

REPORTE

Programa 4 - buscador de palabras

Materia : Teoría de la Computación

Grupo : 4CM1

Alumno : Julio Cesar Hernández Reyes

Docente : Juárez Martínez Genaro

1. Introducción

En este reporte se explicara como se realizo el programa 4 - buscador de palabras, el cual es un programa que identifica 7 palabras reservadas, dentro de un archivo.txt, una vez que se encuentre alguna de las palabras reservadas la ubicación de la palabra encontrada(linea, columna) se guardara en otro archivo.

Para poder identificar las palabras en el texto, el programa usa un autómata finito determinista el cual identifica cada una de las 7 palabras reservadas. Las palabras que el autómata reconoce son: coin, ebay, home, master, page, site y web. Al final se tendrá un archivo que contendrá la cantidad de palabras encontradas, que palabras se encontraron, y la ubicación de estas. También se tendrá un archivo con toda la historia del autómata con lo que se leyó en el archivo original.

Tendrá un menú el cual se podrá: Elegir un archivo de texto, Ingresar una cadena, Ver el grafo del programa, Salir del programa

Resumen del programa:

Entrada: Puede ser un archivo de texto o una simple cadena de caracteres

Salida: Dos archivos de texto:

1-palabrasreservadas

2-archivohistoria

Para la realización del programa se uso Python. Se uso un IDE en vez de compilar y ejecutar por consola, esto para una mayor facilidad a la hora de corregir errores y algunos detalles del programa.



PyCharm Community Edition
2021.2.1
Aplicación

2. Desarrollo

2.1. Código del programa

Código creado en Python en el IDE de Pycharm:

```
1  # Programa 4 Buscador de palabras
2  # Palabras a encontrar:
3  # coin eBay home master page site web
4  # Web para hacer los textos
5  # https://www.loremipzum.com/es/generador-de-texto
6
7  import os
8  import cv2
9  import matplotlib.pyplot as plt
10
11 def automata(cadena, linea):
12     largo = len(cadena)
13     a = 1
14     b = 2
15     c = 6
16     d = 10
17     e = 14
18     f = 20
19     g = 24
20     h = 28
21     i = 3
22     j = 7
23     k = 11
24     l = 15
25     m = 21
26     n = 25
27     enie = 29
28     o = 4
29     p = 8
30     q = 12
31     r = 16
32     s = 22
33     t = 26
34     u = 30
35     v = 5
36     w = 9
37     x = 13
38     y = 17
39     z = 23
```

```

40 _a_ = 27
41 _b_ = 18
42 _c_ = 19
43 actual = a
44 conta = 0
45
46 columna = 0
47 while True:
48     columna = columna + 1
49     #print(actual)
50     if conta >= largo:
51         break
52     # Estado A
53     if actual == a:
54         if cadena[conta] == 'c':
55             historia('B', 'A', 'c')
56             actual = b
57         elif cadena[conta] == 'e':
58             historia('C', 'A', 'e')
59             actual = c
60         elif cadena[conta] == 'h':
61             historia('D', 'A', 'h')
62             actual = d
63         elif cadena[conta] == 'm':
64             historia('E', 'A', 'm')
65             actual = e
66         elif cadena[conta] == 'p':
67             historia('F', 'A', 'p')
68             actual = f
69         elif cadena[conta] == 's':
70             historia('G', 'A', 's')
71             actual = g
72         elif cadena[conta] == 'w':
73             historia('H', 'A', 'w')
74             actual = h
75         else:
76             historia('A', 'A', cadena[conta])
77             actual = a
78     # Estado B
79     elif actual == b:
80         if cadena[conta] == 'o':
81             historia('I', 'B', 'o')
82             actual = i
83         # -----
84         elif cadena[conta] == 'c':

```

```

85         historia('B', 'B', 'c')
86         actual = b
87     elif cadena[conta] == 'e':
88         historia('C', 'B', 'e')
89         actual = c
90     elif cadena[conta] == 'h':
91         historia('D', 'B', 'h')
92         actual = d
93     elif cadena[conta] == 'm':
94         historia('E', 'B', 'm')
95         actual = e
96     elif cadena[conta] == 'p':
97         historia('F', 'B', 'p')
98         actual = f
99     elif cadena[conta] == 's':
100        historia('G', 'B', 's')
101        actual = g
102    elif cadena[conta] == 'w':
103        historia('H', 'B', 'w')
104        actual = h
105    else:
106        historia('A', 'B', cadena[conta])
107        actual = a
108    # Estado C
109    elif actual == c:
110        if cadena[conta] == 'b':
111            historia('J', 'C', 'b')
112            actual = j
113        # -----
114        elif cadena[conta] == 'c':
115            historia('B', 'C', 'c')
116            actual = b
117        elif cadena[conta] == 'e':
118            historia('C', 'C', 'e')
119            actual = c
120        elif cadena[conta] == 'h':
121            historia('D', 'C', 'h')
122            actual = d
123        elif cadena[conta] == 'm':
124            historia('E', 'C', 'm')
125            actual = e
126        elif cadena[conta] == 'p':
127            historia('F', 'C', 'p')
128            actual = f
129        elif cadena[conta] == 's':

```

```

130         historia('G', 'C', 's')
131         actual = g
132     elif cadena[conta] == 'w':
133         historia('H', 'C', 'w')
134         actual = h
135     else:
136         historia('A', 'C', cadena[conta])
137         actual = a
138 # Estado D
139 elif actual == d:
140     if cadena[conta] == 'o':
141         historia('K', 'D', 'o')
142         actual = k
143     # -----
144     elif cadena[conta] == 'c':
145         historia('B', 'D', 'c')
146         actual = b
147     elif cadena[conta] == 'e':
148         historia('C', 'D', 'e')
149         actual = c
150     elif cadena[conta] == 'h':
151         historia('D', 'D', 'o')
152         actual = d
153     elif cadena[conta] == 'm':
154         historia('E', 'D', 'o')
155         actual = e
156     elif cadena[conta] == 'p':
157         historia('F', 'D', 'o')
158         actual = f
159     elif cadena[conta] == 's':
160         historia('G', 'D', 'o')
161         actual = g
162     elif cadena[conta] == 'w':
163         historia('H', 'D', 'o')
164         actual = h
165     else:
166         historia('A', 'D', cadena[conta])
167         actual = a
168 # Estado E
169 elif actual == e:
170     if cadena[conta] == 'a':
171         historia('L', 'E', 'a')
172         actual = l
173     # -----
174     elif cadena[conta] == 'c':

```

```

175         historia('B', 'E', 'c')
176         actual = b
177     elif cadena[conta] == 'e':
178         historia('C', 'E', 'e')
179         actual = c
180     elif cadena[conta] == 'h':
181         historia('D', 'E', 'h')
182         actual = d
183     elif cadena[conta] == 'm':
184         historia('E', 'E', 'm')
185         actual = e
186     elif cadena[conta] == 'p':
187         historia('F', 'E', 'p')
188         actual = f
189     elif cadena[conta] == 's':
190         historia('G', 'E', 's')
191         actual = g
192     elif cadena[conta] == 'w':
193         historia('H', 'E', 'w')
194         actual = h
195     else:
196         historia('A', 'E', cadena[conta])
197         actual = a
198 # Estado F
199 elif actual == f:
200     if cadena[conta] == 'a':
201         historia('M', 'F', 'a')
202         actual = m
203     # -----
204     elif cadena[conta] == 'c':
205         historia('B', 'F', 'c')
206         actual = b
207     elif cadena[conta] == 'e':
208         historia('C', 'F', 'e')
209         actual = c
210     elif cadena[conta] == 'h':
211         historia('D', 'F', 'h')
212         actual = d
213     elif cadena[conta] == 'm':
214         historia('E', 'F', 'm')
215         actual = e
216     elif cadena[conta] == 'p':
217         historia('F', 'F', 'p')
218         actual = f
219     elif cadena[conta] == 's':

```

```

220         historia('G', 'F', 's')
221         actual = g
222     elif cadena[conta] == 'w':
223         historia('H', 'F', 'w')
224         actual = h
225     else:
226         historia('A', 'F', cadena[conta])
227         actual = a
228 # Estado G
229 elif actual == g:
230     if cadena[conta] == 'i':
231         historia('N', 'G', 'i')
232         actual = n
233     # -----
234     elif cadena[conta] == 'c':
235         historia('B', 'G', 'c')
236         actual = b
237     elif cadena[conta] == 'e':
238         historia('C', 'G', 'e')
239         actual = c
240     elif cadena[conta] == 'h':
241         historia('D', 'G', 'h')
242         actual = d
243     elif cadena[conta] == 'm':
244         historia('E', 'G', 'm')
245         actual = e
246     elif cadena[conta] == 'p':
247         historia('F', 'G', 'p')
248         actual = f
249     elif cadena[conta] == 's':
250         historia('G', 'G', 's')
251         actual = g
252     elif cadena[conta] == 'w':
253         historia('H', 'G', 'w')
254         actual = h
255     else:
256         historia('A', 'G', cadena[conta])
257         actual = a
258 # Estado H
259 elif actual == h:
260     if cadena[conta] == 'e':
261         historia('enie', 'H', 'e')
262         actual = enie
263     # -----
264     elif cadena[conta] == 'c':

