

1° AVANCE DEL PROYECTO PARA TEORIA DE LA COMPUTACION

NOMBRE DEL PROYECTO: SIMULADOR DE AUTOMATAS FINITOS (SAF)

OBJETIVO:

DEMOSTRAR EL FUNCIONAMIENTO Y APLICACION DE LOS AUTOMATAS AL PUBLICO EN GENERAL MEDIANTE UNA FORMA DINAMICA Y SENCILLA PARA EL USUARIO PROMEDIO. CON ESTE PROYECTO DESARROLLADO EN JAVA PRETENDEMOS DAR A CONOCER QUE LOS AUTOMATAS SON HERRAMIENTAS UTILES PARA MODELAR Y ANALIZAR CUALQUIER SISTEMA EN EL UNIVERSO. HAN SIDO UTILIZADOS PARA MODELAR SISTEMAS FÍSICOS, CREAR INTÉRPRETES Y TRADUCTORES HASTA LOS MÁS COMPLEJOS COMO LOS COMPILADORES, ASÍ COMO INTERACCIONES ENTRE PARTÍCULAS, FORMACIÓN DE GALAXIAS, CINETICA DE SISTEMAS MOLECULARES Y CRECIMIENTO DE CRISTALES, ASÍ COMO DIVERSOS SISTEMAS BIOLÓGICOS A NIVEL CELULAR, MULTICELULAR Y POBLACIONAL. EL ALUMNO DESARROLLARA UN PROGRAMA HECHO EN JAVA ACORDE A LOS AUTOMATAS, PARA ESTO DEBERA PONER EN PRACTICA TODOS LOS CONOCIMIENTOS ADQUIRIDOS DURANTE SU APRENDIZAJE SOBRE EL LENGUAJE JAVA. EL PROGRAMA A REALIZAR SERVIRA DE HERRAMIENTA DIDACTICA PARA LOS USUARIOS INTERESADOS EN LA PRACTICA DE ESTE TEMA.

MISION:

PERMITIR AL USUARIO EXPERIMENTAR EN VIVO LAS APLICACIONES DE LOS AUTÓMATAS Y LOGRAR INTERESARSE POR EL TEMA DEMOSTRADO, FACILITANDO EL APRENDIZAJE A LOS MISMOS ASI COMO LOGRAR EL MAXIMO APROVECHAMIENTO DE LA HERRAMIENTA HACIA EL USUARIO INTERERESADO.

VISION:

EL USO DEL SOFTWARE DESARROLLADO SERA DIDÁCTICO, DINÁMICO Y ATRACTIVO PARA EL USUARIO. EL PROGRAMA PODRIA SER USADO POR TODOS LOS ESTUDIANTES CON NECESIDAD DE REFORZAR SUS CONOCIMIENTOS SOBRE AUTOMATAS.

MARCO TEORICO:

UN AUTÓMATA FINITO ES UN MODELO MATEMÁTICO DE UNA MÁQUINA QUE ACEPTA CADENAS DE UN LENGUAJE DEFINIDO SOBRE UN ALFABETO A. CONSISTE EN UN CONJUNTO FINITO DE ESTADOS Y UN CONJUNTO DE TRANSICIONES ENTRE ESOS ESTADOS, QUE DEPENDEN DE LOS SÍMBOLOS

DE LA CADENA DE ENTRADA. EL AUTÓMATA FINITO ACEPTA UNA CADENA X SI LA SECUENCIA DE TRANSICIONES CORRESPONDIENTES A LOS SÍMBOLOS DE X CONDUCE DESDE EL ESTADO INICIAL A UN ESTADO FINAL. SI PARA TODO ESTADO DEL AUTÓMATA EXISTE COMO MÁXIMO UNA TRANSICIÓN DEFINIDA PARA CADA SÍMBOLO DEL ALFABETO, SE DICE QUE EL AUTÓMATA ES DETERMINÍSTICO (AFD). SI A PARTIR DE ALGÚN ESTADO Y PARA EL MISMO SÍMBOLO DE ENTRADA, SE DEFINEN DOS O MÁS TRANSICIONES SE DICE QUE EL AUTÓMATA ES NO DETERMINÍSTICO (AFND). FORMALMENTE UN AUTÓMATA FINITO SE DEFINE COMO UNA 5-UPLA.

$M = \langle E, A, D, E_0, F \rangle$ DONDE

E: CONJUNTO FINITO DE ESTADOS

A: ALFABETO O CONJUNTO FINITO DE SÍMBOLOS DE ENTRADA

D: FUNCIÓN DE TRANSICIÓN DE ESTADOS, QUE SE DEFINE COMO

- D: $E \times A \rightarrow E$ SI EL AUTÓMATA ES DETERMINÍSTICO

- D: $E \times A \rightarrow P(E)$ SI EL AUTÓMATA ES NO DETERMINÍSTICO ($P(E)$ ES EL CONJUNTO POTENCIA DE E, ES DECIR EL CONJUNTO DE TODOS LOS SUBCONJUNTOS DE E)

E_0 : ESTADO INICIAL

F: CONJUNTO DE ESTADOS FINALES O ESTADOS DE ACEPTACIÓN

GENERALMENTE SE ASOCIA CON CADA AUTÓMATA UN GRAFO DIRIGIDO, LLAMADO DIAGRAMA DE TRANSICIÓN DE ESTADOS. CADA NODO DEL GRAFO CORRESPONDE A UN ESTADO. EL ESTADO INICIAL SE INDICA MEDIANTE UNA FLECHA QUE NO TIENE NODO ORIGEN. LOS ESTADOS FINALES SE REPRESENTAN CON UN CÍRCULO DOBLE. SI EXISTE UNA TRANSICIÓN DEL ESTADO E_i AL ESTADO E_j PARA UN SÍMBOLO DE ENTRADA A, EXISTE ENTONCES UN ARCO ROTULADO A DESDE EL NODO E_i AL NODO E_j .

LOS AUTÓMATAS VIENEN A SER MECANISMOS FORMALES QUE "REALIZAN" DERIVACIONES EN GRAMÁTICAS FORMALES. LA MANERA EN QUE LAS REALIZAN ES MEDIANTE LA NOCIÓN DE RECONOCIMIENTO. UNA PALABRA SERÁ GENERADA EN UNA GRAMÁTICA SI Y SÓLO SI LA PALABRA HACE TRANSITAR AL AUTÓMATA CORRESPONDIENTE A SUS CONDICIONES TERMINALES. POR ESTO ES QUE LOS AUTÓMATAS SON ANALIZADORES LÉXICOS (LLAMADOS EN INGLÉS "PARSERS") DE LAS GRAMÁTICAS A QUE CORRESPONDEN.

ESTOS SON LOS AUTÓMATAS FINITOS MÁS SENCILLOS. SE CONSTRUYEN A PARTIR DE UN CONJUNTO DE *ESTADOS* Q Y DE UN CONJUNTO DE *SÍMBOLOS DE ENTRADA* T . SU FUNCIONAMIENTO QUEDA DETERMINADO POR UNA

FUNCIÓN DE TRANSICIÓN $t: Q \times T \rightarrow Q$. SI $t(Q, S) = P$ ESTO SE INTERPRETA COMO QUE EL AUTÓMATA TRANSITA DEL ESTADO Q AL ESTADO P CUANDO

ARRIBA EL SÍMBOLO S . EN TODO AUTÓMATA FINITO SE CUENTA CON UN ESTADO INICIAL, $q_0 \in Q$ Y UN CONJUNTO DE ESTADOS FINALES $F \subset Q$. CON TODO ESTO DEFINIDO, LA ESTRUCTURA $\text{AutoReg} = (Q, T, t, q_0, F)$ ES UN AUTÓMATA REGULAR. DE MANERA NATURAL, T SE EXTIENDE A UNA FUNCIÓN DE TRANSICIÓN $T^* \rightarrow Q$: TODA PALABRA SE APLICA AL AUTÓMATA Y ÉSTE, PARTIENDO DEL ESTADO INICIAL, TRANSITA CON CADA SÍMBOLO DE LA PALABRA DADA SEGÚN LO ESPECIFIQUE T , CORRESPONDIENDO A ESE SÍMBOLO Y AL ESTADO ACTUAL EN EL AUTÓMATA. UNA PALABRA ES RECONOCIDA POR EL AUTOMATA SI LO HACE ARRIBAR A UN ESTADO FINAL. EL LENGUAJE DEL AUTÓMATA CONSTA DE TODAS LAS PALABRAS RECONOCIDAS.

EJEMPLO: SEA $\text{AutoReg} = (Q, T, t, q_0, F)$ EL AUTÓMATA CUYO CONJUNTO DE ESTADOS ES $Q = \{a, b, c\}$, EL DE SÍMBOLOS DE ENTRADA ES $T = \{0, 1\}$, SU ESTADO INICIAL ES $q_0 = a$ Y EL CONJUNTO DE ESTADOS FINALES ES $F = \{a\}$. SU TRANSICIÓN QUEDA DETERMINADA POR LA TABLA

t	0	1
a	b	a
b	c	a
c	c	c

OBSERVAMOS QUE, PARTIENDO DEL ESTADO A , MIENTRAS LLEGUEN 1'S SE ESTÁ EN EL ESTADO INICIAL, CON UN 0 SE PASA A B , CON UN SEGUNDO 0 SE PASA A C Y DE AHÍ NO SE SALE MÁS. EN B , AL LLEGAR UN 1 SE REGRESA AL ESTADO INICIAL. ASÍ PUES, PARA ARRIBAR AL ESTADO A DESDE A MISMO LA CADENA DE ENTRADA HA DE SER UNA SARTA DE VARIAS DE 1'S SEPARADAS ÉSTAS POR ÚNICOS 0'S. EN OTRAS PALABRAS, EL AUTÓMATA RECONOCE AL LENGUAJE $(1 + 0)^* 1^+$.

