado número de resultados, cada uno de los cuales tiene una probabilidad, depende sólo del resultado de la etapa inmediatamente anterior. Este proceso en el que intervienen variables aleatorias indexadas en el tiempo se denomina cadenas de Markov, haciendo honor al matemático ruso Andrei Andreyevich Markov, quien creó este método. De esta forma la probabilidad de ocurrencia de un evento depende del evento anterior.

Por lo tanto se requiere un software capaz de obtener de una base de datos las siguientes probabilidades:

- Número
 - $x < 0$
 - $0 \leq x \leq 1$
 - $x \geq 1$
- Cadena Alfanúmerica
 - A...Z
 - Caracteres especiales.
 - A...Z y caracteres especiales.

Y con los resultados anteriores construir la cadena de Markov correspondiente y un reporte con los siguientes puntos:

- Total de números encontrados.
- Número total de cadenas alfanúmericas encontradas.
- Total de números menores a cero.
- Total de números que se encuentren el siguiente intervalo $0 \leq x \leq 1$.
- Total de números mayores a uno.
- Total de cadenas con solo letras.
- Total de cadenas con solo caracteres especiales.
- Total de cadenas con letras y caracteres especiales.

1.1 REQUERIMIENTOS BÁSICOS

- ✓ El usuario podrá introducir el nombre del archivo y se mostrará en pantalla todos los puntos anteriores. Por ejemplo, si se introduce el nombre de archivo **Datos.csv**, el programa deberá mostrar la cadena de Markov correspondiente y generar el siguiente reporte:

Reporte Markov 1.0

25 de octubre del 2019

	<i>Total</i>
<i>Números</i>	<i>2</i>
<i>$x < 0$</i>	<i>0</i>
<i>$0 \leq x \leq 1$</i>	<i>2</i>
<i>$x > 0$</i>	<i>0</i>
<i>Cadenas</i>	<i>1</i>
<i>Cadenas A... Z</i>	<i>1</i>
<i>Especiales</i>	<i>0</i>
<i>Otros</i>	<i>1</i>
<i>Probabilidades</i>	
<i>Números</i>	<i>0.5</i>
<i>$x < 0$</i>	<i>0</i>
<i>$0 \leq x \leq 1$</i>	<i>1</i>
<i>$x > 0$</i>	<i>0</i>
<i>Cadenas</i>	<i>0.25</i>
<i>Cadenas A... Z</i>	<i>1</i>
<i>Especiales</i>	<i>0</i>

Otros

0.25

RESTRICCIONES BÁSICAS

- ✓ Los caracteres no válidos para el nombre de los archivos serán:
 - /
 - *
 - ?
 - “
 - <
 - >
 - |
- ✓ El nombre del archivo deber incluir la ruta absoluta o relativa antes de su nombre físico. Por ejemplo, */Users/developeralta/Datos.csv* o *BaseDatos.csv*.

2. PROPUESTA DE SOLUCIÓN

2.1 PLANTEAMIENTO DE LA SOLUCIÓN

El algoritmo es el siguiente:

Para la solución se emplea un Autómata Finito Determinista.

1. Leer el nombre del archivo.
2. Tratamiento de Datos
 - a. Aplicar los pasos b) a k) mientras no sea el fin del archivo.
 - b. Leer el registro.
 - c. Concatenar el carácter coma al final del registro.
 - d. Para cada item del registro aplicar los pasos e) a j).
 - e. Obtener el estado siguiente del AFD correspondiente al item actual.
 - f. Si item está en estado inválido:
 - i. Sumar uno al contador cadena.
 - ii. Aplicar el paso iii) mientras item sea diferente a una coma o un espacio.
 - iii. Tomar el siguiente item. Volver al paso i).
 - iv. Sumar uno al contador de otro.
 - g. Si item está en estado de aceptación de número:
 - i. Convertir cadena a número.
 - ii. Sumar uno al contador de número.
 - iii. Si número es menor que cero, entonces sumar uno al contador menor cero.