```

```

265         historia('B', 'H', 'c')
266         actual = b
267     elif cadena[conta] == 'e':
268         historia('C', 'H', 'e')
269         actual = c
270     elif cadena[conta] == 'h':
271         historia('D', 'H', 'h')
272         actual = d
273     elif cadena[conta] == 'm':
274         historia('E', 'H', 'm')
275         actual = e
276     elif cadena[conta] == 'p':
277         historia('F', 'H', 'p')
278         actual = f
279     elif cadena[conta] == 's':
280         historia('G', 'H', 's')
281         actual = g
282     elif cadena[conta] == 'w':
283         historia('H', 'H', 'w')
284         actual = h
285     else:
286         historia('A', 'H', cadena[conta])
287         actual = a
288 # Estado I
289 elif actual == i:
290     if cadena[conta] == 'i':
291         historia('O', 'I', 'i')
292         actual = o
293     # -----
294     elif cadena[conta] == 'c':
295         historia('B', 'I', 'c')
296         actual = b
297     elif cadena[conta] == 'e':
298         historia('C', 'I', 'e')
299         actual = c
300     elif cadena[conta] == 'h':
301         historia('D', 'I', 'h')
302         actual = d
303     elif cadena[conta] == 'm':
304         historia('E', 'I', 'm')
305         actual = e
306     elif cadena[conta] == 'p':
307         historia('F', 'I', 'p')
308         actual = f
309     elif cadena[conta] == 's':

```



```

310         historia('G', 'I', 's')
311         actual = g
312     elif cadena[conta] == 'w':
313         historia('H', 'I', 'w')
314         actual = h
315     else:
316         historia('A', 'I', cadena[conta])
317         actual = a
318 # Estado J
319 elif actual == j:
320     if cadena[conta] == 'a':
321         historia('P', 'J', 'a')
322         actual = p
323     # -----
324     elif cadena[conta] == 'c':
325         historia('B', 'J', 'c')
326         actual = b
327     elif cadena[conta] == 'e':
328         historia('C', 'J', 'e')
329         actual = c
330     elif cadena[conta] == 'h':
331         historia('D', 'J', 'h')
332         actual = d
333     elif cadena[conta] == 'm':
334         historia('E', 'J', 'm')
335         actual = e
336     elif cadena[conta] == 'p':
337         historia('F', 'J', 'p')
338         actual = f
339     elif cadena[conta] == 's':
340         historia('G', 'J', 's')
341         actual = g
342     elif cadena[conta] == 'w':
343         historia('H', 'J', 'w')
344         actual = h
345     else:
346         historia('A', 'J', cadena[conta])
347         actual = a
348 # Estado K
349 elif actual == k:
350     if cadena[conta] == 'm':
351         historia('Q', 'K', 'm')
352         actual = q
353     # -----
354     elif cadena[conta] == 'c':

```

```

355         historia('B', 'K', 'c')
356         actual = b
357     elif cadena[conta] == 'e':
358         historia('C', 'K', 'e')
359         actual = c
360     elif cadena[conta] == 'h':
361         historia('D', 'K', 'h')
362         actual = d
363     elif cadena[conta] == 'm':
364         historia('E', 'K', 'm')
365         actual = e
366     elif cadena[conta] == 'p':
367         historia('F', 'K', 'p')
368         actual = f
369     elif cadena[conta] == 's':
370         historia('G', 'K', 's')
371         actual = g
372     elif cadena[conta] == 'w':
373         historia('H', 'K', 'w')
374         actual = h
375     else:
376         historia('A', 'K', cadena[conta])
377         actual = a
378 # Estado L
379 elif actual == l:
380     if cadena[conta] == 's':
381         historia('R', 'L', 's')
382         actual = r
383     # -----
384     elif cadena[conta] == 'c':
385         historia('B', 'L', 'c')
386         actual = b
387     elif cadena[conta] == 'e':
388         historia('C', 'L', 'e')
389         actual = c
390     elif cadena[conta] == 'h':
391         historia('D', 'L', 'h')
392         actual = d
393     elif cadena[conta] == 'm':
394         historia('E', 'L', 'm')
395         actual = e
396     elif cadena[conta] == 'p':
397         historia('F', 'L', 'p')
398         actual = f
399     elif cadena[conta] == 's':

```

```

400         historia('G', 'L', 's')
401         actual = g
402     elif cadena[conta] == 'w':
403         historia('H', 'L', 'w')
404         actual = h
405     else:
406         historia('A', 'L', cadena[conta])
407         actual = a
408 # Estado M
409 elif actual == m:
410     if cadena[conta] == 'g':
411         historia('S', 'M', 'g')
412         actual = s
413     # -----
414     elif cadena[conta] == 'c':
415         historia('B', 'M', 'c')
416         actual = b
417     elif cadena[conta] == 'e':
418         historia('C', 'M', 'e')
419         actual = c
420     elif cadena[conta] == 'h':
421         historia('D', 'M', 'h')
422         actual = d
423     elif cadena[conta] == 'm':
424         historia('E', 'M', 'm')
425         actual = e
426     elif cadena[conta] == 'p':
427         historia('F', 'M', 'p')
428         actual = f
429     elif cadena[conta] == 's':
430         historia('G', 'M', 's')
431         actual = g
432     elif cadena[conta] == 'w':
433         historia('H', 'M', 'w')
434         actual = h
435     else:
436         historia('A', 'M', cadena[conta])
437         actual = a
438 # Estado N
439 elif actual == n:
440     if cadena[conta] == 't':
441         historia('T', 'N', 't')
442         actual = t
443     # -----
444     elif cadena[conta] == 'c':

```

```

445         historia('B', 'N', 'c')
446         actual = b
447     elif cadena[conta] == 'e':
448         historia('C', 'N', 'e')
449         actual = c
450     elif cadena[conta] == 'h':
451         historia('D', 'N', 'h')
452         actual = d
453     elif cadena[conta] == 'm':
454         historia('E', 'N', 'm')
455         actual = e
456     elif cadena[conta] == 'p':
457         historia('F', 'N', 'p')
458         actual = f
459     elif cadena[conta] == 's':
460         historia('G', 'N', 's')
461         actual = g
462     elif cadena[conta] == 'w':
463         historia('H', 'N', 'w')
464         actual = h
465     else:
466         historia('A', 'N', cadena[conta])
467         actual = a
468 # Estado Enie
469 elif actual == enie:
470     if cadena[conta] == 'b':
471         historia('U', 'enie', 'b')
472         actual = u
473         guardarcadenas(linea, columna - 2, "web.txt")
474     # -----
475     elif cadena[conta] == 'c':
476         historia('B', 'enie', 'c')
477         actual = b
478     elif cadena[conta] == 'e':
479         historia('C', 'enie', 'e')
480         actual = c
481     elif cadena[conta] == 'h':
482         historia('D', 'enie', 'h')
483         actual = d
484     elif cadena[conta] == 'm':
485         historia('E', 'enie', 'm')
486         actual = e
487     elif cadena[conta] == 'p':
488         historia('F', 'enie', 'p')
489         actual = f

```

```

490     elif cadena[conta] == 's':
491         historia('G', 'enie', 's')
492         actual = g
493     elif cadena[conta] == 'w':
494         historia('H', 'enie', 'w')
495         actual = h
496     else:
497         historia('A', 'enie', cadena[conta])
498         actual = a
499 # Estado O
500 elif actual == o:
501     if cadena[conta] == 'n':
502         historia('V', 'O', 'n')
503         actual = v
504         guardarcadenas(linea, columna - 3, "coin.txt")
505     # -----
506     elif cadena[conta] == 'c':
507         historia('B', 'O', 'c')
508         actual = b
509     elif cadena[conta] == 'e':
510         historia('C', 'O', 'e')
511         actual = c
512     elif cadena[conta] == 'h':
513         historia('D', 'O', 'h')
514         actual = d
515     elif cadena[conta] == 'm':
516         historia('E', 'O', 'm')
517         actual = e
518     elif cadena[conta] == 'p':
519         historia('F', 'O', 'p')
520         actual = f
521     elif cadena[conta] == 's':
522         historia('G', 'O', 's')
523         actual = g
524     elif cadena[conta] == 'w':
525         historia('H', 'O', 'w')
526         actual = h
527     else:
528         historia('A', 'O', cadena[conta])
529         actual = a
530 # Estado P
531 elif actual == p:
532     if cadena[conta] == 'y':
533         historia('W', 'P', 'y')
534         actual = w