LOS AUTÓMATAS TIENEN COMO APLICACIÓN HACER MAS SENCILLA LAS TAREAS PARA EL SER HUMANO. UNO DE LOS EJEMPLOS DE APLICACIÓN ES LA DE UN CONTROLADOR DE RIEGO. EL AUTÓMATA CONTROLA EL REGADO DE UN CAMPO TENIENDO EN CUENTA VARIOS FACTORES QUE SON:

1. LA CANTIDAD DE AGUA DE UN DEPÓSITO ABASTECIDO CON LLUVIA TENIENDO EN CUENTA EL ÍNDICE DE PLUVIOSIDAD MINUTO A MINUTO.

2. LA HORA DEL DÍA, CON EL FIN DE REALIZAR EL REGADO A LAS HORAS MÁS FAVORABLES, QUE SON AQUELLAS EN LAS QUE LA EVAPORACIÓN ES MENOR.
3. LA HORA DEL ÚLTIMO REGADO, CON EL FIN DE ESPACIARLOS LO NECESARIO Y SUFICIENTE.

ESTO SE HARÁ MEDIANTE UN SISTEMA QUE CONTROLARÁ LA CANTIDAD DE AGUA QUE CAE EN BASE AL TIEMPO (1 ENTRADA). SE USARÁ UNA SALIDA DE RELÉ PARA CONTROL DE APERTURA Y CIERRE DE LA VÁLVULA DEL DEPÓSITO (1 SALIDA RELÉ). EL RESTO SERÁ CONTROL DEL RELOJ EN TIEMPO REAL DE QUE DISPONE EL EQUIPO Y EL PROGRAMA BASIC QUE LO MANEJA TODO.

FORMALMENTE UN AUTÓMATA FINITO DETERMINISTA CONSISTE EN UNA QUÍNTUPLA $(S, \Sigma, \delta, I, F)$ DONDE:

- S : ES UN CONJUNTO FINITO DE ESTADOS.
- Σ : ES EL ALFABETO DE LA MAQUINA.
- δ : ES UNA FUNCIÓN(FUNCIÓN DE TRANSICIÓN) DE $S \times S$ A S
- I : ESTADO INICIAL(UN ELEMENTO DE S)
- F : CONJUNTO DE ESTADOS DE ACEPTACIÓN (SUB-CONJUNTO DE S)

DIAGRAMA DE TRANSICIONES DETERMINISTA

PARA REPRESENTAR UN PROGRAMA EN EL MECANISMO DE CONTROL UTILIZAMOS UN DIAGRAMA DE TRANSICIONES CUYOS ESTADOS REPRESENTAN LOS ESTADOS DE LA MAQUINA Y CUYOS ARCOS REPRESENTAN UNA POSIBLE TRANSICIÓN DE LA MAQUINA. POR LO TANTO, LOS ESTADOS DE INICIO Y ACEPTACIÓN DEL DIAGRAMA CORRESPONDEN A LOS ESTADOS DE INICIO Y ACEPTACIÓN DEL AUTÓMATA.

UN DIAGRAMA PARA UN AFD ACEPTARA λ SI Y SOLO SI SU ESTADO INICIAL ES TAMBIÉN UN ESTADO DE ACEPTACIÓN.

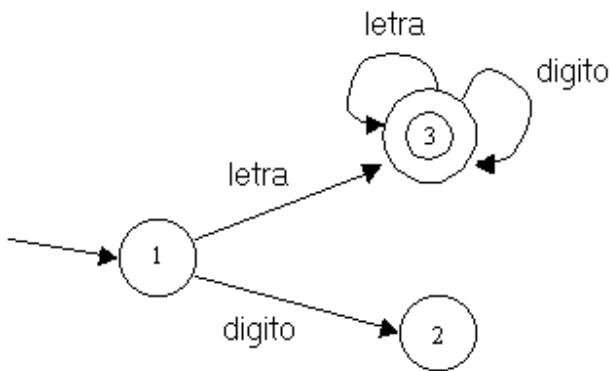
EL REQUISITO DEL DETERMINISMO IMPONE CIERTAS RESTRICCIONES SOBRE LOS DIAGRAMAS DE TRANSICIONES QUE PUEDEN APARECER EN LOS PROGRAMAS PARA UN AUTÓMATA FINITO DETERMINISTA. SE DICE QUE UN DIAGRAMA DE TRANSICIONES ES DETERMINISTA SI CUMPLE LAS SIGUIENTES CONDICIONES:

- EN PARTICULAR, CADA ESTADO DE ESTOS DIAGRAMAS SOLO DEBE TENER UN ARCO QUE SALE PARA CADA SÍMBOLO DEL ALFABETO; DE LO CONTRARIO, UNA MAQUINA QUE LLEGA A ESTE ESTADO SE ENFRENTARA A UNA ELECCIÓN DE CUAL DEBE SER EL ARCO A SEGUIR.

- ADEMÁS, DICHO DIAGRAMA DEBE ESTAR COMPLETAMENTE DEFINIDO, ES DECIR DEBE EXISTIR POR LO MENOS UN ARCO PARA CADA SÍMBOLO DEL ALFABETO; DE LO CONTRARIO, UNA MAQUINA QUE LLEGA A ESTE ESTADO PUEDE ENFRENTARSE A UNA SITUACIÓN DONDE NO PUEDA APLICARSE NINGUNA TRANSICIÓN.

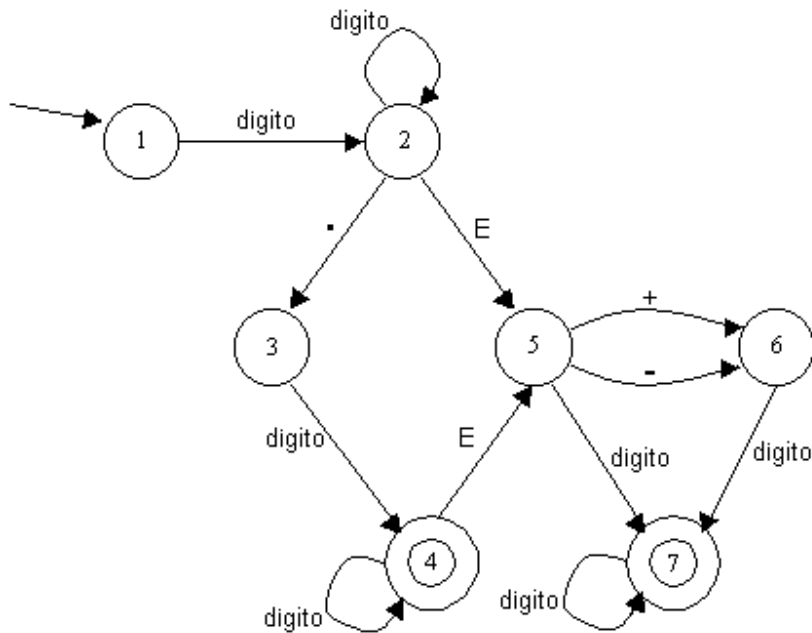
EJEMPLO 1:

EL SIGUIENTE DIAGRAMA NO ES DETERMINISTA YA QUE NO ESTA COMPLETAMENTE DEFINIDO; NO REPRESENTA CUAL SERÁ LA ACCIÓN QUE DEBE OCURRIR SI SE RECIBE UNA LETRA O UN DÍGITO MIENTRAS SE ENCUENTRA EN EL ESTADO 2.



EJEMPLO 2:

EL SIGUIENTE DIAGRAMA TIENE PROBLEMAS SIMILARES YA QUE ENTRE OTRAS COSAS NO DESCRIBE QUE DEBERÁ SUCEDER SI RECIBE UN PUNTO MIENTRAS SE ENCUENTRA EN EL ESTADO INICIAL.



NO OBSTANTE, LOS DOS DIAGRAMAS VISTOS ANTERIORMENTE NO TIENEN MAS DE UN ARCO DE SALIDA DE UN ESTADO PARA CADA SÍMBOLO Y, POR CONSIGUIENTE, PUEDEN MODIFICARSE PARA AJUSTARSE A LOS REQUISITOS DEL DETERMINISMO, APLICANDO LO SIGUIENTE:

- AÑADIMOS UN ESTADO QUE REPRESENTARA UN PAPEL DE CAPTACIÓN GLOBAL
- PARA CADA SÍMBOLO DEL ALFABETO, DIBUJAR UN ARCO ROTULADO CON DICHO SÍMBOLO, QUE EMPIEZA Y TERMINA EN ESTE NUEVO ESTADO.
- AGREGAMOS ARCOS DE LOS OTROS ESTADOS A ESTE NUEVO, HASTA QUE CADA UNO DE LOS ESTADOS SEA EL ORIGEN DE UN ARCO PARA CADA SÍMBOLO DEL ALFABETO.

DIAGRAMA MODIFICADO DEL EJEMPLO 1.

