

## Tabla Informativa

|                      |                                 |
|----------------------|---------------------------------|
| Periodo              | Enero - Abril 2023              |
| Cuatrimestre y grupo | 7°A                             |
| Asignatura           | Lenguajes y Autómatas           |
| Corte                | 1                               |
| Actividad            | L&A.C1.A3 Proyecto              |
| Fecha de asignación  | 2023.01.18                      |
| Fecha de entrega     | 2023.01.31                      |
| Matrícula            | Nombres                         |
| 193241               | OLTRA LEÓN Francisco Erick      |
| 211099               | MENDEZ DÍAZ Mario Alfredo       |
| 211105               | MARTÍNEZ MARROQUÍN Luis Antonio |

## Tabla de contenido

|                                  |    |
|----------------------------------|----|
| Tabla Informativa .....          | 2  |
| Introducción .....               | 6  |
| Problemática .....               | 7  |
| Modelado .....                   | 8  |
| Descripción del Autómata. ....   | 9  |
| Implementación .....             | 10 |
| Resultados .....                 | 12 |
| Discusión .....                  | 14 |
| Conclusión .....                 | 15 |
| Referencias Bibliográficas ..... | 16 |

## Tabla de Ilustraciones

|   |    |
|---|----|
| Ilustración 1. Modelado del autómata para validar formato de fecha..... | 8  |
| Ilustración 2. Vista principal. ....                                    | 10 |
| Ilustración 3. Selección de archivo.....                                | 11 |
| Ilustración 4. Archivo convertido a TXT. ....                           | 12 |
| Ilustración 5. Mensajes en consola.....                                 | 12 |
| Ilustración 6. Visualización de las fechas validadas.....               | 13 |

## Índice de Tablas.

|  |   |
|--|---|
| <i>Tabla 1.</i> Tabla de formatos de fecha. .... | 6 |
|--|---|

## Introducción

Dentro de la teoría de autómatas finitos, hemos definido en trabajos anteriores los diversos conceptos que nos ayudan a adquirir una mejor comprensión acerca del tema. Retomando los puntos más importantes, definiremos a un *Autómata*, como aquel modelo de cálculos abstractos que, de acuerdo con una entrada de diferentes símbolos, podrán realizar transiciones entre los diferentes estados existentes.

Normalmente, las personas no se encuentran familiarizadas con estos conceptos, por lo que llegan a desconocer las ventajas que podrían obtener implementando *Autómatas* computacionales que faciliten las diversas tareas que deben realizar, entre las más comunes podemos mencionar la clasificación de un artículo, validación de datos, consulta de dominios públicos disponibles, entre otros.

Sin embargo, el desarrollo de este proyecto será enfocado a un campo con oportunidades de mejora para la correcta gestión de su información, por lo cual, vamos a implementar el uso de autómatas a un caso, que será detallado a grandes rasgos en el planteamiento de la **Problemática**.

## Problemática

Es muy común encontrarnos con situaciones en donde nuestro trabajo no es realizado de la mejor manera, ¿a qué nos referimos con ello? Bien, existen departamentos cuyas funciones requieren tener mucha información mal documentada, dispersa, o que no tienen la capacitación y/o herramientas adecuadas para filtrar o validar datos. Dentro de esta problemática, nos concentraremos con los formatos de fechas situados en los documentos PDF, algunos son colocados de tal manera que no son válidos para la comprensión humana o una computadora, por lo que es un problema cuando se trata de documentos importantes los cuales la fecha es un factor muy importante.

Nos dimos a la tarea de generar las siguientes tablas para poder proporcionar una mejor comprensión acerca de cómo están estructurados los formatos de fecha.

|                         |                     |
|-------------------------|---------------------|
| mm/dd/aaaa              | 02/21/2012          |
| mm-dd-aaaa              | 02-21-2012          |
| mm/dd/aa                | 02/21/12            |
| mm-dd-aa                | 02-21-12            |
| dd/mm/aaaa              | 21/02/2012          |
| dd-mm-aaaa              | 21-02-2012          |
| dd/mm/aa                | 21/02/12            |
| dd-mm-aa                | 03-02-22            |
| dd de diciembre de aaaa | 28 de marzo de 2002 |
| aaaa-mm-dd              | 2012-02-21          |

*Tabla 1. Formatos de fecha.*

AAAA es el año [todos los dígitos, p.ej. 2012]

MM es el mes [01 (Enero) hasta 12 (Diciembre)]

DD es el día [01 hasta 31]

Las letras en color rojo son los formatos que no son válidos para el programa, los cuales todas aquellas que tengan ese formato dentro de un documento PDF no las validará el autómata.

