



# Instituto Politécnico Nacional Escuela Superior de Cómputo

## Teoría de la computación

Practica 4: Buscador de palabras  
Profesor: Juarez Martinez Genaro  
4CM6

Torres Abonce Luis Miguel  
Mayo 2022

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Marco Teórico</b>	<b>4</b>
<b>3</b>	<b>Desarrollo</b>	<b>5</b>
3.1	Conversion NFA-¿DFA . . . . .	5
3.2	Codigo . . . . .	7
3.3	Explicacion . . . . .	25
3.4	Capturas de los resultados . . . . .	27
<b>4</b>	<b>Conclusiones</b>	<b>29</b>
<b>5</b>	<b>Bibliografia</b>	<b>30</b>

# 1 Introducción

Descripción del problema:

El problema consiste en diseñar un autómata capaz de reconocer cada palabra reservada: web, webpage, website, webmaster, ebay, page, site, este se encontraran en un archivo de texto; además de indicar su posición así como visualizar el propio autómata ya transformado a su forma DFA (Graficarlo).

Se debe de diseñar el NFA y posteriormente transformarlo a su forma DFA para posteriormente programarlo.

En un archivo de texto deberá de guardar la evaluación del autómata por caracter de la cadena así como su cambio de estado (Historia del proceso).

En otro archivo de texto se debe enumerar, contar y anotar donde se encontraron cada una de las palabras reservadas.

## 2 Marco Teórico

DFA. Un autómata finito determinista es un tipo de algoritmo determinista basado en un estado que cambia con las entradas. También se conocen como máquinas de estados finitos deterministas o aceptadores finitos deterministas (Puede estar solo en 1 estado a la vez).

El DFA se compone por:

Q: conjunto finito de estados

: conjunto finito del símbolo de entrada

q0: estado inicial

F: estado final

: función de transición

NFA. Es un autómata finito que, a diferencia de los autómatas finitos deterministas, posee al menos un estado  $q \in Q$ , tal que para un símbolo  $a$  del alfabeto, existe más de una transición posible.

NFA también tiene cinco estados iguales que DFA, pero con una función de transición diferente, como se muestra a continuación:

:  $Q \times \Sigma \rightarrow 2^Q$  donde,

Q: conjunto finito de estados

: conjunto finito del símbolo de entrada

q0: estado inicial

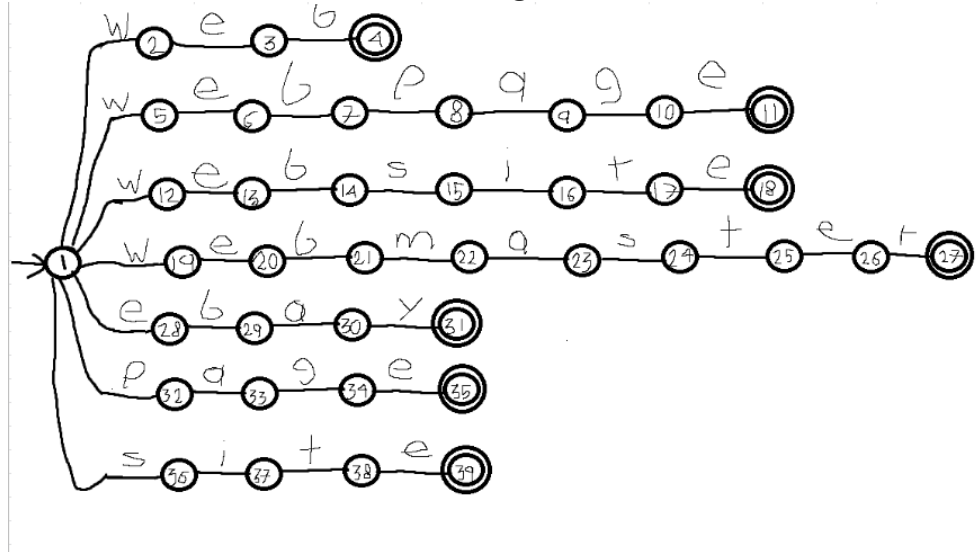
F: estado final

: función de transición

### 3 Desarrollo

#### 3.1 Conversion NFA-¿DFA

Primero diseñe el NFA de la siguiente manera:

