



UNIVERSIDAD POLITÉCNICA DE CHIAPAS

---

ID Y NOMBRE DE LA ACTIVIDAD

**Lenguajes y autómatas**

**2024.C3.07 L&A**

P R E S E N T A :

221189 LUIS ALBERTO BATALLA GONZÁLEZ

Docente:

CARLOS ALBERTO DÍAZ HERNÁNDEZ

CHIAPAS, CUATRIMESTRE SEPTIEMBRE - DICIEMBRE DE 2024

# Índice general

<b>1. Propuesta de aplicativo de autómata finito determinista</b>	<b>3</b>
1.1. RepuIDValidator . . . . .	3
1.2. Descripción . . . . .	3
1.3. Cadenas válidas e inválidas . . . . .	3
1.3.1. Cadenas válidas . . . . .	3
1.3.2. Cadenas inválidas . . . . .	4
1.4. Restricciones . . . . .	4
<b>2. AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto</b>	<b>5</b>
2.1. Descripción de la actividad . . . . .	5
2.2. RepuIDValidator: Validador de identificadores del Reclusorio Preventivo (REPU) que asegura el cumplimiento del formato establecido. . . . .	5
2.3. Descripción . . . . .	5
2.3.1. Cadenas válidas . . . . .	6
2.3.2. Cadenas inválidas . . . . .	6
2.3.3. Restricciones . . . . .	6
2.4. Diseño . . . . .	6
2.4.1. Especificación formal . . . . .	6
2.4.2. Diagrama . . . . .	7
<b>3. Implementación de AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto</b>	<b>9</b>
3.1. Descripción de la actividad . . . . .	9
3.2. RepuIDValidator: . . . . .	9
3.3. Descripción . . . . .	9
3.3.1. Cadenas válidas . . . . .	10
3.3.2. Cadenas inválidas . . . . .	10
3.3.3. Restricciones . . . . .	10
3.3.4. Diseño . . . . .	10
3.4. Implementación . . . . .	10
3.4.1. Interfaz . . . . .	11
3.4.2. Función de transición . . . . .	11
3.4.3. Pruebas realizadas . . . . .	13
3.5. Comentarios . . . . .	13

# Índice de figuras

2.1. Diagrama del AFD $D$ . . . . .	8
-------------------------------------	---

3.1. Diagrama del AFD $D$ . . . . .	11
3.2. Interfaz gráfica del sistema que implementa al AFD $D$ . . . . .	11
3.3. Captura de ejecución 1 $D$ . . . . .	14
3.4. Captura de ejecución 2 $D$ . . . . .	14
3.5. Captura de ejecución 3 $D$ . . . . .	14

# Índice de tablas

2.1. Matriz de transición del autómata finito determinista RepUIDValidator. . . . .	7
---	---

# Listings

3.1. Ejemplo de código insertado . . . . .	11
3.2. Ejemplo de código insertado . . . . .	13

# Lista de ecuaciones

# Capítulo 1

## Propuesta de aplicativo de autómata finito determinista

Periodo:	2024.C3 Septiembre - diciembre de 2024
Grupo:	7A/B
Asignatura:	Lenguajes y autómatas
Corte:	1
Actividad:	2024.C3.07X.L&A.C1.A2
Título:	Propuesta de aplicativo de autómata finito determinista
Fecha de entrega:	2024.09.24
<b>221189</b>	<b>Luis Alberto Batalla González</b>

### 1.1. RepuIDValidator

,RepuIDValidator Validador de identificadores del Reclusorio Preventivo (REPU) que asegura el cumplimiento del formato establecido.

### 1.2. Descripción

El sistema implementará un autómata finito determinista que hará: *Indicar en las viñetas lo que hará*

El autómata recibirá una cadena de entrada y analizará si la cadena corresponde a un identificador del Reclusorio Preventivo, que debe seguir las siguientes reglas: El identificador comenzará con el prefijo REPU", seguido de un guion. Luego debe contener el año de ingreso, representado por 4 dígitos. A continuación, debe haber un guion seguido de un número secuencial de 5 dígitos. Finalmente, debe terminar con un guion y un dígito verificador (0-9). Si la cadena no es válida, el autómata se reiniciará tomando como entrada la cadena a partir de la posición siguiente a donde comenzó. Si encuentra un identificador válido, registrará la posición y la longitud del identificador válido y se reiniciará a partir de la posición siguiente al final del identificador válido.