## Modelado

Tomando en cuenta la información respecto a la problemática para solventar, se modeló el siguiente autómata:

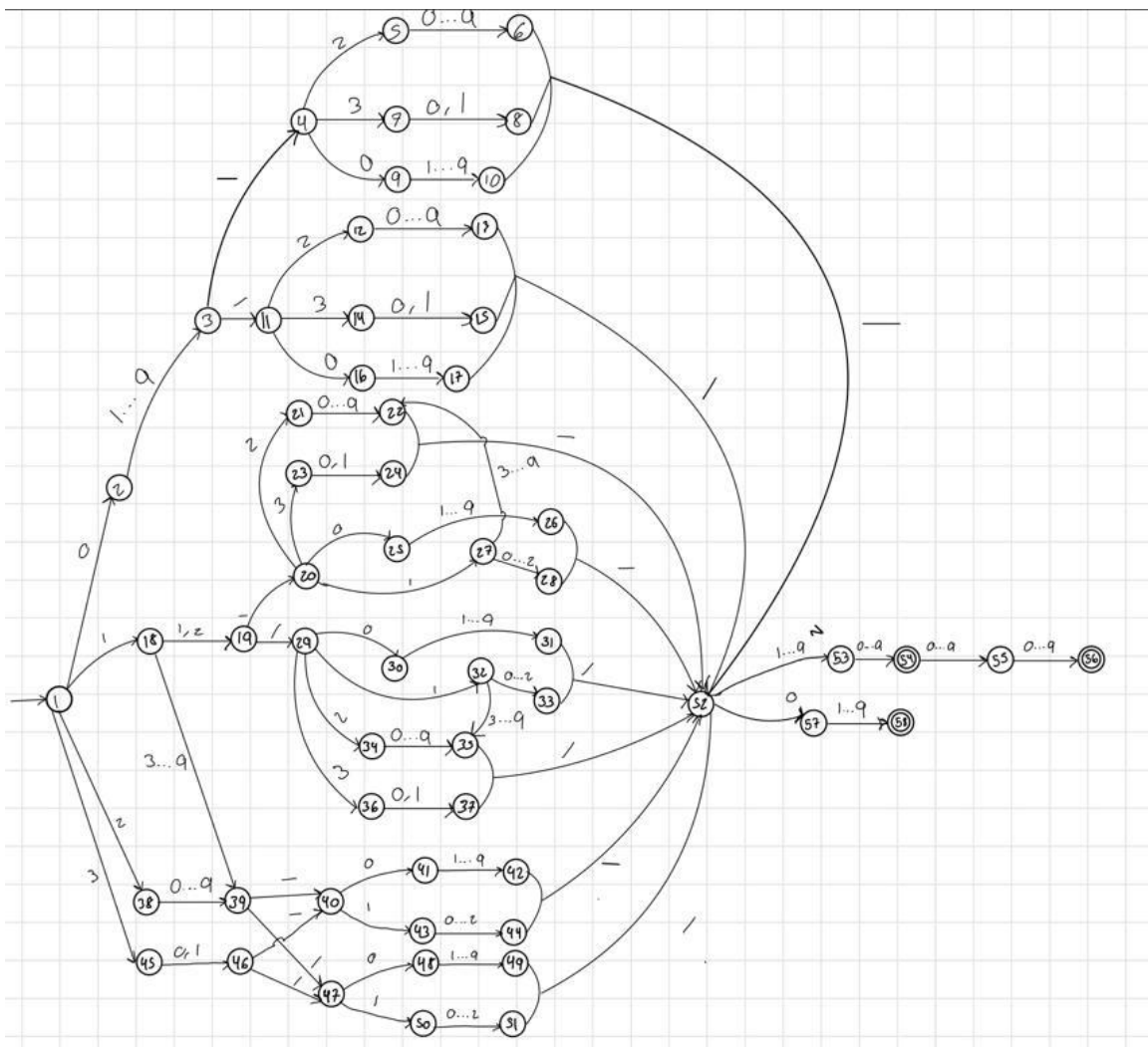


Ilustración 1. Modelado del autómata para validar formato de fecha.