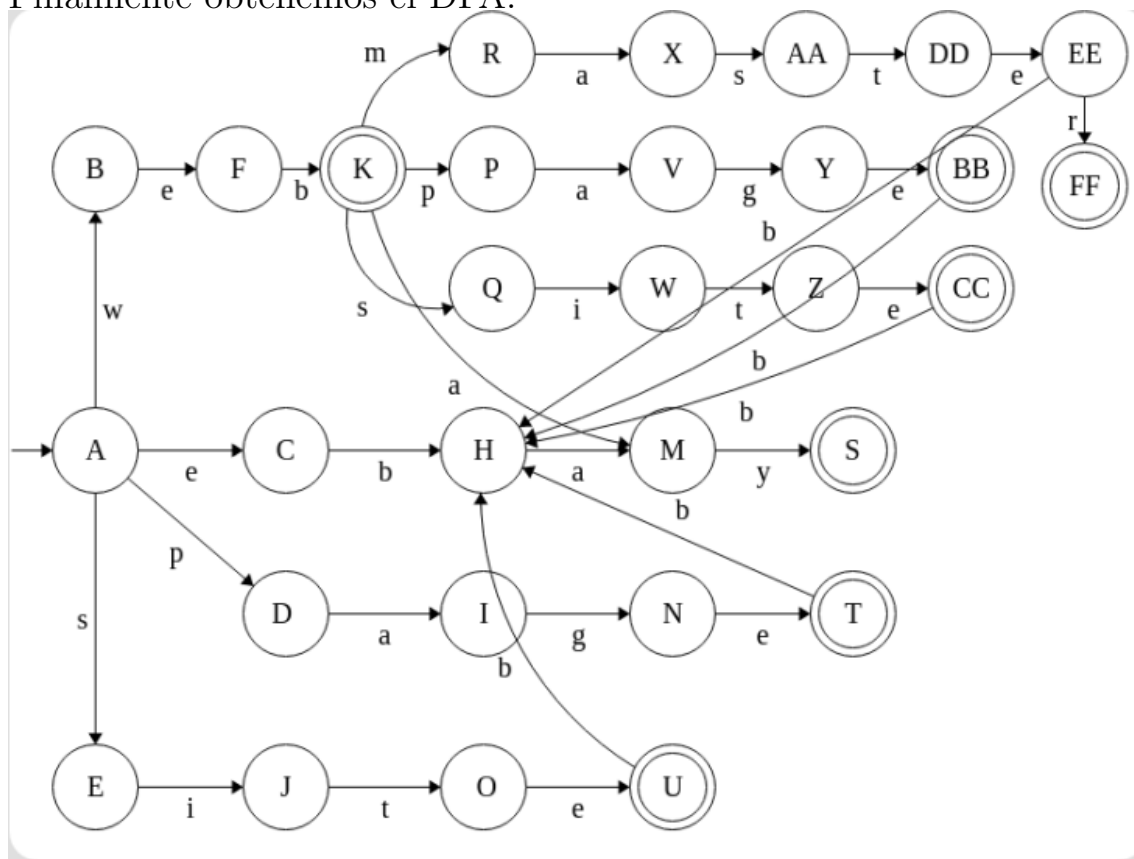


Posteriormente con la conversión de subconjuntos obtuve:

	w	e	a	t	e	r	y
1	2,5,12,19	28	32	36	37	38	39
2	2,5,12,19	28,3	32	36	37	38	39
3	2,5,12,19	28	4	36	37	38	39
4	2,5,12,19	28	32	36	37	38	39
5	2,5,12,19	28,6	32	36	37	38	39
6	2,5,12,19	28	7	36	37	38	39
7	2,5,12,19	28	32,8	36	37	38	39
8	2,5,12,19	28	32	9	36	37	39
9	2,5,12,19	28	32	10	36	37	39
10	2,5,12,19	28,11	32	36	37	38	39
11	2,5,12,19	28	32	36	37	38	39
12	2,5,12,19	28,13	32	36	37	38	39
13	2,5,12,19	28	14	36	37	38	39
14	2,5,12,19	28	32	36,15	37	38	39
15	2,5,12,19	28	32	36	16	37	39
16	2,5,12,19	28	32	36	17	37	39
17	2,5,12,19	28,18	32	36	37	38	39
18	2,5,12,19	28	32	36	37	38	39
19	2,5,12,19	28,20	32	36	37	38	39
20	2,5,12,19	28	21	36	37	38	39
21	2,5,12,19	28	32	36	37	22	39
22	2,5,12,19	28	32	23	36	37	39
23	2,5,12,19	28	32	36	36,24	37	39
24	2,5,12,19	28	32	36	36	25	39
25	2,5,12,19	28,26	32	36	37	38	39
26	2,5,12,19	28	32	36	37	27	39
27	2,5,12,19	28	32	36	37	38	39
28	2,5,12,19	28	29	36	37	38	39
29	2,5,12,19	28	32	30	36	37	39
30	2,5,12,19	28	32	36	37	38	31
31	2,5,12,19	28	32	36	37	38	39
32	2,5,12,19	28	32	33	36	37	39
33	2,5,12,19	28	32	34	36	37	39
34	2,5,12,19	28,35	32	36	37	38	39
35	2,5,12,19	28	32	36	37	38	39
36	2,5,12,19	28	32	36	37	38	39
37	2,5,12,19	28	32	36	37	38	39