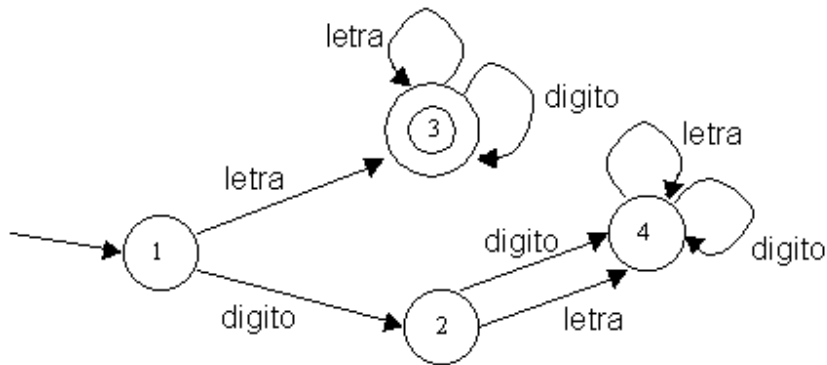
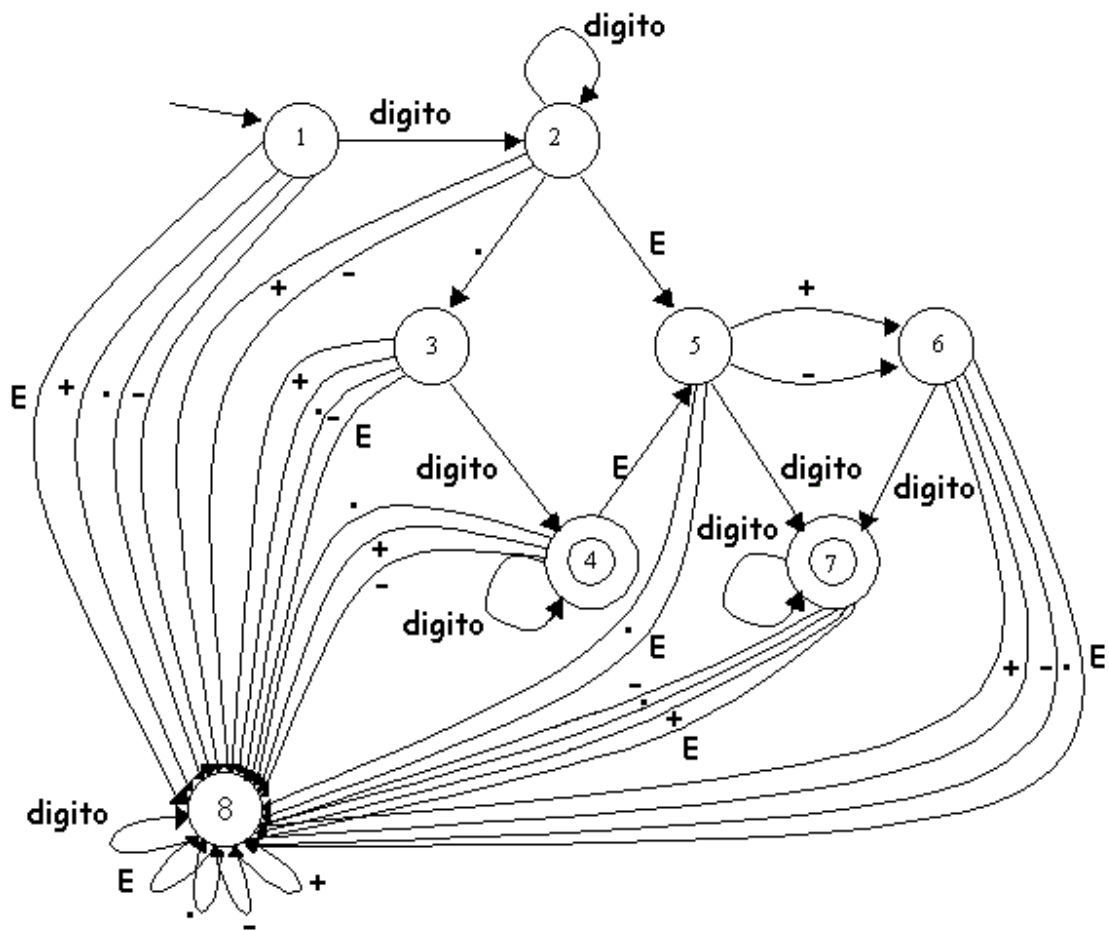


DIAGRAMA MODIFICADO DEL EJEMPLO 2.



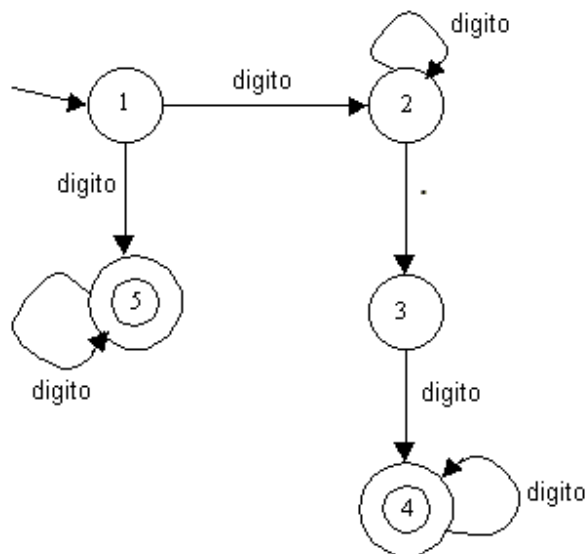
EN ESTE EJERCICIO EL NUEVO ESTADO ES EL NUMERO 8. OBSERVE QUE EN DIAGRAMA ORIGINAL LA OCURRENCIA DE UNA CADENA INACEPTABLE OCASIONABA UN ERROR AL SOLICITAR EL RECORRIDO DE UN ARCO INEXISTENTE. EN EL DIAGRAMA MODIFICADO, UNA CADENA INACEPTABLE OCASIONA QUE LA MAQUINA RECORRA UN ARCO A ESTADO 8, DONDE PERMANECE HASTA ALCANZAR EL FINAL DE LA CADENA DE ENTRADA. AL

LLEGAR A ESTE PUNTO SE RECHAZARA LA CADENA, YA QUE EL ESTADO 8 NO ES DE ACEPTACIÓN. POR ESTO, LOS DOS DIAGRAMAS SON EQUIVALENTES EN LO QUE SE REFIERE A QUE ACEPTAN LAS MISMAS CADENAS; DIFIEREN SOLO EN LA MANERA EN QUE LLEGAN A SUS CONCLUSIONES.

AUTOMATA FINITO NO DETERMINISTA

ESTA MAQUINA SE PARECE MUCHO A UN AFD, PUES TAMBIÉN ANALIZA CADENAS CONSTRUIDAS A PARTIR DE UN S Y SOLO PUEDE TENER UN NUMERO FINITO DE ESTADOS, ALGUNOS DE LOS CUALES SON DE ACEPTACIÓN Y UNO ES EL ESTADO INICIAL. A DIFERENCIA DE LOS AFD, LA TRANSICIÓN QUE SE EJECUTA EN UNA ETAPA DADA DE UN AFN PUEDE SER INCIERTA, ES POSIBLE APLICAR CERO, UNA O MAS DE UNA TRANSICIÓN MEDIANTE EL MISMO SÍMBOLO DE ENTRADA, COMO SUCEDER CON UNA MAQUINA QUE NO ESTA COMPLETAMENTE DEFINIDA.

EJEMPLO, DIAGRAMA DE TRANSICIONES QUE ACEPTA CADENAS QUE REPRESENTAN ENTEROS O CADENAS QUE REPRESENTAN NÚMEROS REALES EN NOTACIÓN DECIMAL.



UN AFN ACEPTA UNA CADENA SI ES POSIBLE QUE SU ANÁLISIS DEJE A LA MAQUINA EN UN ESTADO DE ACEPTACIÓN.

DE MANERA FORMAL, UN AFN SE DEFINE COMO SIGUE, UN AFN CONSISTE EN UNA QUINTUPLA (S, Σ, P, I, F) DONDE:

- S ES UN CONJUNTO FINITO DE ESTADOS.
- Σ ES EL ALFABETO DE LA MAQUINA

- P ES UN SUB-CONJUNTO DE SXS XS LLAMADA RELACIÓN DE TRANSICIONES.
- I ES LE ESTADO INICIAL (UN ELEMENTO DE S)
- F ES LA COLECCIÓN DE ESTADOS DE ACEPTACIÓN (UN SUB-CONJUNTO DE S).

DESARROLLO:

EL PROYECTO TRATARA DE UN SIMULADOR DE AUTÓMATAS , EL SISTEMA PEDIRA EL NUMERO DE ESTADOS DEL AUTOMATA, LOS ESTADOS FINALES Y EL INICIAL, ASI COMO LA CADENA QUE SE QUIERE COMPROBAR. POR CONSIGUIENTE EVALUARA LA CADENA INTRODUCIDA Y SE MOSTRARA LA TABLA DE TRANSICIONES CORRESPONDIENTE. SE PRETENDE QUE SEA UNA HERRAMIENTA DIDÁCTICA EN EL APRENDIZAJE DE AUTÓMATAS, SE ESPERA AGREGAR TODAS LAS OPCIONES MOSTRADAS.