```

```

535         guardarcadenas(linea, columna - 3, "ebay.txt")
536     # -----
537     elif cadena[conta] == 'c':
538         historia('B', 'P', 'c')
539         actual = b
540     elif cadena[conta] == 'e':
541         historia('C', 'P', 'e')
542         actual = c
543     elif cadena[conta] == 'h':
544         historia('D', 'P', 'h')
545         actual = d
546     elif cadena[conta] == 'm':
547         historia('E', 'P', 'm')
548         actual = e
549     elif cadena[conta] == 'p':
550         historia('F', 'P', 'p')
551         actual = f
552     elif cadena[conta] == 's':
553         historia('G', 'P', 's')
554         actual = g
555     elif cadena[conta] == 'w':
556         historia('H', 'P', 'w')
557         actual = h
558     else:
559         historia('A', 'P', cadena[conta])
560         actual = a
561 # Estado Q
562 elif actual == q:
563     if cadena[conta] == 'e':
564         historia('X', 'Q', 'e')
565         actual = x
566         guardarcadenas(linea, columna - 3, "home.txt")
567     # -----
568     elif cadena[conta] == 'a':
569         historia('L', 'Q', 'a')
570         actual = l
571     # -----
572     elif cadena[conta] == 'c':
573         historia('B', 'Q', 'c')
574         actual = b
575     elif cadena[conta] == 'e':
576         historia('C', 'Q', 'e')
577         actual = c
578     elif cadena[conta] == 'h':
579         historia('D', 'Q', 'h')

```

```

580         actual = d
581     elif cadena[conta] == 'm':
582         historia('E', 'Q', 'm')
583         actual = e
584     elif cadena[conta] == 'p':
585         historia('F', 'Q', 'p')
586         actual = f
587     elif cadena[conta] == 's':
588         historia('G', 'Q', 's')
589         actual = g
590     elif cadena[conta] == 'w':
591         historia('H', 'Q', 'w')
592         actual = h
593     else:
594         historia('A', 'Q', cadena[conta])
595         actual = a
596 # Estado R
597 elif actual == r:
598     if cadena[conta] == 't':
599         historia('Y', 'R', 't')
600         actual = y
601     # -----
602     elif cadena[conta] == 'i':
603         historia('N', 'R', 'i')
604         actual = n
605     # -----
606     elif cadena[conta] == 'c':
607         historia('B', 'R', 'c')
608         actual = b
609     elif cadena[conta] == 'e':
610         historia('C', 'R', 'e')
611         actual = c
612     elif cadena[conta] == 'h':
613         historia('D', 'R', 'h')
614         actual = d
615     elif cadena[conta] == 'm':
616         historia('E', 'R', 'm')
617         actual = e
618     elif cadena[conta] == 'p':
619         historia('F', 'R', 'p')
620         actual = f
621     elif cadena[conta] == 's':
622         historia('G', 'R', 's')
623         actual = g
624     elif cadena[conta] == 'w':

```

```

625         historia('H', 'R', 'w')
626         actual = h
627     else:
628         historia('A', 'R', cadena[conta])
629         actual = a
630 # Estado S
631 elif actual == s:
632     if cadena[conta] == 'e':
633         historia('Z', 'S', 'e')
634         actual = z
635         guardarcadenas(linea, columna - 3, "page.txt")
636     # -----
637     elif cadena[conta] == 'c':
638         historia('B', 'S', 'c')
639         actual = b
640     elif cadena[conta] == 'e':
641         historia('C', 'S', 'e')
642         actual = c
643     elif cadena[conta] == 'h':
644         historia('D', 'S', 'h')
645         actual = d
646     elif cadena[conta] == 'm':
647         historia('E', 'S', 'm')
648         actual = e
649     elif cadena[conta] == 'p':
650         historia('F', 'S', 'p')
651         actual = f
652     elif cadena[conta] == 's':
653         historia('G', 'S', 's')
654         actual = g
655     elif cadena[conta] == 'w':
656         historia('H', 'S', 'w')
657         actual = h
658     else:
659         historia('A', 'S', cadena[conta])
660         actual = a
661 # Estado T
662 elif actual == t:
663     if cadena[conta] == 'e':
664         historia("A", 'T', 'e')
665         actual = _a_
666         guardarcadenas(linea, columna - 3, "site.txt")
667     # -----
668     elif cadena[conta] == 'c':
669         historia('B', 'T', 'c')

```



```

670         actual = b
671     elif cadena[conta] == 'e':
672         historia('C', 'T', 'e')
673         actual = c
674     elif cadena[conta] == 'h':
675         historia('D', 'T', 'c')
676         actual = d
677     elif cadena[conta] == 'm':
678         historia('E', 'T', 'c')
679         actual = e
680     elif cadena[conta] == 'p':
681         historia('F', 'T', 'c')
682         actual = f
683     elif cadena[conta] == 's':
684         historia('G', 'T', 'c')
685         actual = g
686     elif cadena[conta] == 'w':
687         historia('H', 'T', 'c')
688         actual = h
689     else:
690         historia('A', 'T', cadena[conta])
691         actual = a
692 # Estado U
693 elif actual == u:
694     if cadena[conta] == 'a':
695         historia('P', 'U', 'a')
696         actual = p
697     # -----
698     elif cadena[conta] == 'c':
699         historia('B', 'U', 'c')
700         actual = b
701     elif cadena[conta] == 'e':
702         historia('C', 'U', 'e')
703         actual = c
704     elif cadena[conta] == 'h':
705         historia('D', 'U', 'h')
706         actual = d
707     elif cadena[conta] == 'm':
708         historia('E', 'U', 'm')
709         actual = e
710     elif cadena[conta] == 'p':
711         historia('F', 'U', 'p')
712         actual = f
713     elif cadena[conta] == 's':
714         historia('G', 'U', 's')

```

```

715         actual = g
716     elif cadena[conta] == 'w':
717         historia('H', 'U', 'w')
718         actual = h
719     else:
720         historia('A', 'U', cadena[conta])
721         actual = a
722 # Estado V
723 elif actual == v:
724     if cadena[conta] == 'c':
725         historia('B', 'V', 'c')
726         actual = b
727     elif cadena[conta] == 'e':
728         historia('C', 'V', 'e')
729         actual = c
730     elif cadena[conta] == 'h':
731         historia('D', 'V', 'h')
732         actual = d
733     elif cadena[conta] == 'm':
734         historia('E', 'V', 'm')
735         actual = e
736     elif cadena[conta] == 'p':
737         historia('F', 'V', 'p')
738         actual = f
739     elif cadena[conta] == 's':
740         historia('G', 'V', 's')
741         actual = g
742     elif cadena[conta] == 'w':
743         historia('H', 'V', 'w')
744         actual = h
745     else:
746         historia('A', 'V', cadena[conta])
747         actual = a
748 # Estado W
749 elif actual == w:
750     if cadena[conta] == 'c':
751         historia('B', 'W', 'c')
752         actual = b
753     elif cadena[conta] == 'e':
754         historia('C', 'W', 'e')
755         actual = c
756     elif cadena[conta] == 'h':
757         historia('D', 'W', 'h')
758         actual = d
759     elif cadena[conta] == 'm':

```

```

760         historia('E', 'W', 'm')
761         actual = e
762     elif cadena[conta] == 'p':
763         historia('F', 'W', 'p')
764         actual = f
765     elif cadena[conta] == 's':
766         historia('G', 'W', 's')
767         actual = g
768     elif cadena[conta] == 'w':
769         historia('H', 'W', 'w')
770         actual = h
771     else:
772         historia('A', 'W', cadena[conta])
773         actual = a
774 # Estado X
775 elif actual == x:
776     if cadena[conta] == 'b':
777         historia('J', 'X', 'b')
778         actual = j
779     # -----
780     elif cadena[conta] == 'c':
781         historia('B', 'X', 'c')
782         actual = b
783     elif cadena[conta] == 'e':
784         historia('C', 'X', 'e')
785         actual = c
786     elif cadena[conta] == 'h':
787         historia('D', 'X', 'h')
788         actual = d
789     elif cadena[conta] == 'm':
790         historia('E', 'X', 'm')
791         actual = e
792     elif cadena[conta] == 'p':
793         historia('F', 'X', 'p')
794         actual = f
795     elif cadena[conta] == 's':
796         historia('G', 'X', 's')
797         actual = g
798     elif cadena[conta] == 'w':
799         historia('H', 'X', 'w')
800         actual = h
801     else:
802         historia('A', 'X', cadena[conta])
803         actual = a
804 # Estado Y