- iv. Si número es mayor que cero y menor que uno, entonces sumar uno al contador intervalo.
 - v. Si número es mayor que cero, entonces sumar uno al contador mayor cero.
 - h. Si item está en estado de aceptación de letra, entonces:
 - i. Sumar uno al contador cadena.
 - ii. Sumar uno al contador de letra.
 - i. Si item está en estado de aceptación de cadena especial, entonces:
 - i. Sumar uno al contador cadena.
 - ii. Sumar uno al contador especial.
 - j. Volver al paso d).
 - k. Volver al paso a).
 - l. Escribir el total de números y cadenas en el reporte.
3. Mostrar reporte en la pantalla.

2.2 DEFINICIÓN DE ENTRADAS Y SALIDAS

- ✓ Entradas
 - Nombre del Archivo : Cadena
 - Opcion : Cadena
- ✓ Salidas
 - Reporte : Archivo de Texto

2.3 LIMITANTES

- El nombre del archivo es sensible a mayúsculas.

3. REPRESENTACIÓN DE LA SOLUCIÓN (Pseudocódigo)

```

PROGRAMA Markov 1.0
/*
AUTOR: Developeralta.
FECHA: 09 de Octubre del 2019.
PROPÓSITO: Por lo tanto se requiere un software capaz de obtener de una base de datos las siguientes
probabilidades:
    Número
    x<0
    0≤x≤1
    x≥1
    Cadena Alfanúmerica
    A...Z
    Caracteres especiales.
    A...Z y caracteres especiales.
Y con los resultados anteriores construir la cadena de Markov correspondiente y un reporte con los
siguientes puntos:
    Total de números encontrados.
    Número total de cadenas alfanúmericas encontradas.
    Total de números menores a cero.
    Total de números que se encuentren el siguiente intervalo 0≤x≤1.
    Total de números mayores a uno.
    Total de cadenas con solo letras.
    Total de cadenas con solo caracteres especiales.

```

```

        Total de cadenas con letras y caracteres especiales.
SUPOSICIONES: Ninguna.
LIMITANTES:
• El nombre del archivo es sensible a mayúsculas.
*/
LIBRERÍAS
| Pantalla
| ArchivosSecuenciales
CONSTANTES
| NReporte="ReporteMarkov-"
| Extension=".txt"
| Numero=1
| Cadena=2
| MayorCero=3
| MenorCero=4
| Intervalo=5
| Letra=6
| Especial=7
| Otro=8
| MaxRenglonesNR=7
| MaxColumnasNR=6
| MatrizAdyacenciaNR : ARREGLO [1..MaxRenglonesNR,1..MaxColumnasNR] DE
ENTERO=((3,2,0,0,6,7),(3,0,0,0,0,0),(3,0,4,1,0,0),(5,0,0,0,0,0),(5,0,0,1,0,0),(0,0,0,1,6,0),(0,0,
0,1,0,7))
TIPOS DE DATOS
| AContadores : ARREGLO [1..8] DE ENTERO
| MatrizMarkov : ARREGLO [1..8,1..9] DE REAL
VARIABLES
| /* VARIABLES DE ENTRADA */
| AOrigen,Opcion : CADENA
| /* VARIABLES DE SALIDA */
| Reporte : TArchivoTxt
| /* VARIABLES AUXILIARES */
| ReporteNF : CADENA
| Contadores : AContadores;
INICIO
| REPITE
| | COMIENZA
| | | MostrarEncabezado("Markov 1.0")
| | | ESCRIBE "Nombre del Archivo CSV: "
| | | LEE(AOrigen)
| | | SI(VerificarArchivoTexto(AOrigen+".csv")) ENTONCES
| | | | COMIENZA
| | | | | ReporteNF<-NReporte+AOrigen+Extension
| | | | | SI(CrearArchivoTexto(ReporteNF)) ENTONCES
| | | | | COMIENZA
| | | | | | ASIGNA(Reporte,ReporteNF)
| | | | | | ABRE Reporte PARA ESCRITURA
| | | | | | ESCRIBE "Reporte Markov 1.0" EN Reporte
| | | | | | ESCRIBE FechaSistema EN Reporte
| | | | | | ESCRIBE "-----" EN Reporte
| | | | | | ESCRIBE "Dato"
| | | | | | TratamientoCSV(AOrigen+".csv",Contadores)
| | | | | | ESCRIBE "Numeros: "+Cadena(Contadores[Numero]) EN REPORTE
| | | | | | ESCRIBE "---> x<0: "+(Contadores[MenorCero]) EN REPORTE
| | | | | | ESCRIBE "---> 0<x<1: "+(Contadores[Intervalo]) EN REPORTE
| | | | | | ESCRIBE "---> x>0: "+(Contadores[MayorCero]) EN REPORTE
| | | | | | ESCRIBE "Cadenas: "+Cadena(Contadores[Cadena]) EN REPORTE
| | | | | | ESCRIBE "--->A..Z: "+(Contadores[Letra]) EN REPORTE
| | | | | | ESCRIBE "--->Especial: "+(Contadores[Especial]) EN REPORTE
| | | | | | ESCRIBE "Otros: "+(Contadores[Otro]) EN REPORTE
| | | | | | CalcularProbabilidades(Reporte,Contadores)
| | | | | | CERRAR(Reporte)
| | | | | | MostrarArchivoTexto(ReporteNF)
| | | | | TERMINA
| | | | SINO
| | | | | ESCRIBE "ERROR 02: No se pudo crear el reporte :("
| | | | | FIN_SI
| | | | TERMINA
| | | SINO