		Letras											
Nombre	Estados	w	e	b	p	a	g	s	i	t	m	r	y
A	1	B	C	e	D	e	e	E	e	e	e	e	e
B	2,5,12,19	B	F	e	D	e	e	E	e	e	e	e	e
C	28	B	C	H	D	e	e	E	e	e	e	e	e
D	32	B	C	e	D	I	e	E	e	e	e	e	e
E	36	B	C	e	D	e	e	E	J	e	e	e	e
F	28,3,6,13,20	B	C	K	D	e	e	E	e	e	e	e	e
H	29	B	C	e	D	M	e	E	e	e	e	e	e
I	33	B	C	e	D	e	N	E	e	e	e	e	e
J	37	B	C	e	D	e	e	E	e	O	e	e	e
K	4,7,14,21,29	B	C	e	P	e	e	Q	e	e	R	e	e
M	30	B	C	e	D	e	e	E	e	e	e	e	S
N	34	B	T	e	D	e	e	E	e	e	e	e	e
O	38	B	U	e	D	e	e	E	e	e	e	e	e
P	32,8	B	C	e	D	V	e	E	e	e	e	e	e
Q	36,15	B	C	e	D	e	e	E	W	e	e	e	e
R	22	B	C	e	D	X	e	E	e	e	e	e	e
S	31	B	C	e	D	e	e	E	e	e	e	e	e
T	28,35	B	C	H	D	e	e	E	e	e	e	e	e
U	28,39	B	C	H	D	e	e	E	e	e	e	e	e
V	9,33	B	C	e	D	e	Y	E	e	e	e	e	e
W	16,37	B	C	e	D	e	e	E	e	Z	e	e	e
X	23	B	C	e	D	e	e	AA	e	e	e	e	e
Y	10,34	B	BB	e	D	e	e	E	e	e	e	e	e
Z	17,38	B	CC	e	D	e	e	E	e	e	e	e	e
AA	36,24	B	C	e	D	e	e	E	J	DD	e	e	e
BB	11,28,35	B	C	H	D	e	e	E	e	e	e	e	e
CC	18,28,39	B	C	H	D	e	e	E	e	e	e	e	e
DD	25	B	EE	e	D	e	e	E	e	e	e	e	e
EE	28,26	B	C	H	D	e	e	E	e	e	e	FF	e
FF	27	B	C	e	D	e	e	E	e	e	e	e	e

Finalmente obtenemos el DFA:



## 3.2 Codigo

```
from time import sleep
import turtle
print("1. Ver automata\n2. Correr Automata")
menu=int(input("Elige una opcion: "))
if menu == 1:
    tortu=turtle
    tortu.speed(0)
    tortu.setup(1350,950)
    tortu.penup()
    tortu.goto(-620,0)#A
    tortu.pendown()
    tortu.circle(40)
    tortu.penup()
    tortu.goto(-625,30)
    tortu.write("A",font=("courier",15, "bold"))
    tortu.penup()
    tortu.goto(-580,40)#A->D
    tortu.pendown()
    tortu.goto(-440,-190)#D
    tortu.penup()
    tortu.goto(-510,-150)
    tortu.write("p",font=("courier",15, "bold"))
    tortu.goto(-580,40)#A->C
    tortu.pendown()
    tortu.goto(-440,40)#C
    tortu.penup()
    tortu.goto(-510,0)
    tortu.write("e",font=("courier",15, "bold"))
    tortu.goto(-620,0)#A->E
    tortu.pendown()
    tortu.goto(-620,-380)#E
    tortu.penup()
    tortu.goto(-620,-190)
    tortu.write("s",font=("courier",15, "bold"))
    tortu.goto(-620,80)#A->B
    tortu.pendown()
    tortu.goto(-620,250)#B
    tortu.circle(40)
    tortu.penup()
    tortu.goto(-625,280)
    tortu.write("B",font=("courier",15, "bold"))
    tortu.goto(-630,170)
    tortu.write("w",font=("courier",15, "bold"))
    tortu.goto(-580,290)#B->F
    tortu.pendown()
    tortu.goto(-510,290)#F
    tortu.penup()
    tortu.goto(-545,290)
    tortu.write("e",font=("courier",15, "bold"))
    tortu.goto(-470,250)#Fz
    tortu.pendown()
    tortu.circle(40)
```

```

tortu.penup()
tortu.goto(-470,280)
tortu.write("F",font=("courier",15, "bold"))
tortu.goto(-430,290)#F->K
tortu.pendown()
tortu.goto(-360,290)#K
tortu.penup()
tortu.goto(-395,290)
tortu.write("b",font=("courier",15, "bold"))
tortu.goto(-320,250)#K
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-320,280)
tortu.write("K",font=("courier",15, "bold"))
tortu.goto(-320,330)#K->R
tortu.pendown()
tortu.goto(-210,420)#R
tortu.penup()
tortu.goto(-265,375)
tortu.write("m",font=("courier",15, "bold"))
tortu.goto(-320,250)#K->Q
tortu.pendown()
tortu.goto(-210,160)#Q
tortu.penup()
tortu.goto(-265,205)
tortu.write("s",font=("courier",15, "bold"))
tortu.goto(-280,290)#K->P
tortu.pendown()
tortu.goto(-210,290)#P
tortu.penup()
tortu.goto(-245,290)
tortu.write("p",font=("courier",15, "bold"))
tortu.goto(-170,250)#P
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-170,280)
tortu.write("P",font=("courier",15, "bold"))
tortu.goto(-130,290)#P->V
tortu.pendown()
tortu.goto(-60,290)#V
tortu.penup()
tortu.goto(-95,290)
tortu.write("a",font=("courier",15, "bold"))
tortu.goto(-20,250)#V
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-20,280)
tortu.write("V",font=("courier",15, "bold"))
tortu.goto(20,290)#V->Y
tortu.pendown()
tortu.goto(90,290)#Y
tortu.penup()
tortu.goto(55,290)
tortu.write("g",font=("courier",15, "bold"))
tortu.goto(130,250)#Y