```

```

805     elif actual == y:
806         if cadena[conta] == 'e':
807             historia("B", 'Y', 'e')
808             actual = _b_
809         # -----
810         elif cadena[conta] == 'c':
811             historia('B', 'Y', 'c')
812             actual = b
813         elif cadena[conta] == 'e':
814             historia('C', 'Y', 'e')
815             actual = c
816         elif cadena[conta] == 'h':
817             historia('D', 'Y', 'h')
818             actual = d
819         elif cadena[conta] == 'm':
820             historia('E', 'Y', 'm')
821             actual = e
822         elif cadena[conta] == 'p':
823             historia('F', 'Y', 'p')
824             actual = f
825         elif cadena[conta] == 's':
826             historia('G', 'Y', 's')
827             actual = g
828         elif cadena[conta] == 'w':
829             historia('H', 'Y', 'w')
830             actual = h
831         else:
832             historia('A', 'Y', cadena[conta])
833             actual = a
834     # Estados Z
835     elif actual == z:
836         if cadena[conta] == 'b':
837             historia('J', 'Z', 'b')
838             actual = j
839         # -----
840         elif cadena[conta] == 'c':
841             historia('B', 'Z', 'c')
842             actual = b
843         elif cadena[conta] == 'e':
844             historia('C', 'Z', 'e')
845             actual = c
846         elif cadena[conta] == 'h':
847             historia('D', 'Z', 'h')
848             actual = d
849         elif cadena[conta] == 'm':

```

```

850         historia('E', 'Z', 'm')
851         actual = e
852     elif cadena[conta] == 'p':
853         historia('F', 'Z', 'p')
854         actual = f
855     elif cadena[conta] == 's':
856         historia('G', 'Z', 's')
857         actual = g
858     elif cadena[conta] == 'w':
859         historia('H', 'Z', 'w')
860         actual = h
861     else:
862         historia('A', 'Z', cadena[conta])
863         actual = a
864 # Estados A'
865 elif actual == _a_:
866     if cadena[conta] == 'b':
867         historia('J', "A'", 'b')
868         actual = j
869     # -----
870     elif cadena[conta] == 'c':
871         historia('B', "A'", 'c')
872         actual = b
873     elif cadena[conta] == 'e':
874         historia('C', "A'", 'e')
875         actual = c
876     elif cadena[conta] == 'h':
877         historia('D', "A'", 'h')
878         actual = d
879     elif cadena[conta] == 'm':
880         historia('E', "A'", 'm')
881         actual = e
882     elif cadena[conta] == 'p':
883         historia('F', "A'", 'p')
884         actual = f
885     elif cadena[conta] == 's':
886         historia('G', "A'", 's')
887         actual = g
888     elif cadena[conta] == 'w':
889         historia('H', "A'", 'w')
890         actual = h
891     else:
892         historia('A', "A'", cadena[conta])
893         actual = a
894 # Estado B'

```

```

895     elif actual == _b_:
896         if cadena[conta] == 'r':
897             historia("C'", "B'", 'r')
898             actual = _c_
899             guardarcadenas(linea, columna - 5, "master.txt")
900         # -----
901         elif cadena[conta] == 'b':
902             historia('J', "B'", 'b')
903             actual = j
904         # -----
905         elif cadena[conta] == 'c':
906             historia('B', "B'", 'c')
907             actual = b
908         elif cadena[conta] == 'e':
909             historia('C', "B'", 'e')
910             actual = c
911         elif cadena[conta] == 'h':
912             historia('D', "B'", 'h')
913             actual = d
914         elif cadena[conta] == 'm':
915             historia('E', "B'", 'm')
916             actual = e
917         elif cadena[conta] == 'p':
918             historia('F', "B'", 'p')
919             actual = f
920         elif cadena[conta] == 's':
921             historia('G', "B'", 's')
922             actual = g
923         elif cadena[conta] == 'w':
924             historia('H', "B'", 'w')
925             actual = h
926         else:
927             historia('A', "B'", cadena[conta])
928             actual = a
929     # Estado C'
930     elif actual == _c_:
931         if cadena[conta] == 'c':
932             historia('B', "C'", 'c')
933             actual = b
934         elif cadena[conta] == 'e':
935             historia('C', "C'", 'e')
936             actual = c
937         elif cadena[conta] == 'h':
938             historia('D', "C'", 'h')
939             actual = d

```

```

940         elif cadena[conta] == 'm':
941             historia('E', "C'", 'm')
942             actual = e
943         elif cadena[conta] == 'p':
944             historia('F', "C'", 'p')
945             actual = f
946         elif cadena[conta] == 's':
947             historia('G', "C'", 's')
948             actual = g
949         elif cadena[conta] == 'w':
950             historia('H', "C'", 'w')
951             actual = h
952         else:
953             historia('A', "C'", cadena[conta])
954             actual = a
955     conta = conta + 1
956
957 def archivohistoria():
958     archivo = open("archivos/archivohistoria.txt", "w")
959     archivo.write("HISTORIA")
960     archivo.write("\n")
961     archivo.write("{(letra actual)[estado actual]->[estado siguiente]}")
962     archivo.write("\n")
963     with open("archivos/historia.txt") as historia:
964         for linea in historia:
965             archivo.write(linea)
966     archivo.close()
967
968
969 def historia(estsiguiente, estactual, letra):
970     archivo = open("archivos/historia.txt", "a")
971     linea = "{(" + letra + ")( " + estactual + ")->[" + estsiguiente + "
972     ]}"
973     archivo.write(linea)
974     archivo.write("\n")
975     archivo.close()
976
977 # Funci n para listar los archivos que esten en el directorio para
978 leerlos
979 def listararchivos():
980     ejemplo_dir = '/Users/cesar/OneDrive/Documentos/Cuarto Semestre/
981     TEORIA DE LA COMPUTACION/Segundo Parcial/Buscador De Palabras/
982     ProgramaFinal/archivos'

```

```

980     with os.scandir(ejemplo_dir) as ficheros:
981         for fichero in ficheros:
982             print(fichero.name)
983
984
985 # Funci n para limpiar la pantalla/consola/terminal
986 def limpiarpantalla():
987     if os.name == "posix":
988         os.system("clear")
989     elif os.name == "ce" or os.name == "nt" or os.name == "dos":
990         os.system("cls")
991
992
993 def guardarcadenas(numlinea, numcolumna, nombreadchivo):
994     ruta = "archivos/" + nombreadchivo
995     archivo = open(ruta, "a")
996     archivo.write("[")
997     archivo.write(str(numlinea))
998     archivo.write(" ")
999     archivo.write(str(numcolumna))
1000    archivo.write("]\n")
1001    archivo.close()
1002
1003
1004 def automataArchivo(nombreadchivo):
1005     ruta = "archivos/" + nombreadchivo
1006     # Usando with el archivo se cierra automaticamente
1007     conta = 0
1008     with open(ruta, 'r') as archivo:
1009         for linea in archivo:
1010             conta = conta + 1
1011             automata(linea, conta)
1012             # print(linea)
1013
1014
1015 def borrararchivos():
1016     if os.path.isfile("archivos/ebay.txt"):
1017         os.remove("archivos/eBay.txt")
1018     if os.path.isfile("archivos/coin.txt"):
1019         os.remove("archivos/coin.txt")
1020     if os.path.isfile("archivos/home.txt"):
1021         os.remove("archivos/home.txt")
1022     if os.path.isfile("archivos/master.txt"):
1023         os.remove("archivos/master.txt")
1024     if os.path.isfile("archivos/page.txt"):