ANALISIS

ANÁLISIS DE LOS REQUISITOS:

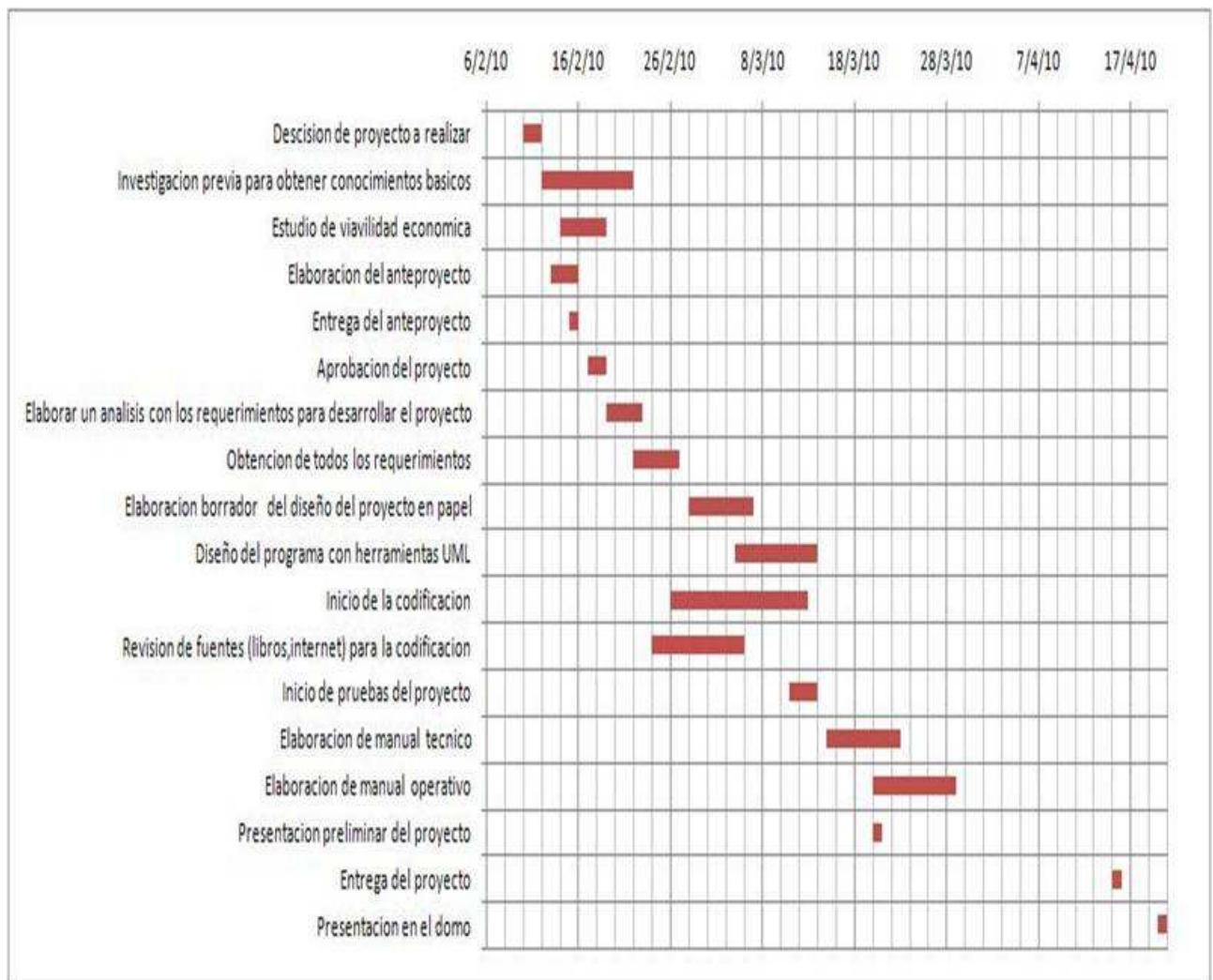
- PLATAFORMA JAVA (JDK 6)
- EDITOR DE CÓDIGO (TEXT PAD, BLUE J, NEAT BEANS, ETC)
- CONOCIMIENTOS PREVIOS DE PROGRAMACIÓN O.O
- CONOCIMIENTOS PREVIOS DE TEORÍA DE LA COMPUTACIÓN
- SISTEMA OPERATIVO WINDOWS (XP, VISTA, SEVEN)

TABLA DE EVENTOS:

ACTIVIDAD	INICIO	DURACION (DIAS)	TERMINACION
A DECISIÓN DE PROYECTO A REALIZAR	10/02/2010	2	12/02/2010
B INVESTIGACIÓN PREVIA PARA OBTENER CONOCIMIENTOS BÁSICOS	12/02/2010	10	22/02/2010
C ESTUDIO DE VIABILIDAD ECONÓMICA	14/02/2010	5	19/02/2010
D ELABORACIÓN DEL ANTEPROYECTO	13/02/2010	3	16/02/2010
E ENTREGA DEL ANTEPROYECTO	15/02/2010	1	16/02/2010
F APROBACIÓN DEL PROYECTO	17/02/2010	2	19/02/2010
G ELABORAR UN ANÁLISIS CON LOS REQUERIMIENTOS PARA DESARROLLAR EL PROYECTO	19/02/2010	4	23/02/2010
H OBTENCIÓN DE TODOS LOS REQUERIMIENTOS	22/02/2010	5	27/02/2010
I ELABORACIÓN BORRADOR DEL DISEÑO DEL PROYECTO EN PAPEL	28/02/2010	7	07/03/2010

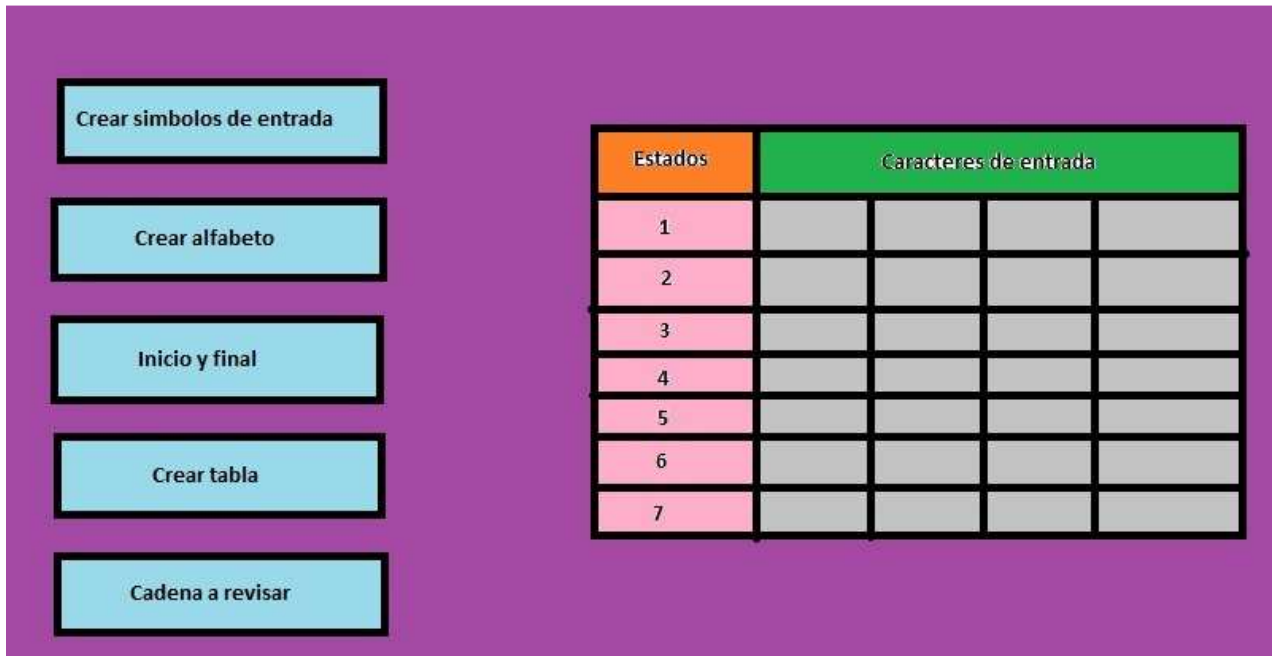
J	DISEÑO DEL PROGRAMA CON HERRAMIENTAS UML	05/03/2010	9	14/03/2010
K	INICIO DE LA CODIFICACIÓN	26/02/2010	15	13/03/2010
L	REVISIÓN DE FUENTES (LIBROS, INTERNET) PARA LA CODIFICACIÓN	24/02/2010	10	06/03/2010
M	INICIO DE PRUEBAS DEL PROYECTO	11/03/2010	3	14/03/2010
N	ELABORACIÓN DE MANUAL TÉCNICO	15/03/2010	8	23/03/2010
O	ELABORACIÓN DE MANUAL OPERATIVO	20/03/2010	9	29/03/2010
P	PRESENTACIÓN PRELIMINAR DEL PROYECTO	20/03/2010	1	21/03/2010
Q	ENTREGA DEL PROYECTO	15/04/2010	1	16/04/2010
R	PRESENTACIÓN EN EL DOMO	20/04/2010	1	21/04/2010