### 1.3. Cadenas válidas e inválidas

#### 1.3.1. Cadenas válidas

Ejemplo de cadenas válidas que debe aceptar el autómata:

- REPU-2023-12345-7
- REPU-2022-67890-3
- REPU-2021-00001-5

### 1.3.2. Cadenas inválidas

Ejemplo de cadenas no válidas para autómatas:

- APU-1999-12345-7
- REPU-1999-12345-1
- REPU-2023-12345-43t

## 1.4. Restricciones

*Prefijo:* Debe iniciar con las letras *REPU*", que son las siglas del Reclusorio Público.

*Año:* Después del prefijo, debe incluir cuatro dígitos representando un año.

*Sección:* A continuación, debe tener una secuencia de cinco dígitos que identifica la sección del recluso.

*Identificador de recluso:* Por último, el identificador termina con un solo dígito que corresponde al recluso en particular.

*Separación:* Cada sección está delimitada por un guion (").

## Capítulo 2

# AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto

Periodo:	2024.C3 Septiembre - diciembre de 2024
Grupo:	7A/B
Asignatura:	Lenguajes y autómatas
Corte:	1
Actividad:	2024.C3.07X.L&A.C1.A3
Título:	AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto
Fecha de entrega:	2024.09.24
<b>221189</b>	<b>Luis Albertoberto Batalla</b>

### 2.1. Descripción de la actividad

Diseñar un AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto.

1. Usar como problema a resolver, el propuesto en [1].
2. Presentar el AFD en la especificación formal de grafos indicada en clase.
3. Presentar el AFD en forma gráfica.

### 2.2. RepuIDValidator: Validador de identificadores del Reclusorio Preventivo (REPU) que asegura el cumplimiento del formato establecido.

### 2.3. Descripción

El sistema implementará un autómata finito determinista que hará: *Indicar en las viñetas lo que hará*

El autómata recibirá una cadena de entrada y analizará si la cadena corresponde a un identificador del Reclusorio Preventivo, que debe seguir las siguientes reglas: El identificador comenzará con el prefijo "REPU", seguido de un guion. Luego debe contener el año de ingreso, representado por 4 dígitos. A continuación, debe haber un guion seguido de un número secuencial de 5 dígitos. Finalmente, debe terminar con un guion y un dígito verificador (0-9). Si la cadena no es válida, el autómata se reiniciará tomando como entrada la cadena a partir de la posición siguiente a donde comenzó. Si encuentra un identificador válido, registrará la posición y la longitud del identificador válido y se reiniciará a partir de la posición siguiente al final del identificador válido.

### 2.3.1. Cadenas válidas

Ejemplo de cadenas válidas que debe aceptar el autómata:

- REPU-2023-12345-7
- REPU-2022-67890-3
- REPU-2021-00001-5

### 2.3.2. Cadenas inválidas

Ejemplo de cadenas no válidas para autómata:

- APU-1999-12345-7
- REPU-1999-12345-1
- REPU-2023-12345-43t

### 2.3.3. Restricciones

*Prefijo:* Debe iniciar con las letras REPU", que son las siglas del Reclusorio Público.

*Año:* Después del prefijo, debe incluir cuatro dígitos representando un año.

*Sección:* A continuación, debe tener una secuencia de cinco dígitos que identifica la sección del recluso.

*Identificador de recluso:* Por último, el identificador termina con un solo dígito que corresponde al recluso en particular.

*Separación:* Cada sección está delimitada por un guion (").