```



```

1025         os.remove("archivos/page.txt")
1026     if os.path.isfile("archivos/site.txt"):
1027         os.remove("archivos/site.txt")
1028     if os.path.isfile("archivos/web.txt"):
1029         os.remove("archivos/web.txt")
1030     if os.path.isfile("archivos/historia.txt"):
1031         os.remove("archivos/historia.txt")
1032
1033
1034
1035     def contarpalabra(nombreamarchivo):
1036         ruta = "archivos/" + nombreamarchivo + ".txt"
1037         with open(ruta, 'r') as archivo:
1038             conta = 0
1039             for linea in archivo:
1040                 conta = conta + 1
1041         return conta
1042
1043
1044     def guardarlineas(nombreamarchivo, file):
1045         ruta = "archivos/" + nombreamarchivo + ".txt"
1046         if os.path.isfile(ruta):
1047             with open(ruta) as archivo:
1048                 file.write(nombreamarchivo)
1049                 file.write("\n")
1050                 cantidad = contarpalabra(nombreamarchivo)
1051                 file.write("cantidad:")
1052                 file.write(str(cantidad))
1053                 file.write("\n[fila columna]\n")
1054                 for linea in archivo:
1055                     nuevalinea = ''
1056                     for caracter in linea:
1057                         if caracter != "\n":
1058                             nuevalinea = nuevalinea + caracter
1059                     file.write(nuevalinea)
1060                 file.write("\n")
1061                 file.write("\n")
1062
1063
1064     def juntararchivos():
1065         with open("archivos/palabrasreservadas.txt", "a") as file:
1066             guardarlineas("coin", file)
1067             guardarlineas("ebay", file)
1068             guardarlineas("home", file)
1069             guardarlineas("master", file)

```

```

1070     guardarlineas("page", file)
1071     guardarlineas("site", file)
1072     guardarlineas("web", file)
1073
1074
1075 def FuncArchivo():
1076     if os.path.isfile("archivos/historia.txt"):
1077         os.remove("archivos/historia.txt")
1078     # Se verifica que el archivo solicitado exista para poder continuar
1079     limpiarpantalla()
1080     print("Archivo")
1081     listararchivos()
1082     try:
1083         print(" Que archivo desea evaluar? ")
1084         respuesta = input()
1085         automataArchivo(respuesta)
1086         print("El archivo ha sido analizado con exito.")
1087     except FileNotFoundError:
1088         limpiarpantalla()
1089         print("Archivo no encontrado")
1090     except FileExistsError:
1091         limpiarpantalla()
1092         print("Archivo no existente")
1093     except:
1094         limpiarpantalla()
1095         print("Ninguna entrada")
1096
1097
1098 def FuncCadena():
1099     limpiarpantalla()
1100     if os.path.isfile("archivos/historia.txt"):
1101         os.remove("archivos/historia.txt")
1102     print("Cadena")
1103     try:
1104         print("Ingrese la cadena a evaluar: ")
1105         respuesta = input()
1106         automata(respuesta, 1)
1107         print("La cadena ha sido analizada con exito.")
1108     except:
1109         limpiarpantalla()
1110         print("Ninguna entrada")
1111
1112
1113 def menu():
1114     while True:

```

```

1115     limpiarpantalla()
1116     print("Menu Programa 4")
1117     print("1. Ingresar un Archivo de Texto")
1118     print("2. Ingresar una cadena")
1119     print("3. Ver el Grafo del Programa")
1120     print("4. Salir")
1121     respuesta = input("Favor de escoger una opcion\n")
1122
1123
1124     if respuesta == '1':
1125         # Archivo()
1126         if os.path.isfile("archivos/palabrasreservadas.txt"):
1127             os.remove("archivos/palabrasreservadas.txt")
1128         if os.path.isfile("archivos/archivohistoria.txt"):
1129             os.remove("archivos/archivohistoria.txt")
1130         borrararchivos()
1131         FuncArchivo()
1132         juntararchivos()
1133         archivohistoria()
1134         borrararchivos()
1135         os.system("pause")
1136     elif respuesta == '2':
1137         # Cadena()
1138         if os.path.isfile("archivos/palabrasreservadas.txt"):
1139             os.remove("archivos/palabrasreservadas.txt")
1140         if os.path.isfile("archivos/archivohistoria.txt"):
1141             os.remove("archivos/archivohistoria.txt")
1142         borrararchivos()
1143         FuncCadena()
1144         juntararchivos()
1145         archivohistoria()
1146         borrararchivos()
1147         os.system("pause")
1148     elif respuesta == '3':
1149         #GRAFO
1150         print("Grafo del Programa")
1151         # Cargar la imagen con openCV
1152         imgBGR = cv2.imread("archivos/GrafoConLetras.png")
1153         # Cambiar espacio de color BGR a RGB
1154         imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)
1155         # Mostrar imagen
1156         plt.xticks([], plt.yticks([]))
1157         plt.imshow(imgRGB, cmap='gray', interpolation='bicubic')
1158         plt.show()
1159

```

```

1160         elif respuesta == '4':
1161             break
1162         else:
1163             print("Favor de escoger una opcion valida")
1164
1165     print("Fin del Programa 4")
1166
1167
1168 if __name__ == '__main__':
1169     menu()

```

2.2. Explicación

El programa tiene la función principal de identificar 7 palabras reservadas, y trabajar con textos, el uso de las funciones es el siguiente: `autómata(cadena, linea)`:

Es la función del `autómata`, en esta función se encuentra todo el comportamiento del `autómata`, esto es que pasa en cada estado cuando llega una letra. El `autómata` tiene en total de 30 estados, representados con letras de la A a la Z y luego A', B', C' pues son mas estados que letras del alfabeto completo. El `autómata` recibe una cadena de caracteres y luego va recorriendo cada letra que se encuentre en esa cadena para poder pasar todas las letras al `autómata`. Si en el proceso la palabra es alguna de las 7 palabras reservadas se guarda la ubicación donde se encontró en un archivo de las mismas palabras. En esta función también se usa la funcion `historia()` para la creación del archivo de Historia, dependiendo de el estado actual, estado siguiente, y la letra recibida.

`archivohistoria()`:

En esta función se le añade un titulo y como esta organizada la información en al archivo historia creado con la funcion `historia()`, haciendo un poco más entendible ese archivo.

`historia(estsiguiente, estactual, letra)`:

En esta función se guarda la historia del automata en un `archivo.txt` de cada cadena que se haya evaluado en el `autómata`.

`listararchivos()`:

Esta función lista que archivos se encuentran en el directorio de archivos, para que despues el usuario pueda elegir entre los archivos de texto existentes.

`limpiarpantalla()`:

Es la función que limpia la consola/pantalla cuando se necesite.

`guardarcadenas(numline, numcolumna, nombearchivo)`:

Esta función guarda donde se encontró alguna de las palabras reservadas, en su archivo de texto

correspondiente. Si es por ejemplo la palabra coin se guarda en coin.txt, si es home se guarda en home.txt y así con todas las palabras reservadas.

automataArchivo(nombrearchivo): Es la función que se usa cuando se selecciona la opción del menú de ingresar archivo. Se encarga de pasar línea por línea del archivo al autómata para su evaluación total.

borrararchivos():

Es la función que borra los archivos que se encuentren en la ruta del directorio para los nuevos archivos que se crearan y que no se tenga problemas de versiones o de datos antiguos, pues la mayoría de los archivos con los que se trabajan son de tipo append por lo que si ya existía el archivo solo se agregan los nuevos datos al final.

contarpalabra(nombrearchivo):

Cuenta la cantidad de líneas que tiene un archivo, esto para poder saber cuantas palabras se encontraron cada vez. Porque cuando se encuentra una palabra se guarda su ubicación en una sola línea, por lo que si hubo varias cada una ocupa una línea diferente del archivo de esa palabra.

guardarlineas(nombrearchivo, file):

Es la función que escribe la información de cada palabra, osea que palabra se encontró, cuantas veces, y todas las ubicaciones de estas palabras, en el archivo final "palabrasreservadas.txt"

juntararchivos():

Es la función que trabaja con la de guardarlineas() para poder juntar todos los datos de todos los archivos creados, uno por cada palabra registrada si es que se crearon.

FuncArchivo():"

Es la función la cual manda llamar las funciones necesarias para la opción del menú de evaluar un archivo de texto.