```

```

| | | | ESCRIBE "ERROR 01: No existe el archivo csv :("
| | | | FIN_SI
| | | | ESCRIBE "¿Quiere tratar otro archivo CSV? (SI)"
| | | | LEE(Opcion)
| | | | Opcion<-MAYÚSCULAS(Opcion)
| | | | Enter("");
| | | | TERMINA
| | | | HASTA(Opcion<>"SI")
FIN
/* PROCEDIMIENTOS Y FUNCIONES */
PROCEDIMIENTO TratamientoDatos(VAL CopiaNF : CADENA,REF Contadores : AContadores)
/*
UTILIDAD: Tratar la información del archivo verificando si son números o letras.
PRECONDICIÓN: 'CopiaNF' existe.
POSCONDICIÓN: 'Contadores' tiene el total de números y cadenas encontrados en el archivo.
*/
VARIABLES
| Archivo : TArchivoTxt
| Registro,CadenaN : CADENA
| Item,Estado,Longitud,EstadoA : ENTERO
| Dato : REAL
COMIENZA
| PARA(Item<-1,8)
| | Contadores[Item]<-0
| FIN_PARA
| ASIGNA(Archivo,CopiaNF)
| ABRE Archivo PARA LECTURA
| MIENTRAS(NO(FDA(Archivo)))
| | COMIENZA
| | | LEE Registro DE Archivo
| | | Registro<-Registro+", "
| | | Longitud<-LONGITUD(Registro)
| | | Item<-1
| | | Estado<-1
| | | CadenaN<-""
| | | MIENTRAS(Item<=Longitud)
| | | | COMIENZA
| | | | | EstadoA<-Estado
| | | | | EstadoActualAFD(Estado,Registro[Item])
| | | | | CASO(Estado)
| | | | | 0: //Estado inválido
| | | | | COMIENZA
| | | | | | Contadores[Otro]<-Contadores[Otro]+1
| | | | | | REPITE
| | | | | | | Item<-Item+1
| | | | | | HASTA(Registro[Item]=',' ) O (Registro[Item]=' ' ) O (Registro[Item]='Tab')
| | | | | | Estado<-1
| | | | | | ESCRIBE "ERROR"
| | | | | | CadenaN<-""
| | | | | | TERMINA
| | | | | 1: //Estado Inicial
| | | | | COMIENZA
| | | | | | SI (LONGITUD (CadenaN)>0) ENTONCES
| | | | | | | COMIENZA
| | | | | | | | ESCRIBE CadenaN
| | | | | | | | SI (EstadoA=3) O (EstadoA=5) ENTONCES
| | | | | | | | COMIENZA
| | | | | | | | | Contadores[Numero]<-Contadores[Numero]+1
| | | | | | | | | Dato<-VALOR (CadenaN)
| | | | | | | | | SI (Dato<0) ENTONCES
| | | | | | | | | Contadores[MenorCero]<-Contadores[MenorCero]+1
| | | | | | | | | SINO
| | | | | | | | | COMIENZA
| | | | | | | | | | SI (Dato<1) ENTONCES
| | | | | | | | | | | Contadores[Intervalo]<-Contadores[Intervalo]+1
| | | | | | | | | | | SINO
| | | | | | | | | | | Contadores[MayorCero]<-Contadores[MayorCero]+1
| | | | | | | | | | | FIN_SI
| | | | | | | | | | | TERMINA
| | | | | | | | | | | FIN_SI