## 2.4. Diseño

### 2.4.1. Especificación formal

Para resolver el problema se diseñó el autómata finito determinista  $D = (Q, \Sigma, q_0, \delta, F)$ , donde:

- $Q = \{qe, q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_f\}$ ,
- $\Sigma = \{R, E, P, U, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$ ,
- $q_0 = q_0$ ,
- $\delta = \{(q_0, R) \rightarrow q_1, (q_0, \sigma \neq R) \rightarrow qe,$   
 $(q_1, E) \rightarrow q_2, (q_1, \sigma \neq E) \rightarrow qe,$   
 $(q_2, P) \rightarrow q_3, (q_2, \sigma \neq P) \rightarrow qe,$   
 $(q_3, U) \rightarrow q_4, (q_3, \sigma \neq U) \rightarrow qe,$   
 $(q_4, -) \rightarrow q_5, (q_4, \sigma \neq -) \rightarrow qe,$   
 $(q_5, 2) \rightarrow q_6, (q_5, \sigma \neq 2) \rightarrow qe,$   
 $(q_6, 0) \rightarrow q_7, (q_6, \sigma \neq 0) \rightarrow qe,$   
 $(q_7, d \in \{0, \dots, 9\}) \rightarrow q_8, (q_7, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_8, d \in \{0, \dots, 9\}) \rightarrow q_9, (q_8, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_9, -) \rightarrow q_{10}, (q_9, \sigma \neq -) \rightarrow qe,$   
 $(q_{10}, d \in \{0, \dots, 9\}) \rightarrow q_{11}, (q_{10}, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_{11}, d \in \{0, \dots, 9\}) \rightarrow q_{12}, (q_{11}, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_{12}, d \in \{0, \dots, 9\}) \rightarrow q_{13}, (q_{12}, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_{13}, d \in \{0, \dots, 9\}) \rightarrow q_{14}, (q_{13}, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_{14}, d \in \{0, \dots, 9\}) \rightarrow q_{15}, (q_{14}, \sigma \notin \{0, \dots, 9\}) \rightarrow qe,$   
 $(q_{15}, -) \rightarrow q_{16}, (q_{15}, \sigma \neq -) \rightarrow qe,$   
 $(q_{16}, d \in \{0, \dots, 9\}) \rightarrow q_f, (q_{16}, \sigma \notin \{0, \dots, 9\}) \rightarrow qe\}$ , y
- $F = \{q_f\}$ .

También pueden optar por presentar  $\delta$  a través de una matriz de transición como se indica en la Tabla 2.4.1.

Tabla 2.1: Matriz de transición del autómata finito determinista RepUIDValidator.

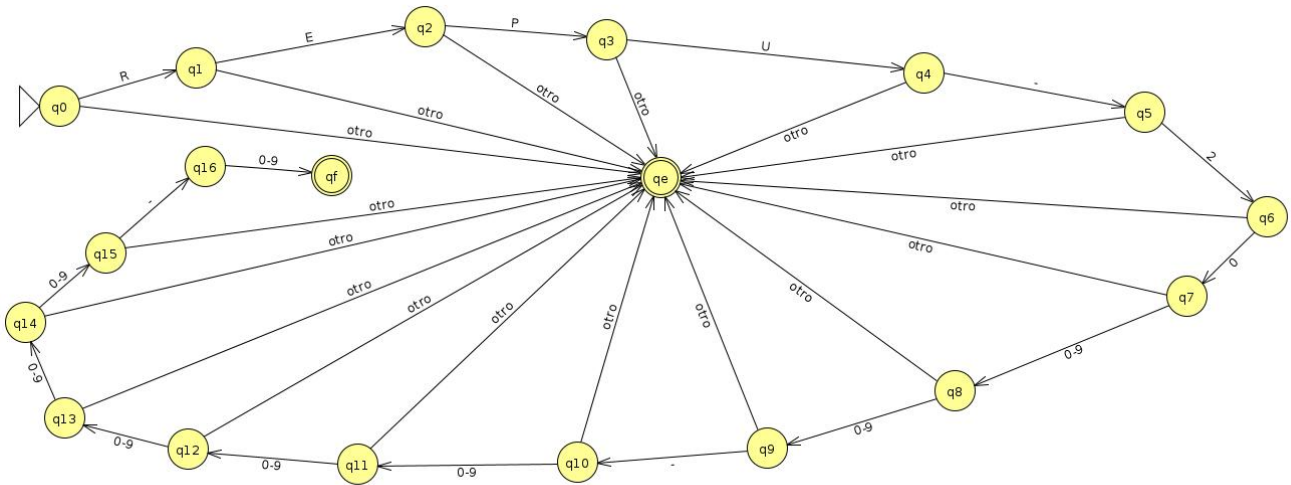
Estado	Símbolo válido	Transición
$\rightarrow q_0$	R	$q_1$
$q_0$	$\sigma \neq R$	$qe$
$q_1$	E	$q_2$
$q_1$	$\sigma \neq E$	$qe$
$q_2$	P	$q_3$
$q_2$	$\sigma \neq P$	$qe$
$q_3$	U	$q_4$
$q_3$	$\sigma \neq U$	$qe$
$q_4$	-	$q_5$
$q_4$	$\sigma \neq -$	$qe$
$q_5$	2	$q_6$
$q_5$	$\sigma \neq 2$	$qe$
$q_6$	0	$q_7$
$q_6$	$\sigma \neq 0$	$qe$
$q_7$	$d \in \{0, \dots, 9\}$	$q_8$
$q_7$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_8$	$d \in \{0, \dots, 9\}$	$q_9$
$q_8$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_9$	-	$q_{10}$
$q_9$	$\sigma \neq -$	$qe$
$q_{10}$	$d \in \{0, \dots, 9\}$	$q_{11}$
$q_{10}$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_{11}$	$d \in \{0, \dots, 9\}$	$q_{12}$
$q_{11}$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_{12}$	$d \in \{0, \dots, 9\}$	$q_{13}$
$q_{12}$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_{13}$	$d \in \{0, \dots, 9\}$	$q_{14}$
$q_{13}$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_{14}$	$d \in \{0, \dots, 9\}$	$q_{15}$
$q_{14}$	$\sigma \notin \{0, \dots, 9\}$	$qe$
$q_{15}$	-	$q_{16}$
$q_{15}$	$\sigma \neq -$	$qe$
$q_{16}$	$d \in \{0, \dots, 9\}$	$q_f$
$q_{16}$	$\sigma \notin \{0, \dots, 9\}$	$qe$