FuncCadena():

Es la función la cual manda llamar las funciones necesarias para la opción del menú de evaluar una cadena.

menu():

Es la función que despliega el menú del programa con las opciones de evaluar un archivo de texto, evaluar una cadena, y la opción de ver el grafo del programa (Que es una imagen para menos complicaciones).

	a	b	c	e	g	h	i	m	n	o	p	r	s	t	w	y	
->A	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1	
B	1,2	1	1,2	1,6	1	1,10	1	1,14	1	1,3	1,20	1	1,24	1	1,28	1	
C	1,6	1	1,7	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
D	1,10	1	1	1,2	1,6	1	1,10	1	1,14	1	1,11	1,20	1	1,24	1	1,28	1
E	1,14	1,15	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
F	1,20	1,21	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
G	1,24	1	1	1,2	1,6	1	1,10	1,25	1,14	1	1	1,20	1	1,24	1	1,28	1
H	1,28	1	1	1,2	1,6,29	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
I	1,3	1	1	1,2	1,6	1	1,10	1,4	1,14	1	1	1,20	1	1,24	1	1,28	1
J	1,7	1,8	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
K	1,11	1	1	1,2	1,6	1	1,10	1	1,12,14	1	1	1,20	1	1,24	1	1,28	1
L	1,15	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,16,24	1	1,28	1
M	1,21	1	1	1,2	1,6	1,22	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
N	1,25	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1,26	1,28	1
N	1,6,29	1	1,7,30	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
O	1,4	1	1	1,2	1,6	1	1,10	1	1,14	1,5	1	1,20	1	1,24	1	1,28	1
P	1,8	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1,9
Q	1,12,14	1,15	1	1,2	1,6,13	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
R	1,16,24	1	1	1,2	1,6	1	1,10	1,25	1,14	1	1	1,20	1	1,24	1,17	1,28	1
S	1,22	1	1	1,2	1,6,23	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
T	1,26	1	1	1,2	1,6,27	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
*U	* 1,7,30	1,8	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
*V	* 1,5	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
*W	* 1,9	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
*X	* 1,6,13	1	1,7	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
Y	1,17	1	1	1,2	1,6,18	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
*Z	* 1,6,23	1	1,7	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
*A'	* 1,6,27	1	1,7	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1
B'	1,6,18	1	1,7	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1,19	1,24	1	1,28	1
*C'	* 1,19	1	1	1,2	1,6	1	1,10	1	1,14	1	1	1,20	1	1,24	1	1,28	1

Tabla del autómata Finito No Determinista

			a	b	c	e	g	h	i	m	n	o	p	r	s	t	w	y
->A	->	1	A	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
B		1,2	A	A	B	C	A	D	A	E	A	I	F	A	G	A	H	A
C		1,6	A	J	B	C	A	D	A	E	A	A	F	A	G	A	H	A
D		1,10	A	A	B	C	A	D	A	E	A	K	F	A	G	A	H	A
E		1,14	L	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
F		1,20	M	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
G		1,24	A	A	B	C	A	D	N	E	A	A	F	A	G	A	H	A
H		1,28	A	A	B	Ñ	A	D	A	E	A	A	F	A	G	A	H	A
I		1,3	A	A	B	C	A	D	O	E	A	A	F	A	G	A	H	A
J		1,7	P	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
K		1,11	A	A	B	C	A	D	A	Q	A	A	F	A	G	A	H	A
L		1,15	A	A	B	C	A	D	A	E	A	A	F	A	R	A	H	A
M		1,21	A	A	B	C	S	D	A	E	A	A	F	A	G	A	H	A
N		1,25	A	A	B	C	A	D	A	E	A	A	F	A	G	T	H	A
Ñ		1,6,29	A	U	B	C	A	D	A	E	A	A	F	A	G	A	H	A
O		1,4	A	A	B	C	A	D	A	E	V	A	F	A	G	A	H	A
P		1,8	A	A	B	C	A	D	A	E	A	A	F	A	G	A	H	W
Q		1,12,14	L	A	B	X	A	D	A	E	A	A	F	A	G	A	H	A
R		1,16,24	A	A	B	C	A	D	N	E	A	A	F	A	G	Y	H	A
S		1,22	A	A	B	Z	A	D	A	E	A	A	F	A	G	A	H	A
T		1,26	A	A	B	A'	A	D	A	E	A	A	F	A	G	A	H	A
*U	*	1,7,30	P	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
*V	*	1,5	A	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
*W	*	1,9	A	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A
*X	*	1,6,13	A	J	B	C	A	D	A	E	A	A	F	A	G	A	H	A
Y		1,17	A	A	B	B'	A	D	A	E	A	A	F	A	G	A	H	A
*Z	*	1,6,23	A	J	B	C	A	D	A	E	A	A	F	A	G	A	H	A
*A'	*	1,6,27	A	J	B	C	A	D	A	E	A	A	F	A	G	A	H	A
B'		1,6,18	A	J	B	C	A	D	A	E	A	A	F	C'	G	A	H	A
*C'	*	1,19	A	A	B	C	A	D	A	E	A	A	F	A	G	A	H	A

Tabla del autómata Finito Determinista

2.3. Ejecución del programa, en Phyton

Menú del programa:

```
Menu Programa 4
1. Ingresar un Archivo de Texto
2. Ingresar una cadena
3. Ver el Grafo del Programa
4. Salir
Favor de escoger una opcion
```

Opción 1. Archivo de Texto:

```
Archivo
GrafoConLetras.png
Palabras.txt
Palabras2.txt
¿Que archivo desea evaluar?
Palabras.txt
El archivo ha sido analizado con exito.
Presione una tecla para continuar . . .
```

Archivos creados por la opción anterior:



```

1 Lorem ipsum dolor sit amet. Master page coin home web page home. Web eBay page web page page home. Master site web eBay site web site coin web site coin site eBay web eBay master
2 web master home? Page site web coin eBay coin eBay web master web master home web. Coin site page home site master web coin master. Web site page web page page web page site
3 master coin! Web master eBay master web page web page page coin master web coin. eBay web eBay eBay home web eBay home eBay page. Master coin web eBay home master
4 site? Page home web master page site coin page site web coin eBay web page! Web coin master web eBay home home coin eBay coin. Web site eBay web page page web master coin
5 coin eBay web coin home web site site web master coin. Web page home web coin master web master web page master? Web site page home eBay page home web eBay coin web master
6 site. eBay site site web site page page site web master master web master. Master page web site home site web master. Web page eBay eBay master web page master page coin page
7 coin web site. Web home page page web site page eBay coin site web master page page? Web page master site coin site coin web page web page eBay page eBay. Web site
8 page coin. Web master page web page coin master home coin page. Coin coin home home eBay web site web eBay page master site. Site web coin site site home web eBay master
9 web page page coin master web page coin. Site home home site master master page web coin site web web page site. Web home master web master home master coin home coin.
10 Site coin web home home eBay web coin home web master eBay site eBay web master home. Web site site master master web master site web eBay master web coin eBay. Web master site
11 web eBay site web page master web eBay eBay home eBay web page page. Master master page site site page home home site site site web coin eBay. Web home page page home web
12 page site web home page. Web eBay master home page site home eBay home site. Web home master web page site home. Web coin site coin web home eBay coin eBay. Web eBay web
13 site web site page coin site. eBay coin web site page home web site page web master coin web coin coin master. Web eBay coin web home home web coin site master page. Home site
14 master page web master master eBay home. Web eBay site eBay master site eBay home web coin site web site web eBay page? Web home page master master web eBay coin master coin
15 web eBay coin site page! Web home web eBay web master site web home site coin web master site. Web page master web eBay eBay web coin web home eBay. Page coin web master
16 eBay web coin! Web page web master page web site eBay web coin coin site eBay coin coin. Site eBay web site coin web home coin. Web page site web home site master home eBay.
17 Web coin master home web site eBay. Coin page web coin master coin eBay page web home page. eBay page web eBay home web home home page. Web site master web site site web
18 coin site coin web eBay web master coin. Home web page site eBay coin coin site. Web master master site site eBay. Web master coin page page web page site web coin master!
19 Web master web site home home page web site page web site web site eBay. Web eBay home page web site page web master eBay. Web page coin web home coin web page page master
20 site.
21 webay
22

```

```

1 coin
2 cantidad:77
3 [fila columna]
4 [1 42][1 138][1 152][2 32][2 43][2 127][3 8][3 15][3 71][3 88][3 154][4 41][4 62][4 87][4 122][4 133][4 178][5 1][5 15][5
5
6 ebay
7 cantidad:1
8 [fila columna]
9 [21 2]
10
11 home
12 cantidad:65
13 [fila columna]
14 [1 47][1 61][1 97][2 12][2 76][2 102][3 116][3 130][3 168][4 12][4 108][4 117][5 20][5 68][5 93][5 137][5 153][6 100][7 20
15
16 master
17 cantidad:75
18 [fila columna]
19 [1 177][2 5][2 58][2 69][2 115][2 132][3 1][3 25][3 39][3 76][3 174][4 22][4 92][4 171][5 43][5 82][5 98][5 114][5 176][6
20
21 page
22 cantidad:88
23 [fila columna]
24 [1 36][1 56][1 76][1 86][1 92][2 97][2 149][2 158][2 163][2 172][3 51][3 60][3 65][3 141][4 30][4 48][4 77][4 157][4 162][
25
26 site
27 cantidad:89
28 [fila columna]
29 [1 110][1 124][1 133][1 147][1 158][2 23][2 91][2 109][2 144][2 177][4 1][4 35][4 53][4 143][5 29][5 34][5 126][6 1][6 12]
30
31 web
32 cantidad:122
33 [fila columna]
34 [1 52][1 81][1 115][1 129][1 143][1 168][2 1][2 28][2 53][2 65][2 81][2 123][2 154][2 168][3 46][3 56][3 83][3 100][3 121]
35