DIAGRAMA DE GANTT

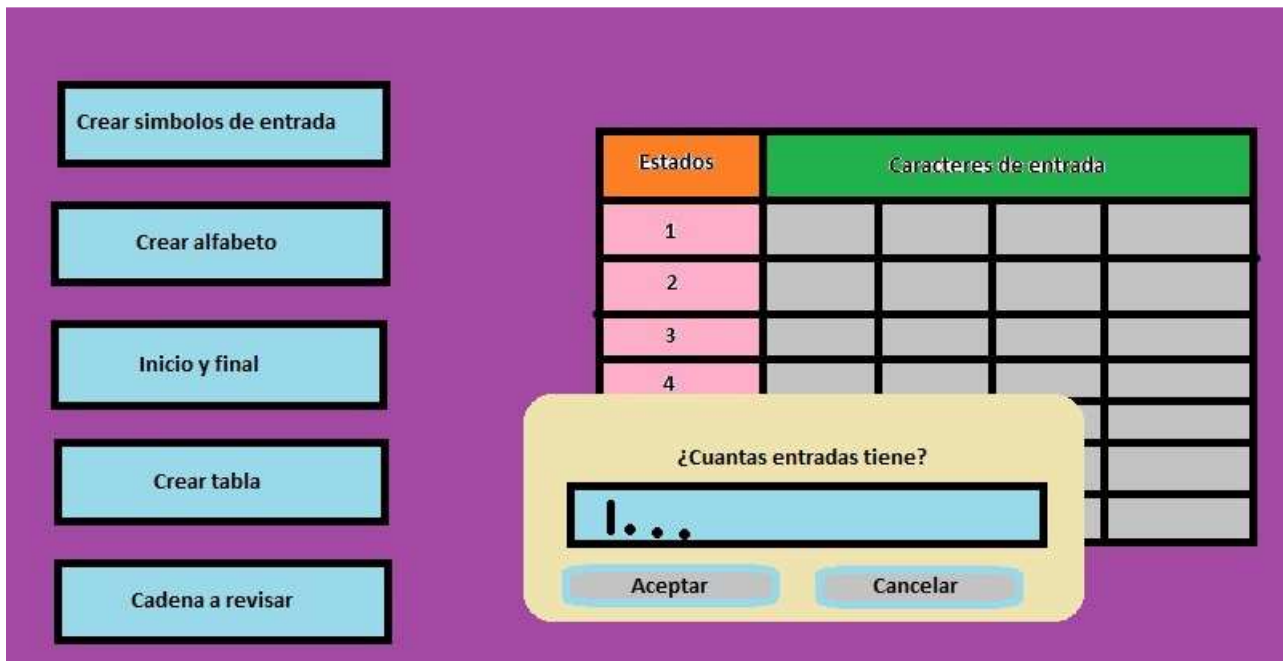


DISEÑO

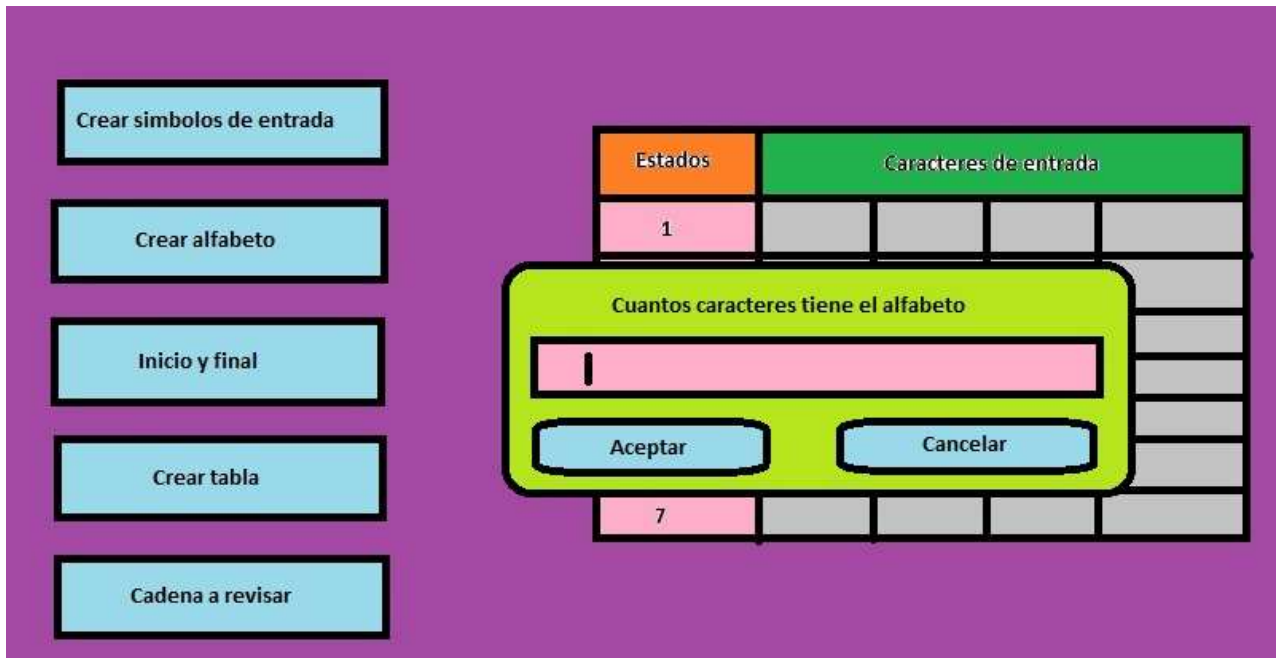
LA VENTANA DE ENTRADA ES LA SIGUIENTE



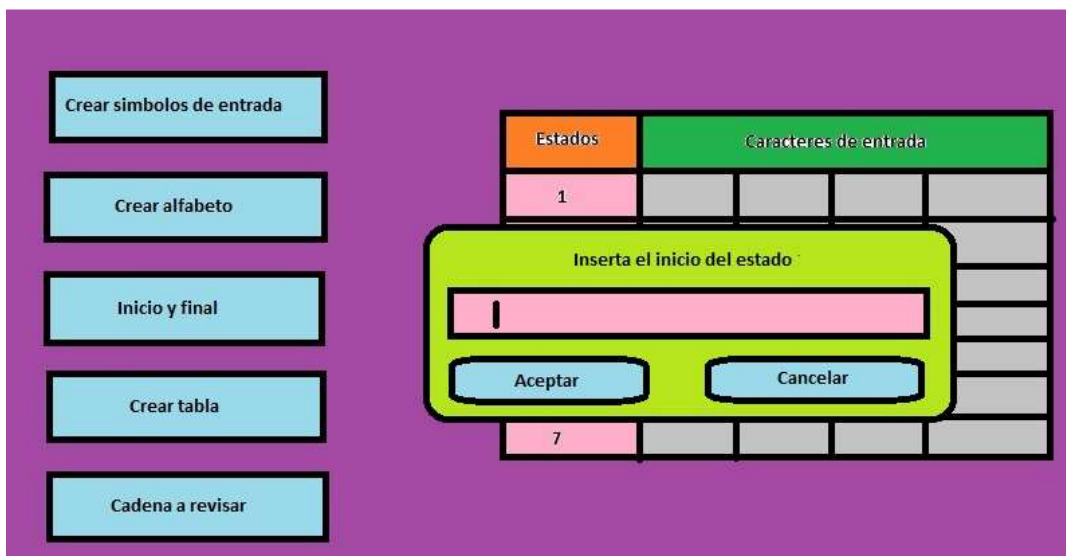
TENDRA 5 BOTONES PARA EVALUAR LA CADENA Y UN TABLA DE TRANSICIONES.EL PRIMER BOTON PEDIRA CUANTAS ENTRADAS TIENE EL AUTOMATA ES DECIR LOS ESTADOS QUE LO COMPONEN:



EL SEGUNDO BOTON "CREAR ALFABETO" PEDIRA LOS SIMBOLOS DE ENTRADA DEL AUTOMATA, PRIMERO DEBEN ENTRAR EL NUMERO DE CARACTERES DEL ALFABETO Y DESPUES LOS SIMBOLOS UNO POR UNO COMO SE MUESTRA A CONTINUACION:

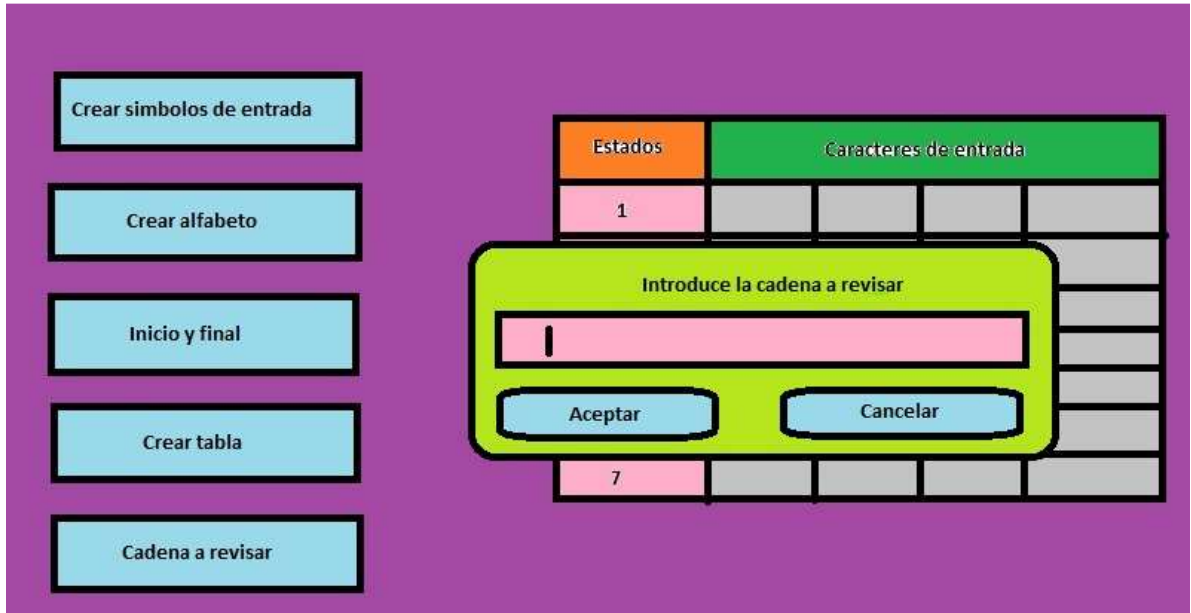


EL TERCER BOTON "INICIO Y FINAL" ES PARA ESPECIFICAR QUE ESTADO ES EL PRIMERO Y CUAL O CUALES SON LAS SALIDAS.