### 2.4.2. Diagrama

En la Figura 2.4.2, se presenta el diagrama del autómata finito  $D$  diseñado para resolver el problema.



Figura 2.1: Diagrama del AFD  $D$ .



## Capítulo 3

# Implementación de AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto

Periodo:	2024.C3 Septiembre - diciembre de 2024
Grupo:	7A/B
Asignatura:	Lenguajes y autómatas
Corte:	1
Actividad:	2024.C3.07X.L&A.C1.A4
Título:	Implementación de AFD para analizar y extraer patrones de cadenas en el cuerpo de un texto
Fecha de entrega:	2024.09.24
<b>221189</b>	<b>Luis Alberto Batalla Gonzales</b>

### 3.1. Descripción de la actividad

Implementar el AFD diseñado, la implementación debe permitir:

1. Leer la especificación del AFD en la notación formal.
2. Leer un archivo que contiene los patrones de texto a buscar, se permite: xlsx, csv, docx o html.
3. Reportar las ocurrencias, incluir fila-columna/posición y el texto de la ocurrencia.
4. Reportar la salida en un archivo txt.

### 3.2. RepuIDValidator:

- Validador de identificadores del Reclusorio Preventivo (REPU) que asegura el cumplimiento del formato establecido.

### 3.3. Descripción

Autómata que valida los identificadores del reclusorio preventivo, para asegurar que complan con el formato establecido: *Indicar en las viñetas lo que hará*

### 3.3.1. Cadenas válidas

Ejemplo de cadenas válidas que debe aceptar el autómata:

- REPU-2025-12345-0
- REPU-2023-12345-7
- REPU-2023-55555-9

### 3.3.2. Cadenas inválidas

Ejemplo de cadenas no válidas para autómata:

- APU-1234-99-1212
- REPU-987-98765-2
- REPU-12345-12345-12345

### 3.3.3. Restricciones

*Indicar con viñetas las restricciones que se consideren.*

- Prefijo: Debe iniciar con las letras REPU", que son las siglas del Reclusorio Público.
- Año: Después del prefijo, debe incluir cuatro dígitos representando un año.
- Sección: A continuación, debe tener una secuencia de cinco dígitos que identifica la sección del recluso.
- Identificador de recluso: Por último, el identificador termina con un solo dígito que corresponde al recluso en particular.
- Separación: Cada sección está delimitada por un guion (").