```

```

1 HISTORIA
2 {(letra actual)[estado actual]->[estado siguiente]}
3 {(L)[A]->[A]}
4 {(O)[A]->[A]}
5 {(R)[A]->[A]}
6 {(E)[A]->[C]}
7 {(M)[C]->[E]}
8 {( ) [E]->[A]}
9 {(I)[A]->[A]}
10 {(P)[A]->[F]}
11 {(S)[F]->[G]}
12 {(U)[G]->[A]}
13 {(M)[A]->[E]}
14 {( ) [E]->[A]}
15 {(D)[A]->[A]}
16 {(O)[A]->[A]}
17 {(L)[A]->[A]}
18 {(O)[A]->[A]}
19 {(R)[A]->[A]}
20 {( ) [A]->[A]}
21 {(S)[A]->[G]}
22 {(I)[G]->[N]}
23 {(T)[N]->[T]}
24 {( ) [T]->[A]}
25 {(A)[A]->[A]}
26 {(M)[A]->[E]}
27 {(E)[E]->[C]}
28 {(T)[C]->[A]}
29 {(.) [A]->[A]}
30 {( ) [A]->[A]}
31 {(M)[A]->[A]}
32 {(A)[A]->[A]}
33 {(S)[A]->[G]}
34 {(T)[G]->[A]}
35 {(E)[A]->[C]}

```

Opción 2. Una cadena de caracteres:

```
Cadena
Ingrese la cadena a evaluar:
web, webpage, website, webmaster, webhome, ebay, coin
La cadena ha sido analizada con exito.
Presione una tecla para continuar . . .
```

Archivos creados por la opción anterior:

```
palabrasreservadas.txt
1 coin
2 cantidad:1
3 [fila columna]
4 [1 50]
5
6 ebay
7 cantidad:1
8 [fila columna]
9 [1 44]
10
11 home
12 cantidad:1
13 [fila columna]
14 [1 38]
15
16 master
17 cantidad:1
18 [fila columna]
19 [1 27]
20
21 page
22 cantidad:1
23 [fila columna]
24 [1 9]
25
26 site
27 cantidad:1
28 [fila columna]
29 [1 18]
30
31 web
32 cantidad:5
33 [fila columna]
34 [1 1][1 6][1 15][1 24][1 35]
```

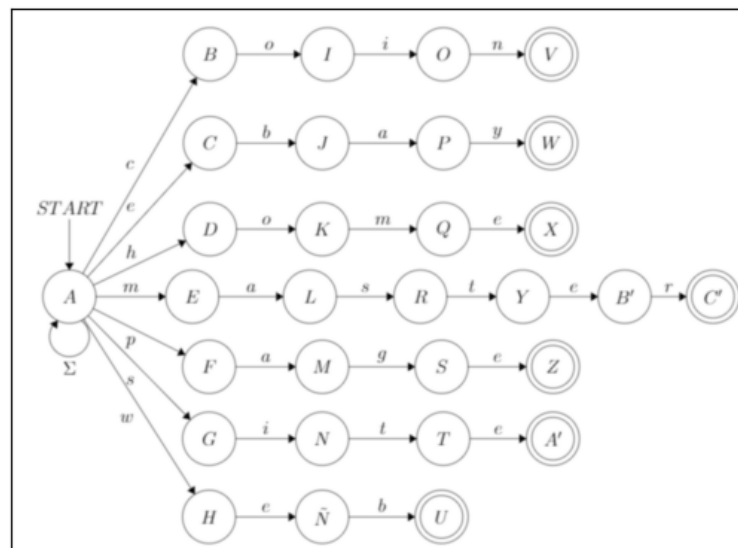
```

1 HISTORIA
2 {(letra actual)[estado actual]->[estado siguiente]}
3 {(w)[A]->[H]}
4 {(e)[H]->[enie]}
5 {(b)[enie]->[U]}
6 {(,)[U]->[A]}
7 {( ) [A]->[A]}
8 {(w)[A]->[H]}
9 {(e)[H]->[enie]}
10 {(b)[enie]->[U]}
11 {(p)[U]->[F]}
12 {(a)[F]->[M]}
13 {(g)[M]->[S]}
14 {(e)[S]->[Z]}
15 {(,)[Z]->[A]}
16 {( ) [A]->[A]}
17 {(w)[A]->[H]}
18 {(e)[H]->[enie]}
19 {(b)[enie]->[U]}
20 {(s)[U]->[G]}
21 {(i)[G]->[N]}
22 {(t)[N]->[T]}
23 {(e)[T]->[A']}
24 {(,)[A']->[A]}
25 {( ) [A]->[A]}
26 {(w)[A]->[H]}
27 {(e)[H]->[enie]}
28 {(b)[enie]->[U]}
29 {(m)[U]->[E]}
30 {(a)[E]->[L]}
31 {(s)[L]->[R]}
32 {(t)[R]->[Y]}
33 {(e)[Y]->[B']}
34 {(r)[B']->[C']}

```

Opción 3. Ver el Grafo del Programa

Figure 1



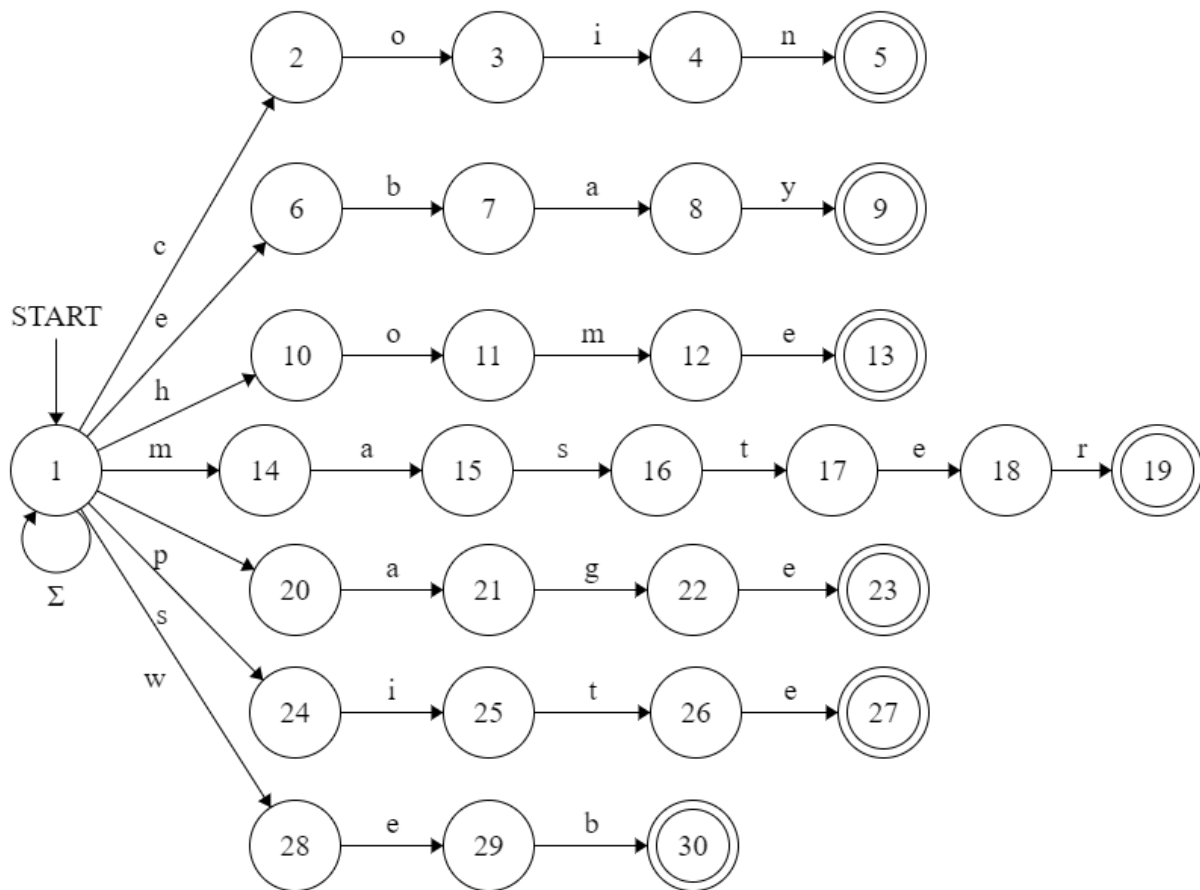
x=416. y=220.
[255, 255, 255]