```



```

tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(130,280)
tortu.write("Y",font=("courier",15, "bold"))
tortu.goto(170,290)#Y->BB
tortu.pendown()
tortu.goto(240,290)#BB
tortu.penup()
tortu.goto(205,290)
tortu.write("e",font=("courier",15, "bold"))
tortu.goto(280,250)#BB
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(280,280)
tortu.write("BB",font=("courier",15, "bold"))

tortu.penup()
tortu.goto(-170,380)#R
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-170,410)
tortu.write("R",font=("courier",15, "bold"))
tortu.goto(-130,420)#R->X
tortu.pendown()
tortu.goto(-60,420)
tortu.penup()
tortu.goto(-95,420)
tortu.write("a",font=("courier",15, "bold"))
tortu.goto(-20,380)#X
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-20,410)
tortu.write("X",font=("courier",15, "bold"))
tortu.goto(20,420)#X->AA
tortu.pendown()
tortu.goto(90,420)
tortu.penup()
tortu.goto(55,420)
tortu.write("s",font=("courier",15, "bold"))
tortu.goto(130,380)#AA
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(130,410)
tortu.write("AA",font=("courier",15, "bold"))
tortu.goto(170,420)#AA->DD
tortu.pendown()
tortu.goto(240,420)
tortu.penup()
tortu.goto(205,420)
tortu.write("t",font=("courier",15, "bold"))
tortu.goto(280,380)#DD
tortu.pendown()
tortu.circle(40)

```

```

tortu.penup()
tortu.goto(280,410)
tortu.write("DD",font=("courier",15, "bold"))
tortu.goto(320,420)#DD->EE
tortu.pendown()
tortu.goto(390,420)
tortu.penup()
tortu.goto(355,420)
tortu.write("e",font=("courier",15, "bold"))
tortu.goto(430,380)#EE
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(430,410)
tortu.write("EE",font=("courier",15, "bold"))
tortu.goto(430,380)#EE->FF
tortu.pendown()
tortu.goto(430,330)#FF
tortu.penup()
tortu.goto(430,355)
tortu.write("r",font=("courier",15, "bold"))
tortu.goto(430,250)#FF
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(430,280)
tortu.write("FF",font=("courier",15, "bold"))

tortu.penup()
tortu.goto(-170,120)#Q
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-170,150)
tortu.write("Q",font=("courier",15, "bold"))
tortu.goto(-130,160)#Q->W
tortu.pendown()
tortu.goto(-60,160)#W
tortu.penup()
tortu.goto(-95,160)
tortu.write("i",font=("courier",15, "bold"))
tortu.goto(-20,120)#W
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-20,150)
tortu.write("W",font=("courier",15, "bold"))
tortu.goto(20,160)#W->Z
tortu.pendown()
tortu.goto(90,160)
tortu.penup()
tortu.goto(55,160)
tortu.write("t",font=("courier",15, "bold"))
tortu.goto(130,120)#Z
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(130,150)