```



```
| | | | | | | | | TERMINA
| | | | | | | | | SINO
| | | | | | | | | COMIENZA
| | | | | | | | | Contadores[Cadena]<-Contadores[Cadena]+1
| | | | | | | | | SI (EstadoA=6) ENTONCES
| | | | | | | | | Contadores[Letra]<-Contadores[Letra]+1
| | | | | | | | | SINO
| | | | | | | | | Contadores[Especial]<-Contadores[Especial]+1
| | | | | | | | | FIN_SI
| | | | | | | | | TERMINA
| | | | | | | | | FIN_SI
| | | | | | | | | CadenaN<-""
| | | | | | | | | TERMINA
| | | | | | | | | FIN_SI
| | | | | | | | | TERMINA
| | | | | | | | | OTRO: //Estado inválido
| | | | | | | | | CadenaN<-CadenaN+Registro[Item]
| | | | | | | | | FIN_CASO
| | | | | | | | | Item<-Item+1
| | | | | | | | | TERMINA
| | | | | | | | | FIN_MIENTRAS
| | | | | | | | | TERMINA
| | | | | | | | | FIN_MIENTRAS
| | | | | | | | | CERRAR(Archivo)
| | | | | | | | | TERMINA
```

```
PROCEDIMIENTO EstadoActualAFD(REF Nodo : ENTERO,VAL CopiaSimbolo : CHARACTER)
/*
UTILIDAD: Encontrar el nodo al que llega el simbolo.
PRECONDICIÓN: 'Nodo' tiene el estado actual y 'CopiaSimbolo' el caracter siguiente.
POSCONDICIÓN: Se devuelve el valor del nodo en el que se encuentra 'CopiaSimbolo'.
*/
```

```
COMIENZA
| SI (CopiaSimbolo=Digitos) ENTONCES
| | Nodo<-MatrizAdyacenciaNR[Nodo,1]
| SINO
| | COMIENZA
| | | SI (CopiaSimbolo=Letras) ENTONCES
| | | | Nodo<-MatrizAdyacenciaNR[Nodo,5]
| | | SINO
| | | | COMIENZA
| | | | | CASO (CopiaSimbolo)
| | | | | '-' : /*Verificar si símbolo pertenece al caracter '-'.*/
| | | | |         Nodo<-MatrizAdyacenciaNR[Nodo,2]
| | | | | '.' : /*Verificar si símbolo es el caracter '.'.*/
| | | | |         Nodo<-MatrizAdyacenciaNR[Nodo,3]
| | | | | ',' : /*Verificar si símbolo es el caracter ','.*/
| | | | |         Nodo<-MatrizAdyacenciaNR[Nodo,4]
| | | | | ' ' : /*Verificar si símbolo es el caracter ' '.*/
| | | | |         Nodo<-MatrizAdyacenciaNR[Nodo,4]
| | | | | 'Tab' : /*Verificar si símbolo es el caracter 'Tab'.*/
| | | | |         Nodo<-MatrizAdyacenciaNR[Nodo,4]
| | | | | OTRO : /*Verifica si símbolo no pertenece al alfabeto*/
| | | | |         Nodo<-MatrizAdyacenciaNR[Nodo,6]
| | | FIN_CASO
| | TERMINA
| FIN_SI
| TERMINA
FIN_SI
TERMINA
```

```
PROCEDIMIENTO CalcularProbabilidades(REF Reporte : TArchivoTxt,VAL CContadores : AContadores)
/*
UTILIDAD: Calcular las probabilidades de los 'CContadores' como valores.
PRECONDICIÓN: 'CContadores' tiene el total de números y cadenas y 'Reporte' existe y está abierto.
POSCONDICIÓN: Las probabilidades se almacenan en el archivo 'Reporte'.
*/
VARIABLES
| Total : ENTERO
| Probabilidad : CADENA
```

```

COMIENZA
| Total<-CContadores[Numero]+CContadores[Cadena]+CContadores[Otro]
| ESCRIBE "Probabilidades" EN Reporte
| ESCRIBE "-----" EN Reporte
| Probabilidad<-CADENA(CContadores[Numero]/Total)
| ESCRIBE "Número: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[MenorCero]/CContadores[Numero])
| ESCRIBE "----> x<0: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[Intervalo]/CContadores[Numero])
| ESCRIBE "----> 0<x<1: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[MayorCero]/CContadores[Numero])
| ESCRIBE "----> x>1: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[Cadena]/Total)
| ESCRIBE "Cadena: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[Letra]/CContadores[Cadena])
| ESCRIBE "---->A...Z: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[Especial]/CContadores[Cadena])
| ESCRIBE "---->Especial: "+Probabilidad EN Reporte
| Probabilidad<-CADENA(CContadores[Otro]/Total)
| ESCRIBE "Otros: "+Probabilidad EN Reporte
TERMINA