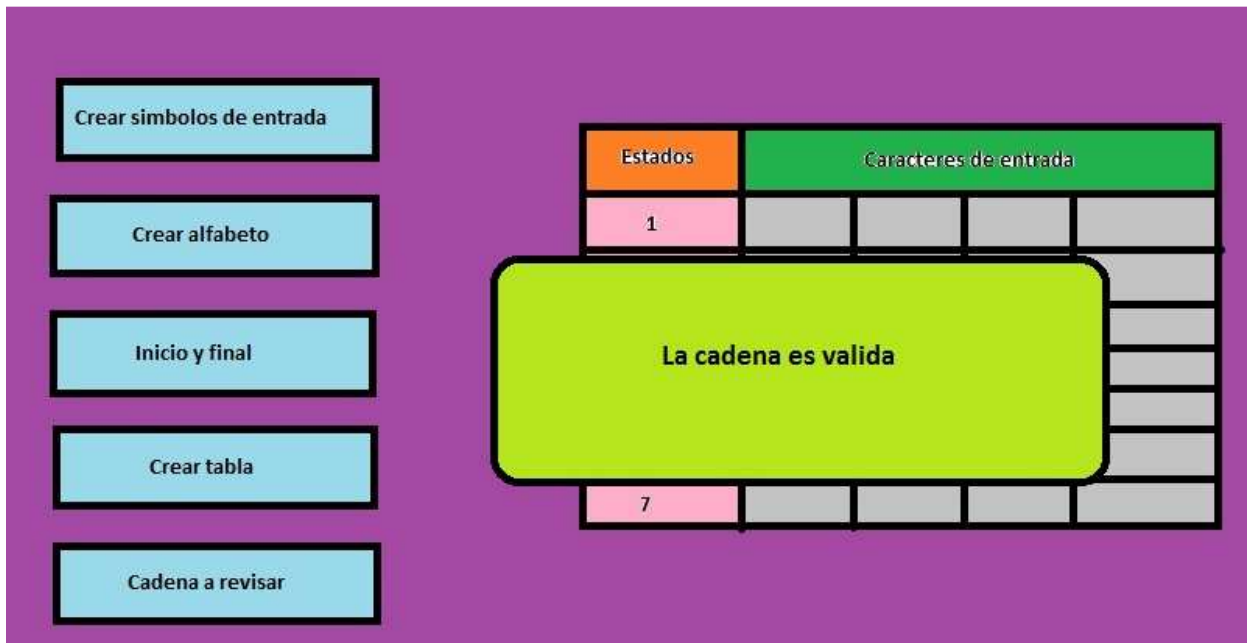


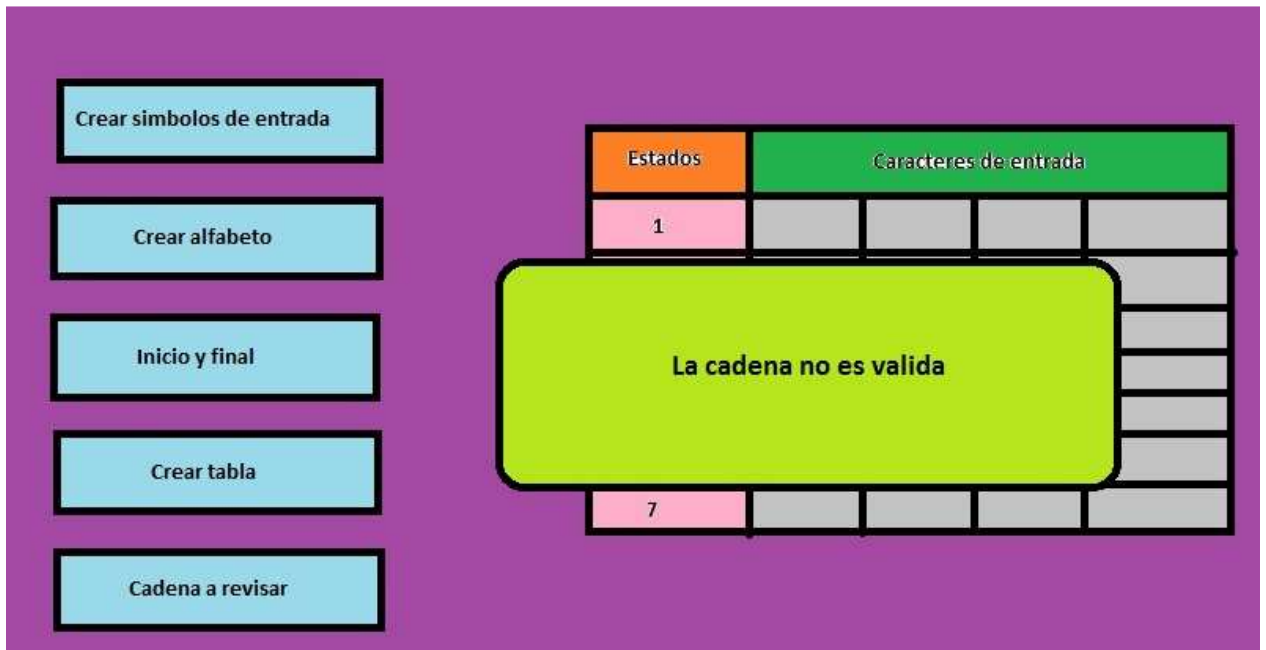
EL SIGUIENTE BOTON "CREAR TABLA" TENDRA LA FINALIDAD DE CREAR LA TABLA DE TRANSICION CON LOS DATOS RECABADOS Y SE LLENARAN LOS ESPACIOS QUE ANTES ESTABAN VACIOS DE LA TABLA. PARA TERMINAR EL

BOTON " CADENA A REVISAR" PEDIRA UNA CADENA QUE EVALUARA POSTERIORMENTE :



EL RESULTADO DE LA EVALUACION SE MOSTRARA POR UNA VENTANA QUE SE MUESTRE YA SEA VALIDA O NO:





CODIFICACION

UNA PRIMERA PARTE DEL PROGRAMA A DESARROLLAR SE MUESTRA A CONTRINUACION, PARA CREAR LA INTERFAZ PENSADA NECESITAMOS DOS CLASES, UNA PARA LOS GRAFICOS Y LOS EVENTOS PRINCIPALES ; LA OTRA ES UNA CLASE METODO DONDE TRABAJAREMOS CON EL METODO METODO QUE NECESITAMOS PARA COMPLEMENTARLA:

CLASE GRAFICO

[VIEW SOURCE](#)

[PRINT?](#)

```

001 IMPORT JAVA.AWT.*;
002 IMPORT JAVA.AWT.EVENT.*;
003 IMPORT JAVAX.SWING.*;
004
005 PUBLIC CLASS GRAFICO EXTENDS JFrame
006 {
007     //VARIABLES
008     PRIVATE METODO METODO = NEW METODO();
009     STRING INICIO,ULTIMO;
010     STRING Q[];

```

```

011     INT CUANTOSQ;
012     STRING TABLA[][];
013     INT CUANTOSALF;
014     STRING ALFA[];
015
016     PRIVATE CONTAINER CONTENEDOR;
017     PRIVATE GRIDBAGLAYOUT ESQUEMA;
018     PRIVATE GRIDBAGCONSTRAINTS RESTRICCIONES;
019     PRIVATE JBUTTON BOTONQ, BOTONALFA, BOTONTABLA,
    BOTONREVISAR, BOTONINIFIN;
020
021     PUBLIC GRAFICO( )
022     {
023         SUPER( "AUTOMATAS" );
024         CONTENEDOR = GETCONTENTPANE();
025         ESQUEMA = NEW GRIDBAGLAYOUT();
026         CONTENEDOR.SETLAYOUT( ESQUEMA );
027
028         BOTONQ = NEW JBUTTON("CREAR SIMB. DE EDOS.");
029         BOTONALFA = NEW JBUTTON("CREAR ALFABETO");
030         BOTONTABLA = NEW JBUTTON("CREAR TABLA");
031         BOTONREVISAR = NEW JBUTTON ("CADENA A REVISAR");
032         BOTONINIFIN = NEW JBUTTON ("INICIO Y FINAL");
033         //DESACTIVAR BOTONES
034         BOTONTABLA.SETENABLED(FALSE);
035         BOTONREVISAR.SETENABLED(FALSE);
036         BOTONINIFIN.SETENABLED(FALSE);
037
038
039
040         // INSTANCIAR RESTRICCIONES DE GRIDBAGLAYOUT
041         RESTRICCIONES = NEW GRIDBAGCONSTRAINTS();
042
043
044         // CREAR COMPONENTES DE GUI
045
046         // WEIGHTX Y WEIGHTY PARA AREATEXTO1 SON 0: EL VALOR
    PREDETERMINADO
047         // ANCHOR PARA TODOS LOS COMPONENTES ES CENTER: EL VALOR
    PREDETERMINADO
048         RESTRICCIONES.FILL = GRIDBAGCONSTRAINTS.BOTH;
049         AGREGARCOMPONENTE(BOTONQ, 0, 0, 2, 1);