## Descripción del Autómata.

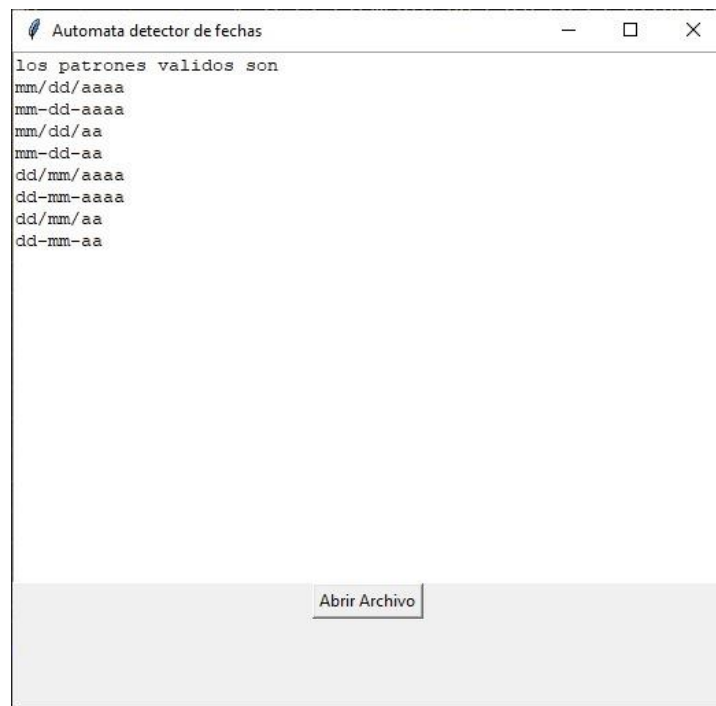
En la siguiente tabla, podrá observar la descripción del autómata diseñado:

|                             |   |
|-----------------------------|---|
| <b>q0:</b>                  | 1   |
| <b>Q:</b>                   | {1, ... 56}   |
| <b>F:</b>                   | {54, 56, 58}  |
| <b><math>\Sigma</math>:</b> | {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, /}  |
| <b><math>\delta</math>:</b> | <p>{(1, 0, 2), (1, 1, 18), (1, 2, 38), (1, 3, 45), (2, [1, 2, 3, 4, 5, 6, 7, 8, 9], 3), (3, -, 4), (3, /, 11), (4, 2, 5), (4, 3, 7), (4, 0, 9), (5, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 6), (6, -, 52), (7, [0, 1], 8), (8, -, 52), (9, [1, 2, 3, 4, 5, 6, 7, 8, 9], 10), (10, -, 52), (11, 2, 12), (11, 3, 14), (11, 0, 16), (12, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 13), (13, /, 52), (14, [0, 1], 15), (15, /, 52), (16, [1, 2, 3, 4, 5, 6, 7, 8, 9], 17), (17, /, 52), (18, [1, 2], 19), (18, [3, 4, 5, 6, 7, 8, 9], 39), (19, -, 20), (19, /, 29), (20, 2, 21), (20, 3, 23), (20, 0, 25), (20, 1, 27), (21, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 22), (22, -, 52), (23, [0, 1], 24), (24, -, 52), (25, [1, 2, 3, 4, 5, 6, 7, 8, 9], 26), (26, -, 52), (27, [0, 1, 2], 28), (27, [3, 4, 5, 6, 7, 8, 9], 22), (28, -, 52), (29, 0, 30), (29, 1, 32), (29, 2, 34), (29, 3, 36), (30, [1, 2, 3, 4, 5, 6, 7, 8, 9], 31), (31, /, 52), (32, [0, 1, 2], 33), (32, [3, 4, 5, 6, 7, 8, 9], 35), (33, /, 52), (34, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 35), (35, /, 52), (36, [0, 1], 37), (37, /, 52), (38, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 39), (39, -, 40), (39, /, 47), (40, 0, 41), (40, 1, 43), (41, [1, 2, 3, 4, 5, 6, 7, 8, 9], 42), (42, -, 52), (43, [0, 1, 2], 44), (44, -, 52), (45, [0, 1], 46), (46, -, 40), (46, /, 47), (47, 0, 48), (47, 1, 50), (48, [1, 2, 3, 4, 5, 6, 7, 8, 9], 49), (49, /, 52), (50, [0, 1, 2], 51), (51, /, 52), (52, [1, 2, 3, 4, 5, 6, 7, 8, 9], 53), (52, 0, 57), (53, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 54), (54, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 55), (55, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], 56), (57, [1, 2, 3, 4, 5, 6, 7, 8, 9], 58)}</p> |