```

4. PRUEBAS DE SOFTWARE

4.1 PRUEBAS UNITARIAS

El archivo con extensión PDF contiene los experimentos de cada módulo. Ctrl + Clic para visualizarlo.

[../Pruebas de Software/Control de Pruebas Unitarias-Markov 1.0.pdf](http://.../Pruebas de Software/Control de Pruebas Unitarias-Markov 1.0.pdf)

4.2 PRUEBAS DE INTEGRACIÓN

El archivo con extensión PDF contiene la prueba del programa principal. Ctrl + Clic para visualizarlo.

[../Pruebas de Software/Pruebas de Integración-Markov 1.0.pdf](http://.../Pruebas de Software/Pruebas de Integración-Markov 1.0.pdf)

4.3 PLAN DE PRUEBAS

No. de Prueba	Juegos de Datos		Resultados Esperados	Juego de Datos	Observaciones
	Archivo CSV	Variables		Opcion=NO	
1	22,45,78,89		22	SI	
	1,2,3,4,5,6,7		45		
			78		
			89		
			1		

MARKOV VERSIÓN 1.0

			2		
			3		
			4		
			5		
			6		
			7		
2	12,85x,67 23 78,89		12 ERROR 67 23 78 89	SI	

5. REQUERIMIENTOS MINIMOS DE HARDWARE Y SOFTWARE PARA EL DESARROLLO Y MANTENIMIENTO

5.1 HARDWARE

- ✓ Procesador Intel® Pentium® 233 MHz o superior.
- ✓ 64 MB RAM (se recomiendan 128 MB).
- ✓ 400 MB de espacio disponible en disco duro.
- ✓ Monitor con resolución 800 x 600 p o superior.
- ✓ Mouse.

5.2 SOFTWARE

- ✓ macOS Sierra Versión 10.12.6 o Superior.
- ✓ Geany Versión 1.34.1.
- ✓ Sublime Text Versión 3.2.1.
- ✓ Word 2016 o superior.
- ✓ Excel 2016 o superior.