```

```

tortu.write("Z",font=("courier",15, "bold"))
tortu.goto(170,160)#Z->CC
tortu.pendown()
tortu.goto(240,160)
tortu.penup()
tortu.goto(205,160)
tortu.write("e",font=("courier",15, "bold"))
tortu.goto(280,120)#CC
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(280,150)
tortu.write("CC",font=("courier",15, "bold"))

tortu.penup()
tortu.goto(-400,0)#C
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-400,30)
tortu.write("C",font=("courier",15, "bold"))
tortu.goto(-360,40)#C->H
tortu.pendown()
tortu.goto(-210,40)
tortu.penup()
tortu.goto(-285,40)
tortu.write("b",font=("courier",15, "bold"))
tortu.goto(-170,0)#H
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-170,30)
tortu.write("H",font=("courier",15, "bold"))
tortu.goto(-130,40)#H>M
tortu.pendown()
tortu.goto(90,40)
tortu.penup()
tortu.goto(-20,40)
tortu.write("a",font=("courier",15, "bold"))
tortu.goto(130,0)#M
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(130,30)
tortu.write("M",font=("courier",15, "bold"))
tortu.goto(170,40)#M->S
tortu.pendown()
tortu.goto(390,40)
tortu.penup()
tortu.goto(280,40)
tortu.write("y",font=("courier",15, "bold"))
tortu.goto(430,0)#S
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(430,30)
tortu.write("S",font=("courier",15, "bold"))

```

```

tortu.penup()
tortu.goto(-400,-230)#D
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-400,-200)
tortu.write("D",font=("courier",15, "bold"))
tortu.goto(-360,-190)#D->I
tortu.pendown()
tortu.goto(-210,-190)
tortu.penup()
tortu.goto(-285,-190)
tortu.write("a",font=("courier",15, "bold"))
tortu.goto(-170,-230)#I
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-170,-200)
tortu.write("I",font=("courier",15, "bold"))
tortu.goto(-130,-190)#I->N
tortu.pendown()
tortu.goto(90,-190)
tortu.penup()
tortu.goto(-20,-190)
tortu.write("g",font=("courier",15, "bold"))
tortu.goto(130,-230)#N
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(130,-200)
tortu.write("N",font=("courier",15, "bold"))
tortu.goto(170,-190)#N->T
tortu.pendown()
tortu.goto(390,-190)
tortu.penup()
tortu.goto(280,-190)
tortu.write("e",font=("courier",15, "bold"))
tortu.goto(430,-230)#T
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(430,-200)
tortu.write("T",font=("courier",15, "bold"))

tortu.goto(-620,-460)#E
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-620,-430)
tortu.write("E",font=("courier",15, "bold"))
tortu.goto(-580,-420)#E->J
tortu.pendown()
tortu.goto(-440,-420)
tortu.penup()
tortu.goto(-510,-420)
tortu.write("i",font=("courier",15, "bold"))
tortu.goto(-400,-460)#J
tortu.pendown()

```

```

tortu.circle(40)
tortu.penup()
tortu.goto(-400,-430)
tortu.write("J",font=("courier",15, "bold"))
tortu.goto(-360,-420)#J->0
tortu.pendown()
tortu.goto(-210,-420)
tortu.penup()
tortu.goto(-285,-420)
tortu.write("t",font=("courier",15, "bold"))
tortu.goto(-170,-460)#0
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(-170,-430)
tortu.write("0",font=("courier",15, "bold"))
tortu.goto(-130,-420)#0->U
tortu.pendown()
tortu.goto(90,-420)
tortu.penup()
tortu.goto(-20,-420)
tortu.write("e",font=("courier",15, "bold"))
tortu.goto(130,-460)#U
tortu.pendown()
tortu.circle(40)
tortu.penup()
tortu.goto(130,-430)
tortu.write("U",font=("courier",15, "bold"))
sleep(3)
else:
    lectura=open("lectura.txt","r")#abrir documento
    historia=open("historia.txt","w")
    encontradas=open("encontradas.txt","w")
    #Listas donde guardaremos la ubicacion donde se encontraron
    listWeb=[]
    listWeb2=[]
    listWebPage=[]
    listWebPage2=[]
    listWebSite=[]
    listWebSite2=[]
    listWebMaster=[]
    listWebMaster2=[]
    listEbay=[]
    listEbay2=[]
    listPage=[]
    listPage2=[]
    listSite=[]
    listSite2=[]
    i=0 #eje x
    k=0 #eje y
    estado="A"#Declarar estado inicial
    while 1:
        caracter=lectura.read(1)#recorrer caracter por caracter el documento
        historia.write("(" +caracter+", "+estado)
        if estado == "A":
            if caracter == "w":
                estado="B"
                historia.write("->B)\n")

```

```

elif caracter == "e":
    estado="C"
    historia.write("->C)\n")
elif caracter == "p":
    estado="D"
    historia.write("->D)\n")
elif caracter == "s":
    estado = "E"
    historia.write("->E)\n")
else:
    estado="A"
    historia.write("->A)\n")
elif estado == "B":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "e":
        estado="F"
        historia.write("->F)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "C":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "H"
        historia.write("->H)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "D":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":