Opción 4. Salir del Programa

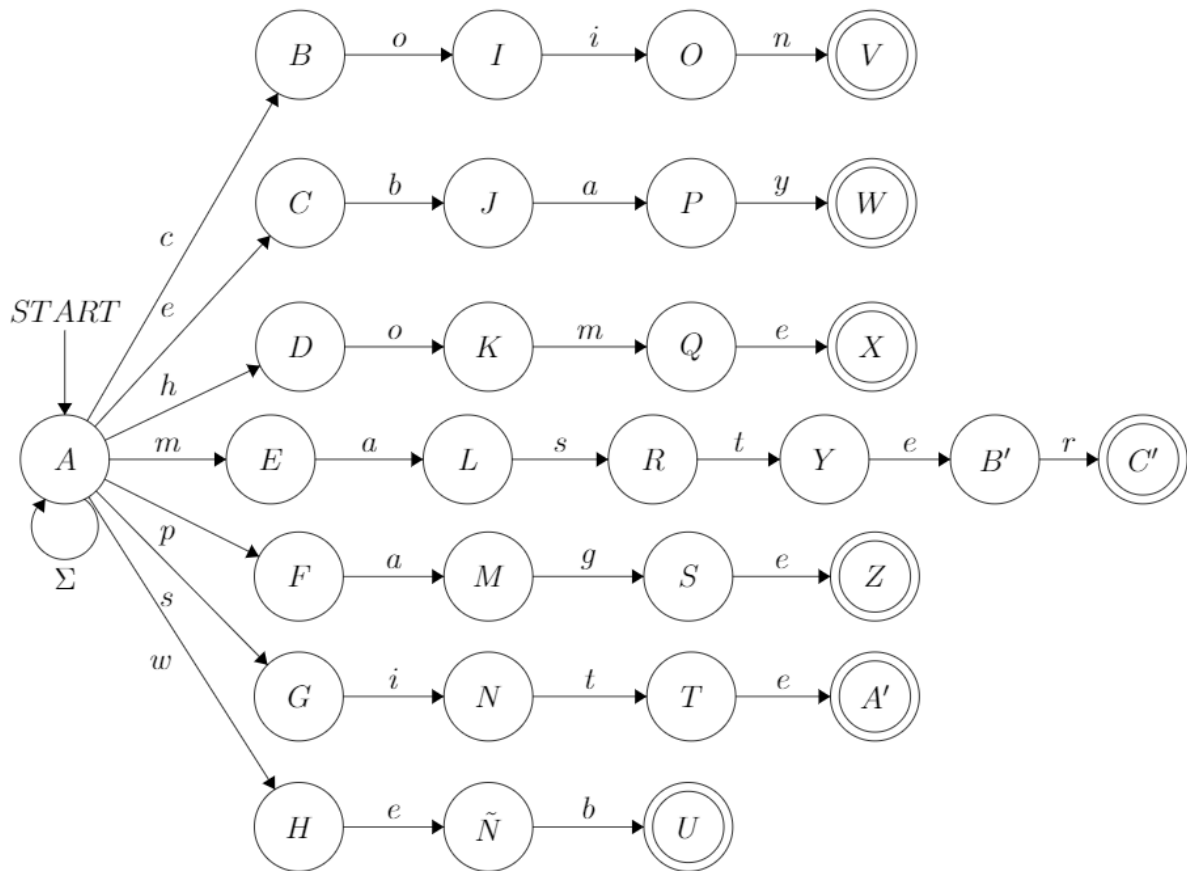
```
Menu Programa 4
1. Ingresar un Archivo de Texto
2. Ingresar una cadena
3. Ver el Grafo del Programa
4. Salir
Favor de escoger una opcion
4
Fin del Programa 4
```

3. Notas Importantes

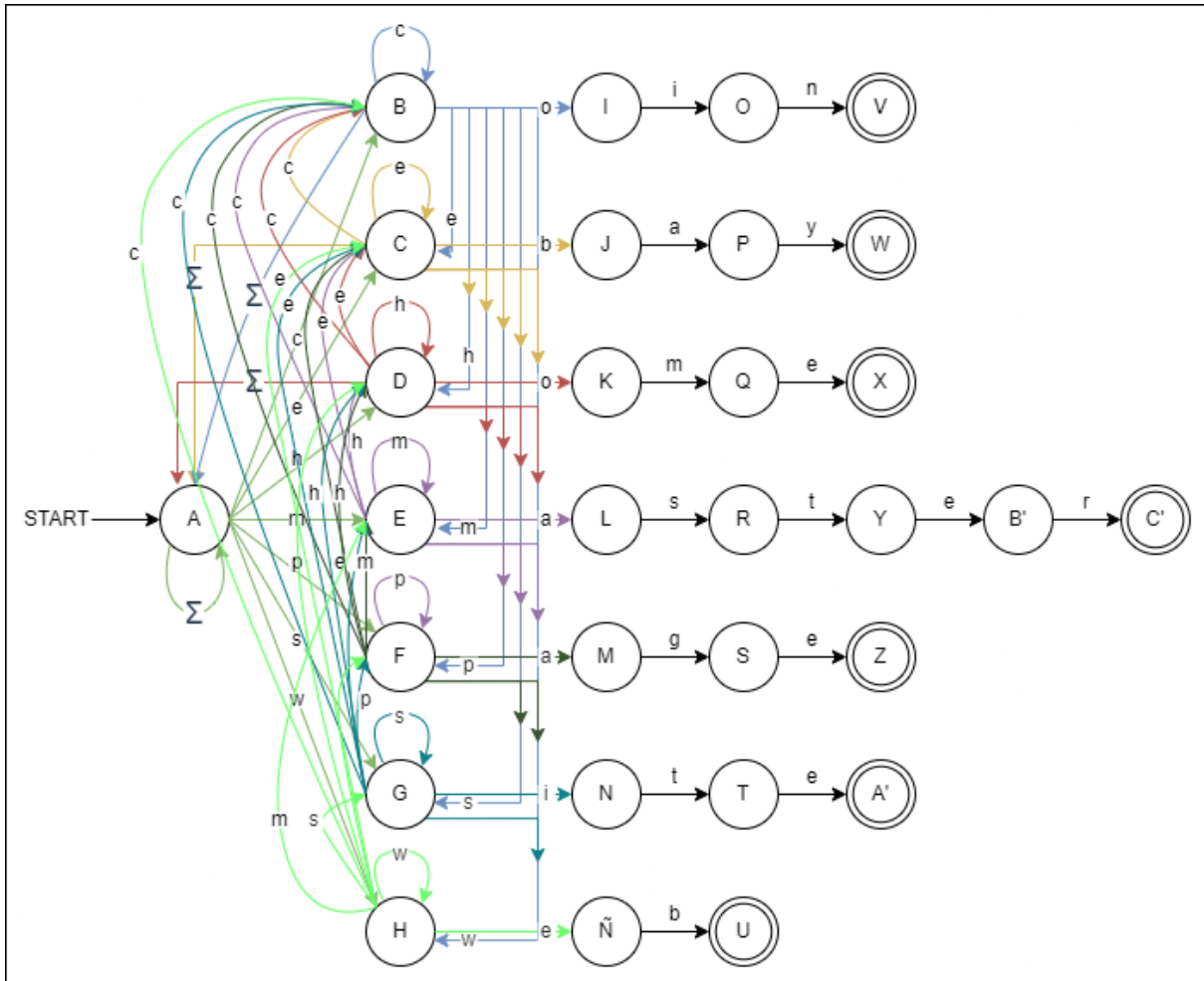
El grafo en un principio se trabajo con números:



Después con las letras:



Pero el Grafo del AFD estaba muy complicado terminarlo y entenderlo por lo que quedo de la siguiente manera, lo que falta es que en cada estado que falta desde la I hasta la C' cuando les llegan las letras iniciales de las palabras reservadas, todos estos estados se van a los estados B,C,D,E,F,G o H dependiendo la letra, y si no es ninguna de las letras iniciales todos los estados se van al estado A para empezar otra vez el autómata:



Por lo que se quedo el grafo del AFND en la opción de mostrar grafo en el punto 3 del programa.

4. Conclusiones

Este programa me tomo mucho tiempo completarlo, aunque estuvo entretenido hacerlo, fue muy confuso en algunas partes al codificar el autómata pues me perdía con tantos estados y las conexiones entre estos. Al final creo salio bien como se esperaba en un principio.