6. IMPLEMENTACIÓN (Codificación Comentada)

```
PROGRAM Markov;
(*
AUTOR: Developeralta.
FECHA: 09 de Octubre del 2019.
PROPÓSITO: Por lo tanto se requiere un software capaz de obtener de una base de datos las siguientes
probabilidades:
    Número
    x<0
    0≤x≤1
    x≥1
    Cadena Alfanúmerica
    A...Z
    Caracteres especiales.
    A...Z y caracteres especiales.
Y con los resultados anteriores construir la cadena de Markov correspondiente y un reporte con los
siguientes puntos:
    Total de números encontrados.
    Número total de cadenas alfanúmericas encontradas.
    Total de números menores a cero.
    Total de números que se encuentren el siguiente intervalo 0≤x≤1.
    Total de números mayores a uno.
    Total de cadenas con solo letras.
    Total de cadenas con solo caracteres especiales.
    Total de cadenas con letras y caracteres especiales.
SUPOSICIONES: Ninguna.
LIMITANTES:
• El nombre del archivo es sensible a mayúsculas.
*)
//LIBRERÍAS
USES Pantalla,ArchivosSecuenciales,sysutils,Dos,Crt;
CONST
NReporte='ReporteMarkov-';
Extension='.txt';
```

```

Numero=1;
Cadena=2;
MayorCero=3;
MenorCero=4;
Intervalo=5;
Letra=6;
Especial=7;
Otro=8;
MaxRenglonesNR=7;
MaxColumnasNR=6;
MatrizAdyacenciaNR      :      ARRAY      [1..MaxRenglonesNR,1..MaxColumnasNR]      OF
INTEGER=((3,2,0,0,6,7),(3,0,0,0,0,0),(3,0,4,1,0,0),(5,0,0,0,0,0),(5,0,0,1,0,0),(0,0,0,1,6,0),(0,0
,0,1,0,7));
TYPE
  AContadores=ARRAY [1..8] OF INTEGER;
(* PROCEDIMIENTOS Y FUNCIONES *)
PROCEDURE EstadoActualAFD(VAR Nodo : INTEGER;CopiaSimbolo : CHAR);
(*
UTILIDAD: Encontrar el nodo al que llega el símbolo.
PRECONDICIÓN: 'Nodo' tiene el estado actual y 'CopiaSimbolo' el caracter siguiente.
POSCONDICIÓN: Se devuelve el valor del nodo en el que se encuentra 'CopiaSimbolo'.
*)
BEGIN
  IF(CopiaSimbolo>='0') AND (CopiaSimbolo<='9') THEN
    Nodo:=MatrizAdyacenciaNR[Nodo,1]
  ELSE
    BEGIN
      IF((CopiaSimbolo>='A')      AND      (CopiaSimbolo<='Z'))      OR      ((CopiaSimbolo>='a')      AND
(CopiaSimbolo<='z')) THEN
        Nodo:=MatrizAdyacenciaNR[Nodo,5]
      ELSE
        BEGIN
          CASE(CopiaSimbolo)OF
            '-':      (*Verificar si símbolo pertenece al caracter '-'.*)
              Nodo:=MatrizAdyacenciaNR[Nodo,2];
            '.':      (*Verificar si símbolo es el caracter '.'.*)
              Nodo:=MatrizAdyacenciaNR[Nodo,3];
            ',':      (*Verificar si símbolo es el caracter ','.*)
              Nodo:=MatrizAdyacenciaNR[Nodo,4];
            ' ':      (*Verificar si símbolo es el caracter ' '.*)
              Nodo:=MatrizAdyacenciaNR[Nodo,4];
            ELSE      (*Verifica si símbolo no pertenece al alfabeto*)
              BEGIN
                IF(ORD(CopiaSimbolo)=9) THEN
                  Nodo:=MatrizAdyacenciaNR[Nodo,4]
                ELSE
                  Nodo:=MatrizAdyacenciaNR[Nodo,6];
                END;
              END;
            END;
          END;
        END;
      END;
    END;
  END;
END;

PROCEDURE TratamientoDatos(CopiaNF : STRING;VAR Contadores : AContadores);
(*
UTILIDAD: Tratar la información del archivo verificando si son números o letras.
PRECONDICIÓN: 'CopiaNF' existe.
POSCONDICIÓN: 'Contadores' tiene el total de números y cadenas encontrados en el archivo.
*)
VAR
  Archivo : TArchivoTxt;
  Registro,CadenaN : STRING;
  Item,Estado,Longitud,EstadoA : INTEGER;
  Dato : REAL;
BEGIN
  FOR Item:=1 TO 8 DO
    Contadores[Item]:=0;
  ASSIGN(Archivo,CopiaNF);
  RESET(Archivo);
  WHILE (NOT (EOF(Archivo))) DO