```

```

        estado = "E"
        historia.write("->E)\n")
    elif caracter == "a":
        estado = "I"
        historia.write("->I)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "E":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "i":
        estado = "J"
        historia.write("->A)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "F":
    if caracter == "w":
        estado = "B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado = "C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado = "D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "K"
        historia.write("->K)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "H":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"

```

```

        historia.write("->E)\n")
    elif caracter == "a":
        estado = "M"
        historia.write("->M)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "I":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "g":
        estado = "N"
        historia.write("->N)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "J":
    if caracter == "w":
        estado = "B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado = "C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "t":
        estado = "Q"
        historia.write("->Q)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "K":#Estado final
    listWeb.append(i-3)#guardar eje x
    listWeb2.append(k)#guardar eje y
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "m":
        estado = "R"
        historia.write("->R)\n")
    elif caracter == "p":

```



```

        estado = "P"
        historia.write("->P)\n")
    elif caracter == "s":
        estado = "Q"
        historia.write("->Q)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "M":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "y":
        estado = "S"
        historia.write("->S*)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "N":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "e":
        estado = "T"
        historia.write("->T*)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "O":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "e":
        estado = "U"
        historia.write("->U*)\n")
    else:
        estado="A"

```

```

        historia.write("->A)\n")
elif estado == "P":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "a":
        estado = "V"
        historia.write("->V)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "Q":
    if caracter == "w":
        estado="B"
        historia.write("->Q)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "i":
        estado = "W"
        historia.write("->W)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "R":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "a":
        estado = "X"
        historia.write("->X)\n")
    else:
        estado="A"
        historia.write("->A)\n")

```

```

elif estado == "S":#Estado Final
    listEbay.append(i-4)
    listEbay2.append(k)
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "T":#Estado Final
    listPage.append(i-4)
    listPage2.append(k)
    if caracter == "w":
        estado="B"
        historia.write("->W)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "H"
        historia.write("->H)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "U":#Estado Final
    listSite.append(i-4)
    listSite2.append(k)
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "H"
        historia.write("->H)\n")
    else:

```

```

        estado="H"
        historia.write("->H)\n")
elif estado == "V":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "g":
        estado = "Y"
        historia.write("->Y)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "W":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "t":
        estado = "Z"
        historia.write("->Z)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "X":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "AA"
        historia.write("->AA)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "Y":
    if caracter == "w":

```

```

        estado="B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "e":
        estado = "BB"
        historia.write("->BB*)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "Z":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "e":
        estado = "CC"
        historia.write("->CC*)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "AA":
    if caracter == "w":
        estado="B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado="C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado="D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "t":
        estado = "DD"
        historia.write("->DD)\n")
    elif caracter == "i":
        estado = "J"
        historia.write("->J)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "BB":#Estado Final
    listWebPage.append(i-7)
    listWebPage2.append(k)
    listPage.append(i-4)
    listPage2.append(k)
    if caracter == "w":

```

```

        estado = "B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado = "C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado = "D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "H"
        historia.write("->H)\n")
    else:
        estado = "A"
        historia.write("->A)\n")
elif estado == "CC":#Estado Final
    listWebSite.append(i-7)
    listWebSite2.append(k)
    listSite.append(i-4)
    listSite2.append(k)
    if caracter == "w":
        estado = "B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado = "C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado = "D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "H"
        historia.write("->H)\n")
    else:
        estado = "A"
        historia.write("->A)\n")
elif estado == "DD":
    if caracter == "w":
        estado = "B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado = "D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "e":
        estado = "EE"
        historia.write("->EE)\n")
    else:
        estado = "A"
        historia.write("->A)\n")
elif estado == "EE":
    if caracter == "w":

```

```

        estado = "B"
        historia.write("->B)\n")
    elif caracter == "p":
        estado = "D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    elif caracter == "b":
        estado = "H"
        historia.write("->H)\n")
    elif caracter == "r":
        estado = "FF"
        historia.write("->FF*)\n")
    else:
        estado="A"
        historia.write("->A)\n")
elif estado == "FF":#Estado Final
    listWebMaster.append(i-9)
    listWebMaster2.append(k)
    if caracter == "w":
        estado = "B"
        historia.write("->B)\n")
    elif caracter == "e":
        estado = "C"
        historia.write("->C)\n")
    elif caracter == "p":
        estado = "D"
        historia.write("->D)\n")
    elif caracter == "s":
        estado = "E"
        historia.write("->E)\n")
    else:
        estado = "A"
        historia.write("->A)\n")

if not caracter:#Si no hay es caracter es el fin
    break
i+=1#Llevar la cuenta de caracteres eje x
if caracter == "\n":
    k+=1
    i=0
encontradas.write("Palabra web encontradas en el archivo y su ubicacion: \n")
j=0
for i,k in zip(listWeb,listWeb2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+", "+str(i)+"\n")
encontradas.write("\nPalabra webPage encontradas en el archivo y su ubicacion: \n")
j=0
for i,k in zip(listWebPage,listWebPage2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+", "+str(i)+"\n")
encontradas.write("\nPalabra WebSite encontradas en el archivo y su ubicacion: \n")
j=0
for i,k in zip(listWebSite,listWebSite2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+", "+str(i)+"\n")
encontradas.write("\nPalabra WebMaster encontradas en el archivo y su ubicacion: \n")