```

```

050     AGREGARCOMPONENTE(BOTONALFA,1,0,2,1);
051     AGREGARCOMPONENTE(BOTONINIFIN,2,0,2,1);
052     AGREGARCOMPONENTE(BOTONTABLA,3,0,2,1);
053     AGREGARCOMPONENTE(BOTONREVISAR,4,0,2,1);
054     MANEJADORBOTON MANEJADOR = NEW MANEJADORBOTON();
055     BOTONQ.ADDACTIONLISTENER(MANEJADOR);
056     BOTONALFA.ADDACTIONLISTENER(MANEJADOR);
057     BOTONTABLA.ADDACTIONLISTENER(MANEJADOR);
058     BOTONINIFIN.ADDACTIONLISTENER(MANEJADOR);
059     BOTONREVISAR.ADDACTIONLISTENER(MANEJADOR);
060     //DESACTIVAR BOTONES
061
062     SETSIZE( 500, 500 );
063     SETLOCATIONRELATIVETO(NULL);
064     SETVISIBLE( TRUE );
065
066 } // FIN DEL CONSTRUCTOR DE DEMOGRIDBAG
067
068 // M?TODO PARA ESTABLECER RESTRICCIONES
069
070 PRIVATE CLASS MANEJADORBOTON IMPLEMENTS ACTIONLISTENER
071 {
072     PUBLIC VOID ACTIONPERFORMED( ACTIONEVENT EVENTO )
073     {
074         IF(EVENTO.GETSOURCE() == BOTONINIFIN)
075         {
076             INICIO = JOPTIONPANE.SHOWINPUTDIALOG("INSERTA EL INICIO
DE ESTADO");
077             AGREGARCOMPONENTE(NEW JTEXTFIELD("INICIO: " +INICIO+"  ")
,6,0,1,1);
078             ULTIMO = JOPTIONPANE.SHOWINPUTDIALOG("INSERTA EL FINAL
DE ESTADO");
079             AGREGARCOMPONENTE(NEW JTEXTFIELD("FINAL: "+ULTIMO)
,6,1,1,1);
080             SWINGUTILITIES.UPDATECOMPONENTTREEUI(CONTENEDOR);
081         }
082         IF(EVENTO.GETSOURCE() ==BOTONQ)
083         {
084             CUANTOSQ=METODO.CREAINT("¿CUANTAS ENTRADAS SON?: (Q)");
085             Q=NEW STRING[CUANTOSQ];
086             Q = METODO.CREARQ (CUANTOSQ,Q);
087             IF(CUANTOSQ>=1 && CUANTOSALF>=1)
088             {

```



```

089             BOTONTABLA.SETENABLED(TRUE);
090         }
091         BOTONINIFIN.SETENABLED(TRUE);
092         FOR(INT Z=0;Z < Q.LENGTH; Z++)
093             AGREGARCOMPONENTE(NEW JTEXTFIELD("
"+Q[Z]),Z+1,2,1,1);
094             SWINGUTILITIES.UPDATECOMPONENTTREEUI(CONTENEDOR);
095         }
096         IF(EVENTO.GETSOURCE()==BOTONALFA)
097         {
098             CUANTOSALF=METODO.CREAINT("¿CUANTOS CARACTERES TIENE EL
ALFABETO?:");
099             ALFA=NEW STRING[CUANTOSALF];
100             ALFA = METODO.CREARALFABETO(CUANTOSALF, ALFA);
101             IF(CUANTOSQ>=1 && CUANTOSALF>=1)
102             {
103                 BOTONTABLA.SETENABLED(TRUE);
104             }
105
106             AGREGARCOMPONENTE(NEW JTEXTFIELD("EDO/ALFA") ,0,2,1,1);
107             FOR(INT M=0;M<=ALFA.LENGTH-1;M++)
108                 AGREGARCOMPONENTE(NEW JTEXTFIELD("
"+ALFA[M]),0,M+3,1,1);
109             SWINGUTILITIES.UPDATECOMPONENTTREEUI(CONTENEDOR);
110         }
111         IF(EVENTO.GETSOURCE()==BOTONREVISAR)
112         {
113             INT CONTADOR=0;
114             STRING CAD[] = NEW STRING[METODO.CREAINT("¿QUE LARGO
TIENE LA CADENA?")];
115
116             FOR(INT CONTA=0; CONTA< CAD.LENGTH; CONTA++)
117                 CAD[CONTA] = JOPTIONPANE.SHOWINPUTDIALOG("INSERTAR EL
VALOR "+(CONTA+1)+" DE LA CADENA");
118
119             FOR(INT CONTA1=0; CONTA1< CAD.LENGTH; CONTA1++)
120                 FOR(INT CONTA=0; CONTA< ALFA.LENGTH; CONTA++)
121                     IF(ALFA[CONTA].EQUALS(CAD[CONTA1]) ||
CAD[CONTA1].EQUALS("")) || CAD[CONTA1].EQUALS(" ") )
122                         CONTADOR++;
123
124             IF(CONTADOR==CAD.LENGTH)
125                 INICIO = METODO.REVISAR(ALFA, Q, TABLA, CAD, INICIO);

```

```

126             ELSE
127                 JOPTIONPANE.SHOWMESSAGEDIALOG(NULL, "LA CADENA TIENE
ERRORES");
128
129
130             IF( INICIO.EQUALS(ULTIMO))
131             {
132                 JOPTIONPANE.SHOWMESSAGEDIALOG(NULL, "EL AUTOMATA
133 COMPLETO LA CADENA");
134             }
135             ELSE
136                 JOPTIONPANE.SHOWMESSAGEDIALOG(NULL, "EL AUTOMATA NO
137 COMPLETO LA CADENA");
138             SWINGUTILITIES.UPDATECOMPONENTTREEUI (CONTENEDOR);
139         }
140         IF(EVENTO.GETSOURCE()==BOTONTABLA)
141         {
142             TABLA = METODO.CREARTABLA(ALFA, CUANTOSQ,
143 CUANTOSALF, Q);
144             FOR(INT Z=0; Z < Q.LENGTH; Z++)
145             {
146                 FOR(INT J=0; J < ALFA.LENGTH; J++)
147                 {
148                     AGREGARCOMPONENTE (NEW JTEXTFIELD ("
149 "+TABLA[Z][J]), Z+1, J+3, 1, 1);
150                 }
151             }
152             INT CONTA=0;
153             FOR(INT Z=0; Z < Q.LENGTH; Z++)
154             FOR(INT J=0; J < ALFA.LENGTH; J++)
155             {
156                 FOR(INT SIM=0; SIM < Q.LENGTH; SIM++)
157                 {
158                     IF(Q[SIM].EQUALS(TABLA[Z][J]) )
159                     CONTA++;
160                 }
161                 IF( TABLA[Z][J].EQUALS(" ") ||
162 TABLA[Z][J].EQUALS(" ") )
163                     CONTA++;
164             }
165             IF(CONTA == CUANTOSQ*CUANTOSALF)
166                 BOTONREVISAR.SETENABLED(TRUE);
167             ELSE

```

```

164         {
165             BOTONREVISAR.SETENABLED(FALSE);
166             JOPTIONPANE.SHOWMESSAGEDIALOG(NULL, "LA TABLA DE
ESTADO TIENE SIMBOLOS DE ESTADO QUE NO ENTIENDE\NCORREGIR POR
FAVOR.");
167         }
168         SWINGUTILITIES.UPDATECOMPONENTTREEUI( CONTENEDOR );
169     }
170
171 }
172 }
173     PRIVATE VOID AGREGARCOMPONENTE( COMPONENT COMPONENTE, INT FILA, INT
COLUMNNA, INT ANCHURA, INT ALTURA )
174     {
175         // ESTABLECER GRIDX Y GRIDY
176         RESTRICCIONES.GRIDX = COLUMNNA;
177         RESTRICCIONES.GRIDY = FILA;
178
179         // ESTABLECER GRIDWIDTH Y GRIDHEIGHT
180         RESTRICCIONES.GRIDWIDTH = ANCHURA;
181         RESTRICCIONES.GRIDHEIGHT = ALTURA;
182
183         // ESTABLECER RESTRICCIONES Y AGREGAR COMPONENTE
184         ESQUEMA.SETCONSTRAINTS( COMPONENTE, RESTRICCIONES );
185         CONTENEDOR.ADD( COMPONENTE );
186     }
187     PUBLIC STATIC VOID MAIN( STRING ARGS[] )
188     {
189         GRAFICO APLICACION = NEW GRAFICO();
190         APLICACION.SETDEFAULTCLOSEOPERATION( JFrame.EXIT_ON_CLOSE );
191     }
192
193 } // FIN DE LA CLASE DEMOGRIDBAG

```

CLASE METODO

[VIEW SOURCE](#)

[PRINT?](#)

```

01 IMPORT JAVAX.SWING.*;
02 PUBLIC CLASS METODO
03 {
04     INT ACTUALQ=0;
05     INT ACTUALALF=0;
06     PUBLIC STRING[] CREARQ(INT TOTALQ,STRING[] Q)
07     {