```

```

BEGIN
  READLN (Archivo,Registro);
  Registro:=Registro+', ';
  Longitud:=LENGTH(Registro);
  Item:=1;
  Estado:=1;
  CadenaN:='';
  WHILE (Item<=Longitud) DO
    BEGIN
      EstadoA:=Estado;
      EstadoActualAFD(Estado,Registro[Item]);
      CASE (Estado) OF
        0: //Estado inválido
          BEGIN
            Contadores[Otro]:=Contadores[Otro]+1;
            REPEAT
              Item:=Item+1;
            UNTIL (Registro[Item]=' ' OR (Registro[Item]=' ') OR (ORD(Registro[Item])=9);
            Estado:=1;
            WRITELN('ERROR');
            CadenaN:='';
          END;
        1: //Estado Inicial
          BEGIN
            IF (LENGTH (CadenaN)>0) THEN
              BEGIN
                WRITELN(CadenaN);
                IF (EstadoA=3) OR (EstadoA=5) THEN
                  BEGIN
                    Contadores[Numero]:=Contadores[Numero]+1;
                    VAL (CadenaN,Dato);
                    IF (Dato<0) THEN
                      Contadores[MenorCero]:=Contadores[MenorCero]+1
                    ELSE
                      BEGIN
                        IF (Dato<1) THEN
                          Contadores[Intervalo]:=Contadores[Intervalo]+1
                        ELSE
                          Contadores[MayorCero]:=Contadores[MayorCero]+1;
                        END;
                      END
                    ELSE
                      BEGIN
                        Contadores[Cadena]:=Contadores[Cadena]+1;
                        IF (EstadoA=6) THEN
                          Contadores[Letra]:=Contadores[Letra]+1
                        ELSE
                          Contadores[Especial]:=Contadores[Especial]+1;
                        END;
                      END;
                    CadenaN:='';
                  END;
                ELSE //Estado inválido
                  CadenaN:=CadenaN+Registro[Item];
                END;
                Item:=Item+1;
              END;
            END;
          END;
        CLOSE (Archivo);
      END;

PROCEDURE CalcularProbabilidades(VAR Reporte : TArchivoTxt;CContadores : AContadores);
(*
UTILIDAD: Calcular las probabilidades de los 'CContadores' como valores.
PRECONDICIÓN: 'CContadores' tiene el total de números y cadenas y 'Reporte' existe y está abierto.
POSCONDICIÓN: Las probabilidades se almacenan en el archivo 'Reporte'.
*)
VAR
  Total : INTEGER;
  Probabilidad : STRING;