### 3.3.4. Diseño

#### Especificación formal

Para resolver el problema se diseñó el autómata finito determinista  $D = (Q, \Sigma, q_0, \delta, F)$ , donde:

- $Q = \{q_e, q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_f\}$ ,
- $\Sigma = \{R, E, P, U, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$ ,
- $q_0 = q_0$ ,
- $\delta = \{(q_0, R, q_1), (q_0, \sigma \neq R, q_e), (q_1, E, q_2), (q_1, \sigma \neq E, q_e), (q_2, P, q_3), (q_2, \sigma \neq P, q_e), (q_3, U, q_4), (q_3, \sigma \neq U, q_e), (q_4, -, q_5), (q_4, -, q_e), (q_5, 2, q_6), (q_5, \sigma \neq 2, q_e), (q_6, 0, q_7), (q_6, \sigma \neq 0, q_e), (q_7, d \in \{0-9\}, q_8), (q_7, \sigma \notin \{0-9\}, q_e), (q_8, d \in \{0-9\}, q_9), (q_8, \sigma \notin \{0-9\}, q_e), (q_9, -, q_{10}), (q_9, \sigma \neq -, q_e), (q_{10}, d \in \{0-9\}, q_{11}), (q_{10}, \sigma \notin \{0-9\}, q_e), (q_{11}, d \in \{0-9\}, q_{12}), (q_{11}, \sigma \notin \{0-9\}, q_e), (q_{12}, d \in \{0-9\}, q_{13}), (q_{12}, \sigma \notin \{0-9\}, q_e), (q_{13}, d \in \{0-9\}, q_{14}), (q_{13}, \sigma \notin \{0-9\}, q_e), (q_{14}, d \in \{0-9\}, q_{15}), (q_{14}, \sigma \notin \{0-9\}, q_e), (q_{15}, -, q_{16}), (q_{15}, \sigma \neq -, q_e), (q_{16}, d \in \{0-9\}, q_f), (q_{16}, \sigma \notin \{0-9\}, q_e)\}$ , y
- $F = \{q_f\}$ .

También pueden optar por presentar  $\delta$  a través de una matriz de transición como se indica en la Tabla ??.

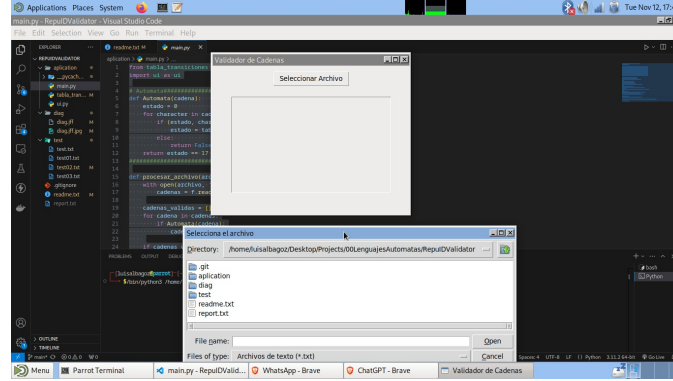
#### Diagrama

En la Figura 3.3.4, se presenta el diagrama del autómata finito  $D$  diseñado para resolver el problema.

## 3.4. Implementación

La implementación del AFD se realizó en Python.

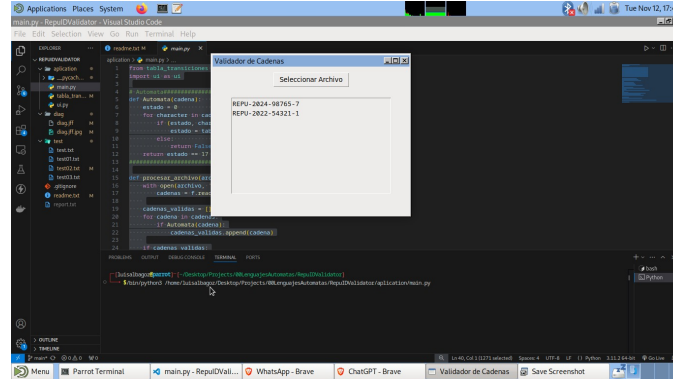
Figura 3.1: Diagrama del AFD  $D$ .