```

```

j=0
for i,k in zip(listWebMaster,listWebMaster2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+","+str(i)+"\n")
encontradas.write("\nPalabra Ebay encontradas en el archivo y su ubicacion: \n")
j=0
for i,k in zip(listEbay,listEbay2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+","+str(i)+"\n")
encontradas.write("\nPalabra Page encontradas en el archivo y su ubicacion: \n")
j=0
for i,k in zip(listPage,listPage2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+","+str(i)+"\n")
encontradas.write("\nPalabra Site encontradas en el archivo y su ubicacion: \n")
j=0
for i,k in zip(listSite,listSite2):
    j+=1
    encontradas.write(str(j)+". "+str(k)+","+str(i)+"\n")

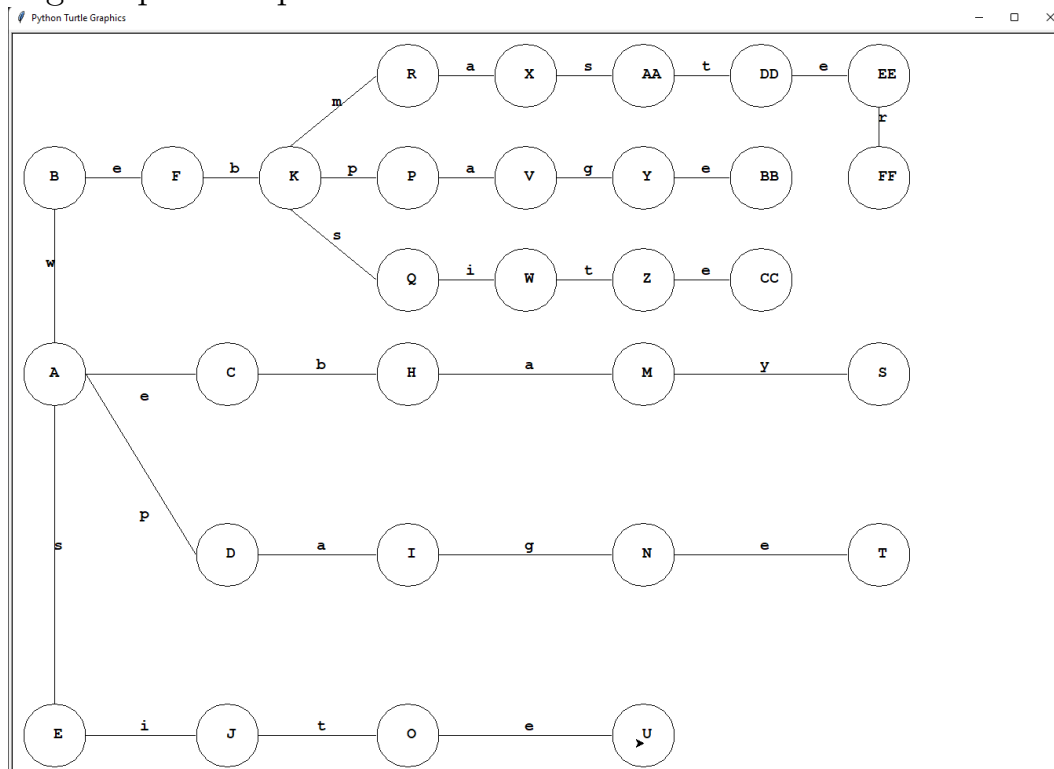
lectura.close()
encontradas.close()
historia.close()
print("Se han encontrada todas las palabras, se encuentran en el archivo de texto encontradas.txt, la historia en el archivo histor

```



### 3.3 Explicacion

Este programa se programa en el lenguaje de pyhthon, lo primero que realiza el programa es preguntar si se desea ver el autómatata o correr el autómatata en caso de que se eliga la primer opción se mostrara el autómatata:



Este se realizo con la librería turtle, para animar el autómatata primeramente dibuje todos los círculos en el lugar que estarían posteriormente su nombre de estado, después todas las líneas que conectan a los estados y finalmente la cadena que recibirían para hacer el cambio de estado para esto tome las coordenadas de los estados los sume y divide entre 2 para que quedaran centradas. Esto es básicamente la primera parte del programa.