```

```

08     ACTUALQ=0;
09     WHILE(ACTUALQ<=TOTALQ-1)
10     {
11         Q[ACTUALQ]=JOPTIONPANE.SHOWINPUTDIALOG("INSERTA LA ENTRADA
"+ ACTUALQ + ":");
12         ACTUALQ++;
13     }
14     RETURN(Q);
15 }
16 PUBLIC STRING[] CREAMALFABETO(INT TOTALALF, STRING ALFABETO[])
17 {
18     ACTUALALF=0;
19     WHILE(ACTUALALF < TOTALALF)
20     {
21         ALFABETO[ACTUALALF]=JOPTIONPANE.SHOWINPUTDIALOG("INSERTAR
EL CARACTER:" + ACTUALALF + " DEL ALFABETO");
22         ACTUALALF++;
23     }
24     RETURN(ALFABETO);
25 }
26 PUBLIC INT CREAMINT(STRING S)
27 {
28     RETURN(INTEGER.PARSEINT(JOPTIONPANE.SHOWINPUTDIALOG(S)));
29 }
30
31 PUBLIC STRING[][] CREARTABLA(STRING ALFA[], INT CUANTOSQ, INT
CUANTOSALF, STRING Q[])
32 {
33     STRING [][]TABLA=NEW STRING[CUANTOSQ][CUANTOSALF];
34     FOR(INT I=0;I<=CUANTOSQ-1; I++)
35         FOR(INT J=0; J<=CUANTOSALF-1; J++)
36             TABLA[I][J]=JOPTIONPANE.SHOWINPUTDIALOG("INSERTE
VALORES DE LA TABLA DE EDO.");
37     RETURN(TABLA);
38 }
39
40
41
42 PUBLIC STRING REVISAR(STRING ALFA[], STRING Q[], STRING
TABLA[][] , STRING[] CADENA, STRING INICIO)
43 {
44     INT G=0,H=0;
45     WHILE(G < CADENA.LENGTH)

```

```

46     {
47         IF(CADENA[G].EQUALS(" ") || CADENA[G].EQUALS(""))
48     {
49         INICIO=CADENA[G];
50         G=CADENA.LENGTH;
51
52     }
53     ELSE
54     {
55         WHILE(H < ALFA.LENGTH)
56     {
57         IF(CADENA[G].EQUALS(ALFA[H]))
58     {
59         FOR(INT A=0; A < Q.LENGTH;A++)
60     {
61         IF(INICIO.EQUALS(Q[A]))
62     {
63         INICIO=TABLA[A][H];
64         A=Q.LENGTH;
65         H=ALFA.LENGTH;
66         G++;
67     }
68     ELSE
69     {
70     INICIO.EQUALS(" ") ||
71         {
72         G=ALFA.LENGTH;
73         INICIO = TABLA[A][H];
74     }
75     }
76     }
77     }
78     H++;
79     }
80     H=0;
81     }
82     }
83     RETURN( INICIO );
84     }
85
86

```

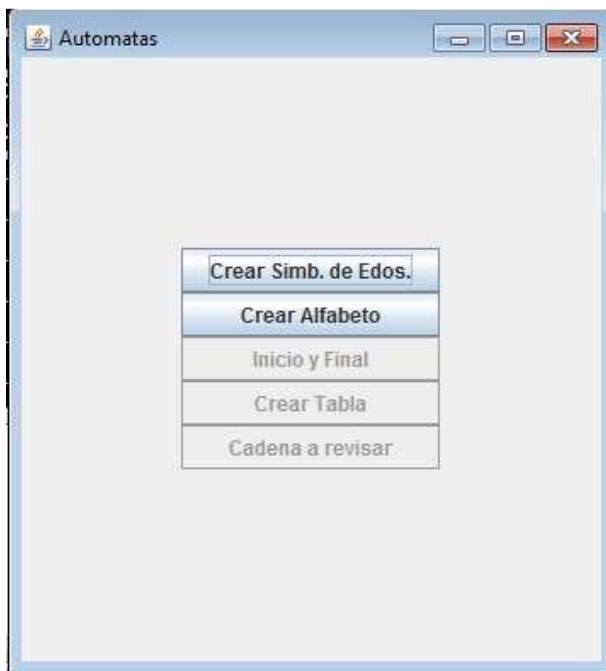
87

88

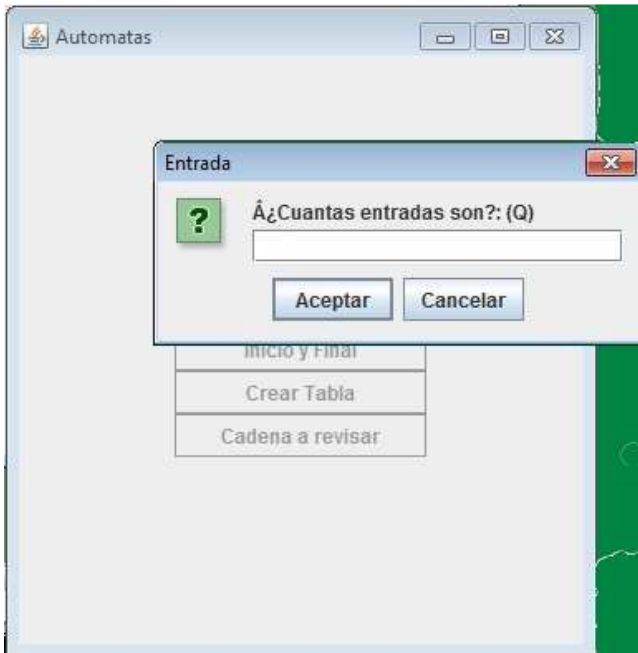
89 }

PRUEBAS Y DEPURACION

LA PRIMERA PRUEBA DE NUESTRO CODIGO ES UNA VERSION ACERCADA DEL DISEÑO PRINCIPAL, ESTA ES UNA VISTA PRELIMINAR DEL PROGRAMA:



COMO PRIMERA PRUEBA SE HAN CREADO LOS BOTONES RESPECTIVOS Y SOLO EL BOTON "CREAR SIMBOLOS DE ESTADOS" Y "CREAR ALFABETO" SON FUNCIONALES:



CONCLUSIONES ESPERADAS

CREEMOS FIRMEMENTE QUE UNA HERRAMIENTA COMO EL SOFTWARE QUE QUEREMOS REALIZAR SERIA DE MUCHA UTILIDAD PARA LOS ALUMNOS DE LA MATERIA DE TEORÍA DE LA COMPUTACIÓN. AL USAR GRÁFICOS EL APRENDIZAJE SE VUELVE MÁS SENCILLO, ES POR ESO QUE EL SIMULADOR DE AUTÓMATAS DEBE SER APROBADO.

GLOSARIO TECNICO

- LA TEORÍA DE LA COMPUTACIÓN: ES UNA CIENCIA, EN PARTICULAR UNA RAMA DE LA MATEMÁTICA Y DE LA COMPUTACIÓN QUE CENTRA SU INTERÉS EN EL ESTUDIO Y DEFINICIÓN FORMAL DE LOS CÓMPUTOS.
- AUTÓMATA: ES UNA MÁQUINA, UN MECANISMO ARTIFICIAL. PERO TAMBIÉN PUEDE SER UN INSTRUMENTO MUSICAL QUE TOCA CON AYUDA DE UN MECANISMO OCULTO.
- UN AUTÓMATA FINITO O MÁQUINA DE ESTADO FINITO :ES UN MODELO MATEMÁTICO DE UN SISTEMA QUE RECIBE UNA CADENA CONSTITUIDA POR SÍMBOLOS DE UN ALFABETO Y DETERMINA SI ESA CADENA PERTENECE AL LENGUAJE QUE EL AUTÓMATA RECONOCE.
- GRAFO :ES UN CONJUNTO DE OBJETOS LLAMADOS VÉRTICES O NODOS UNIDOS POR ENLACES LLAMADOS ARISTAS O ARCOS, QUE PERMITEN REPRESENTAR RELACIONES BINARIAS ENTRE ELEMENTOS DE UN CONJUNTO.
- AUTÓMATA PROGRAMABLE, O PLC (CONTROLADOR LÓGICO PROGRAMABLE): TODA MÁQUINA ELECTRÓNICA, DISEÑADA PARA CONTROLAR EN TIEMPO REAL Y EN MEDIO INDUSTRIAL PROCESOS SECUENCIALES. SU MANEJO Y PROGRAMACIÓN PUEDE SER REALIZADA POR PERSONAL ELÉCTRICO O ELECTRÓNICO SIN CONOCIMIENTOS INFORMÁTICOS. REALIZA FUNCIONES LÓGICAS: SERIES, PARALELOS, TEMPORIZACIONES, CONTAJES Y OTRAS MÁS POTENTES COMO CÁLCULOS, REGULACIONES, ETC.
- CADENAS: CONJUNTO DE CARACTERES ENLAZADOS ENTRE SI.
- EL LENGUAJE DE PROGRAMACIÓN ES EL MEDIO QUE UTILIZAN LOS PROGRAMADORES PARA CREAR UN PROGRAMA DE ORDENADOR; UN LENGUAJE DE MARCAS ES EL MEDIO PARA DESCRIBIR A UN

ORDENADOR EL FORMATO O LA ESTRUCTURA DE UN DOCUMENTO;
ETC.

BIBLIOGRAFIA

[HTTP://HTML.RINCONDELVAGO.COM/AUTOMATAS-PROGRAMABLES_2.HTML](http://HTML.RINCONDELVAGO.COM/AUTOMATAS-PROGRAMABLES_2.HTML)
[HTTP://ES.WIKIPEDIA.ORG/WIKI/DESARROLLO_EN_CASCADA](http://ES.WIKIPEDIA.ORG/WIKI/DESARROLLO_EN_CASCADA)