```

```

BEGIN
  Total:=CContadores[Numero]+CContadores[Cadena]+CContadores[Otro];
  WRITELN(Reporte, 'Probabilidades');
  WRITELN(Reporte, '-----');
  STR(CContadores[Numero]/Total:0:2, Probabilidad);
  WRITELN(Reporte, 'Número: '+Probabilidad);
  STR(CContadores[MenorCero]/CContadores[Numero]:0:2, Probabilidad);
  WRITELN(Reporte, '---> x<0: '+Probabilidad);
  STR(CContadores[Intervalo]/CContadores[Numero]:0:2, Probabilidad);
  WRITELN(Reporte, '---> 0<x<1: '+Probabilidad);
  STR(CContadores[MayorCero]/CContadores[Numero]:0:2, Probabilidad);
  WRITELN(Reporte, '---> x>1: '+Probabilidad);
  STR(CContadores[Cadena]/Total:0:2, Probabilidad);
  WRITELN(Reporte, 'Cadena: '+Probabilidad);
  STR(CContadores[Letra]/CContadores[Cadena]:0:2, Probabilidad);
  WRITELN(Reporte, '---> A...Z: '+Probabilidad);
  STR(CContadores[Especial]/CContadores[Cadena]:0:2, Probabilidad);
  WRITELN(Reporte, '---> Especial: '+Probabilidad);
  STR(CContadores[Otro]/Total:0:2, Probabilidad);
  WRITELN(Reporte, 'Otros: '+Probabilidad);
END;

VAR
  (* VARIABLES DE ENTRADA *)
  AOrigen, Opcion : STRING;
  (* VARIABLES DE SALIDA *)
  Reporte : TArchivoTxt;
  (* VARIABLES AUXILIARES *)
  ReporteNF : STRING;
  Contadores : AContadores;
BEGIN
  REPEAT
    BEGIN
      ClrScr;
      MostrarEncabezado('Markov 1.0');
      WRITE('Nombre del Archivo CSV: ');
      READLN(AOrigen);
      IF (VerificarArchivoTexto(AOrigen+'.csv')) THEN
        BEGIN
          ReporteNF:=NReporte+AOrigen+Extension;
          IF (CrearArchivoTexto(ReporteNF)) THEN
            BEGIN
              ASSIGN(Reporte, ReporteNF);
              APPEND(Reporte);
              WRITELN(Reporte, 'Reporte Markov Version 1.0':44);
              WRITELN(Reporte, DateTimeToStr(Now):38);
              WRITELN(Reporte, '-----');
              WRITELN(Reporte, 'Dato');
              TratamientoDatos(AOrigen+'.csv', Contadores);
              STR(Contadores[Numero], Opcion);
              WRITELN(Reporte, 'Numeros: '+Opcion);
              STR(Contadores[MenorCero], Opcion);
              WRITELN(Reporte, '---> x<0: '+Opcion);
              STR(Contadores[Intervalo], Opcion);
              WRITELN(Reporte, '---> 0<x<1 '+Opcion);
              STR(Contadores[MayorCero], Opcion);
              WRITELN(Reporte, '---> x>0: '+Opcion);
              STR(Contadores[Cadena], Opcion);
              WRITELN(Reporte, 'Cadenas: '+Opcion);
              STR(Contadores[Letra], Opcion);
              WRITELN(Reporte, '---> A...Z: '+Opcion);
              STR(Contadores[Especial], Opcion);
              WRITELN(Reporte, '---> Especial: '+Opcion);
              STR(Contadores[Otro], Opcion);
              WRITELN(Reporte, 'Otros: '+Opcion);
              CalcularProbabilidades(Reporte, Contadores);
              CLOSE(Reporte);
              MostrarArchivoTexto(ReporteNF);
            END
          END
        END
      END
    END
  UNTIL FALSE;
END

```



```
ELSE
  Writeln('ERROR 02: No se pudo crear el reporte :(');
END
ELSE
  Writeln('ERROR 01: No existe el archivo csv :(');
  Writeln('¿Quiere tratar otro archivo CSV? (SI)');
  Readln(Opcion);
  Opcion:=UpperCase(Opcion);
  Enter('');
END
UNTIL(Opcion<>'SI');
END.
```

7. RECOMENDACIONES

En futuras versiones, el programa puede optimizarse con base a las siguientes sugerencias:

- Mejorar la interfaz del usuario para incrementar la usabilidad.

8. BIBLIOGRAFÍA

Joyanes, L. (2003). *Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos*: McGraw Hill. Tercera edición.