### 3.4.1. Interfaz

La Figura 3.4.1 muestra la interfaz gráfica diseñada para mostrar el funcionamiento del sistema. *Describir de manera muy resumida los elementos de la interfaz y las interacciones.*

Figura 3.2: Interfaz gráfica del sistema que implementa al AFD  $D$ .



### 3.4.2. Función de transición

La funcionalidad principal del AFD está en la función de transición, la cual se muestra en el Listado 3.1.

```
1  tabla = {
2      (0, 'R'): 1,
3      (1, 'E'): 2,
4      (2, 'P'): 3,
5      (3, 'U'): 4,
6
7      (4, '-'): 5,
8
9      (5, '2'): 6,
10
11     (6, '0'): 7,
12
13     (7, '0'): 8,
14     (7, '1'): 8,
15     (7, '2'): 8,
16     (7, '3'): 8,
17     (7, '4'): 8,
18     (7, '5'): 8,
19     (7, '6'): 8,
20     (7, '7'): 8,
21     (7, '8'): 8,
22     (7, '9'): 8,
23
24     (8, '0'): 9,
```

```

25 (8, '1'): 9,
26 (8, '2'): 9,
27 (8, '3'): 9,
28 (8, '4'): 9,
29 (8, '5'): 9,
30 (8, '6'): 9,
31 (8, '7'): 9,
32 (8, '8'): 9,
33 (8, '9'): 9,
34
35 (9, '-'): 10,
36
37 (10, '0'): 11,
38 (10, '1'): 11,
39 (10, '2'): 11,
40 (10, '3'): 11,
41 (10, '4'): 11,
42 (10, '5'): 11,
43 (10, '6'): 11,
44 (10, '7'): 11,
45 (10, '8'): 11,
46 (10, '9'): 11,
47
48 (11, '0'): 12,
49 (11, '1'): 12,
50 (11, '2'): 12,
51 (11, '3'): 12,
52 (11, '4'): 12,
53 (11, '5'): 12,
54 (11, '6'): 12,
55 (11, '7'): 12,
56 (11, '8'): 12,
57 (11, '9'): 12,
58
59 (12, '0'): 13,
60 (12, '1'): 13,
61 (12, '2'): 13,
62 (12, '3'): 13,
63 (12, '4'): 13,
64 (12, '5'): 13,
65 (12, '6'): 13,
66 (12, '7'): 13,
67 (12, '8'): 13,
68 (12, '9'): 13,
69
70 (13, '0'): 14,
71 (13, '1'): 14,
72 (13, '2'): 14,
73 (13, '3'): 14,
74 (13, '4'): 14,
75 (13, '5'): 14,
76 (13, '6'): 14,
77 (13, '7'): 14,
78 (13, '8'): 14,
79 (13, '9'): 14,
80
81 (14, '0'): 15,
82 (14, '1'): 15,
83 (14, '2'): 15,
84 (14, '3'): 15,
85 (14, '4'): 15,
86 (14, '5'): 15,
87 (14, '6'): 15,
88 (14, '7'): 15,
89 (14, '8'): 15,
90 (14, '9'): 15,
91
92 (15, '-'): 16,
93
94 (16, '0'): 17,
95 (16, '1'): 17,
96 (16, '2'): 17,

```

```

97     (16, '3'): 17,
98     (16, '4'): 17,
99     (16, '5'): 17,
100    (16, '6'): 17,
101    (16, '7'): 17,
102    (16, '8'): 17,
103    (16, '9'): 17,
104 }

```

Listing 3.1: Ejemplo de código insertado

Otra funcionalidad interesante es el llamado repetitivo del AFD, desde el principio y hasta el final del texto como se muestra en el Listado 3.2.

```

1  from tabla_transiciones import tabla
2  import ui as ui
3
4  # Automata#####
5  def Automata(cadena):
6      estado = 0
7      for character in cadena:
8          if (estado, character) in tabla:
9              estado = tabla[(estado, character)]
10         else:
11             return False # qe
12     return estado == 17 # qf
13 #####
14
15 def procesar_archivo(archivo):
16     with open(archivo, 'r') as f:
17         cadenas = f.read().splitlines()
18
19     cadenas_validas = []
20     for cadena in cadenas:
21         if Automata(cadena):
22             cadenas_validas.append(cadena)
23
24     if cadenas_validas:
25         resultado = "\n".join(cadenas_validas)
26     else:
27         resultado = "No se encontraron cadenas v lidas."
28
29     # reporte
30     with open("report.txt", "a") as report_file:
31         for cadena in cadenas_validas:
32             report_file.write(cadena + "\n")
33
34     # imprimir
35     mostrar_resultado_callback(resultado)
36
37 root, mostrar_resultado_callback = ui.crear_interfaz(procesar_archivo)
38
39 root.mainloop()