La segunda parte del programa corre el buscador de palabras; lo primero que realiza es abrir los documentos donde se guardara la historia, las palabras encontradas y donde las buscara, posteriormente todas las listas para guardar la ubicación de las palabras encontradas tanto en el eje x como y, posteriormente con un ciclo while "Infinito"

donde terminara cuando ya no se puedan leer mas caracteres del documento, cada if representa un estado si, por ejemplo, entra al estado "K"(que es un estado final), verificara que carácter esta leyendo de la cadena y dependiendo de la opción escribirá en el archivo de historia hacia que estado va y cambia de estado, de esta manera recorrerá todo el documento y encontrando todas las palabras. Para comprobar que el documento ya no tiene ningún carácter solo utilizo una condicional "if not character" rompe el ciclo while, además cuando se tiene un salto de linea se reinicia el eje x y se aumenta una unidad al eje y.

Por ultimo se escriben todas las palabras encontradas así como su ubicación de cada una de ellas. Se cierran los archivos que se utilizaron y se imprime un mensaje de finalización.

### 3.4 Capturas de los resultados

Palabras a encontrar:

```
≡ lectura.txt
1 web web web asf weasfweb website webpage webmaster ebay page site site ajfkjsf websitepage
2
3
4 webmaster
```

Palabras encontradas:

```
≡ encontradas.txt
1 Palabra web encontradas en el archivo y su ubicacion:
2 1. 0,0
3 2. 0,9
4 3. 0,23
5 4. 0,27
6 5. 0,35
7 6. 0,43
8 7. 0,83
9 8. 3,0
10
11 Palabra webPage encontradas en el archivo y su ubicacion:
12 1. 0,35
13
14 Palabra WebSite encontradas en el archivo y su ubicacion:
15 1. 0,27
16 2. 0,83
17
18 Palabra WebMaster encontradas en el archivo y su ubicacion:
19 1. 0,43
20 2. 3,0
21
22 Palabra Ebay encontradas en el archivo y su ubicacion:
23 1. 0,53
24
25 Palabra Page encontradas en el archivo y su ubicacion:
26 1. 0,38
27 2. 0,59
28 3. 0,90
29
30 Palabra Site encontradas en el archivo y su ubicacion:
31 1. 0,30
32 2. 0,65
33 3. 0,70
34 4. 0,86
35
```

Historia del proceso:

```
≡ historia.txt
1  (w,A->B)
2  (e,B->F)
3  (b,F->K)
4  ( ,K->A)
5  (w,A->B)
6  (e,B->F)
7  (h,F->A)
8  ( ,A->A)
9  ( ,A->A)
10 (w,A->B)
11 (e,B->F)
12 (b,F->K)
13 ( ,K->A)
14 ( ,A->A)
15 (a,A->A)
16 (s,A->E)
17 (f,E->A)
18 ( ,A->A)
19 (w,A->B)
20 (e,B->F)
21 (a,F->A)
22 (s,A->E)
23 (f,E->A)
24 (w,A->B)
25 (e,B->F)
26 (b,F->K)
27 ( ,K->A)
28 (w,A->B)
29 (e,B->F)
30 (b,F->K)
31 (s,K->Q)
32 (i,Q->W)
33 (t,W->Z)
34 (e,Z->CC*)
35 ( ,CC->A)
36 (w,A->B)
37 (e,B->F)
38 (b,F->K)
39 (p,K->P)
```

## 4 Conclusiones

El programa fue bastante sencillo de realizar ya que siguiendo la transformación del NFA a DFA se puede realizar con unas condicionales, la dificultad se concentro en la misma transformación ya que eran bastante letras y se iba haciendo mas y mas grande la tabla, pero con organización en el excel marcando con colores los conjuntos de estado así como estados finales se me hizo mas sencillo realizarlo.

La graficación como siempre es mas tardada ya que tenemos que estar atento a todas las coordenadas de los estados.

## 5 Bibliografia

1. libro "Introduction to Automata Theory, languages, and computation", John E. Hopcroft, 2nd edition
2. DFA — Deterministic Finite Automata - Javatpoint. (s. f.). Wwww.Javatpoint.Com. Recuperado 21 de mayo de 2022, de <https://www.javatpoint.com/deterministic-finite-automata>
3. NFA — Non-Deterministic Finite Automata - Javatpoint. (s. f.). Wwww.Javatpoint.Com. Recuperado 21 de mayo de 2022, de <https://www.javatpoint.com/non-deterministic-finite-automata>