```

Listing 3.2: Ejemplo de código insertado

### 3.4.3. Pruebas realizadas

Agregar figuras donde se evidencie la carga del archivo, el resultado de la ejecución y la exportación. Reportar al menos tres conjuntos de imágenes.

## 3.5. Comentarios

Este proyecto ha permitido entender cómo los autómatas finitos deterministas (AFD) se pueden aplicar en la validación de cadenas, como en el análisis de datos o validación de formatos. A través del diseño de la tabla de transiciones y la ejecución del AFD, se pudo verificar su efectividad. Este conocimiento es útil para proyectos futuros en los que se necesite procesar datos o crear validadores, especialmente cuando se busca garantizar que las entradas sigan una estructura predefinida.

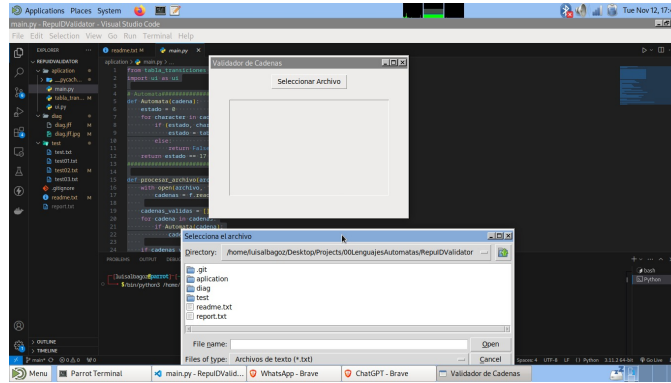


Figura 3.3: Captura de ejecución 1 *D*.

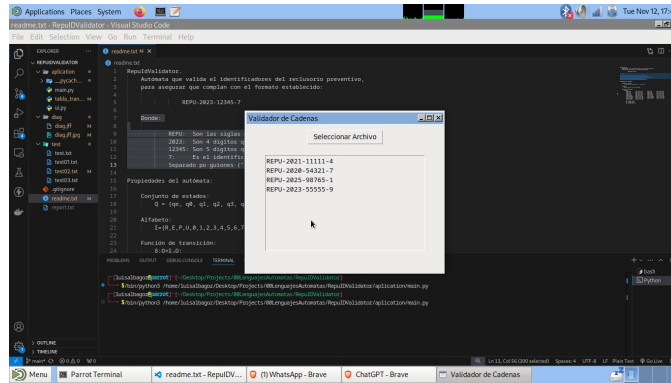


Figura 3.4: Captura de ejecución 2 *D*.

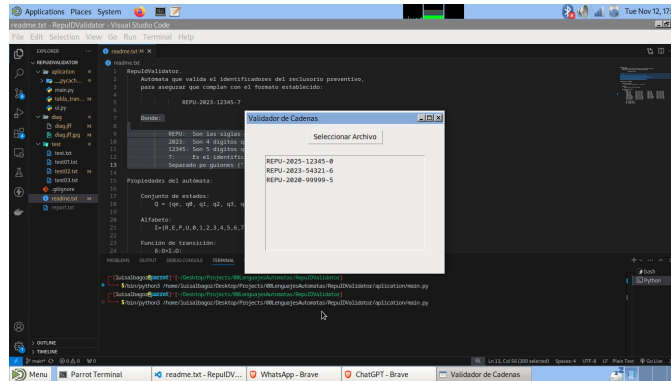


Figura 3.5: Captura de ejecución 3 *D*.

# Bibliografía

- [1] Clase LA. Listado de propuestas de aplicativos de autómatas finitos deterministas, 2024. <https://docs.google.com/spreadsheets/d/1kUtUhhcjtfIzn2-osH2RwJ9zC-gWwRV4wcR9Xu6v9kw/>. Última consulta 2024.10.04.