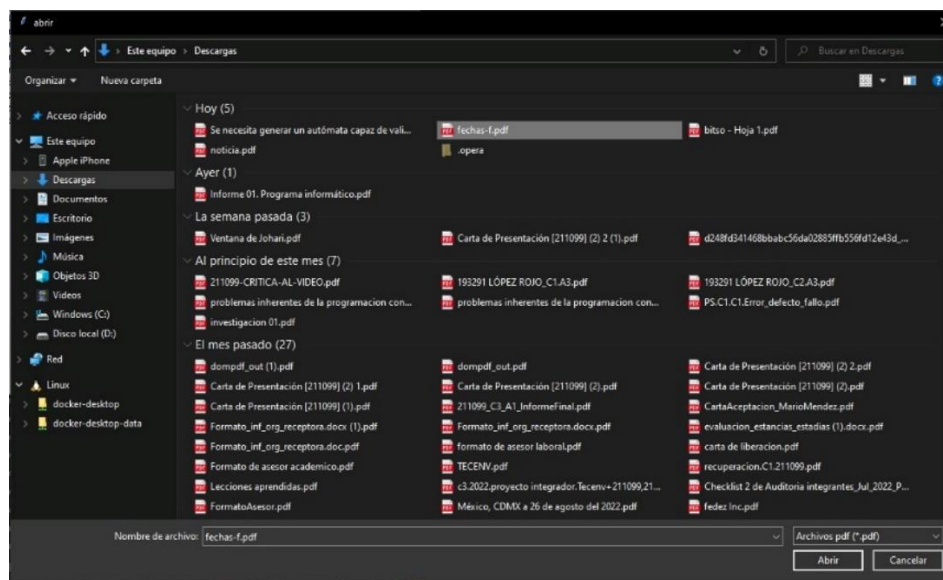
## Implementación

Para poder desarrollar este proyecto utilizamos el lenguaje de programación Python debido a su simplicidad al momento de ejecutar diversas funciones y tareas, además de sus amplias librerías. Tal y como se ha mencionado en anteriores reportes de actividades este programa requiere de un archivo de texto en formato PDF para poder funcionar óptimamente y para poder realizar los respectivos procesos. Para poder llevar el proyecto a una interfaz gráfica usamos las librerías de Tkinter para de esta forma usar elementos como botones, inputs, mensajes que nos ayudara a verificar el comportamiento de los estados y las transiciones que realizaba al paso que iba. Lo primero era seleccionar el PDF con la interfaz gráfica del programa



*Ilustración 2. Vista principal.*

Una vez presionado el botón el programa accedía a leer los archivos de nuestro ordenador y seleccionar el archivo deseado con formato PDF



*Ilustración 3. Selección de archivo.*

## Resultados

Gracias a la propiedad de PdfFileReader nos fue posible hacer la lectura del PDF para posteriormente convertirlo en un txt que nos permita recorrer todo el archivo y así localizar los formatos de fecha en cada cadena de palabras o símbolos.

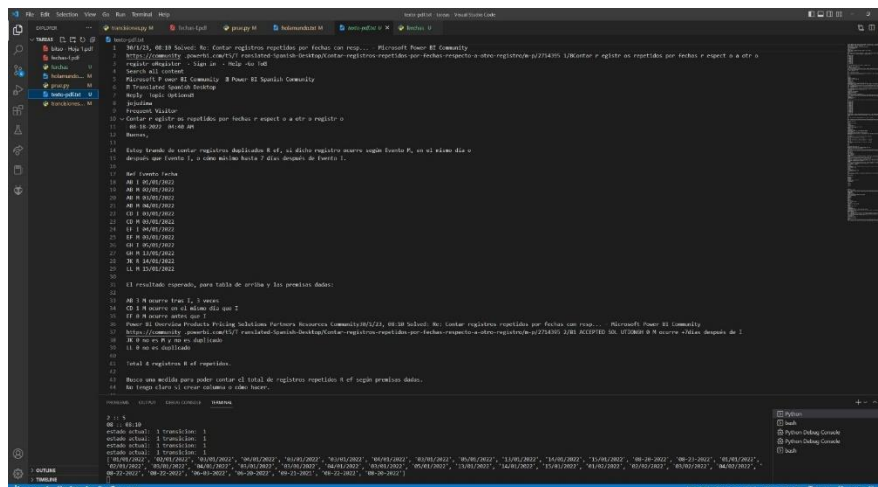


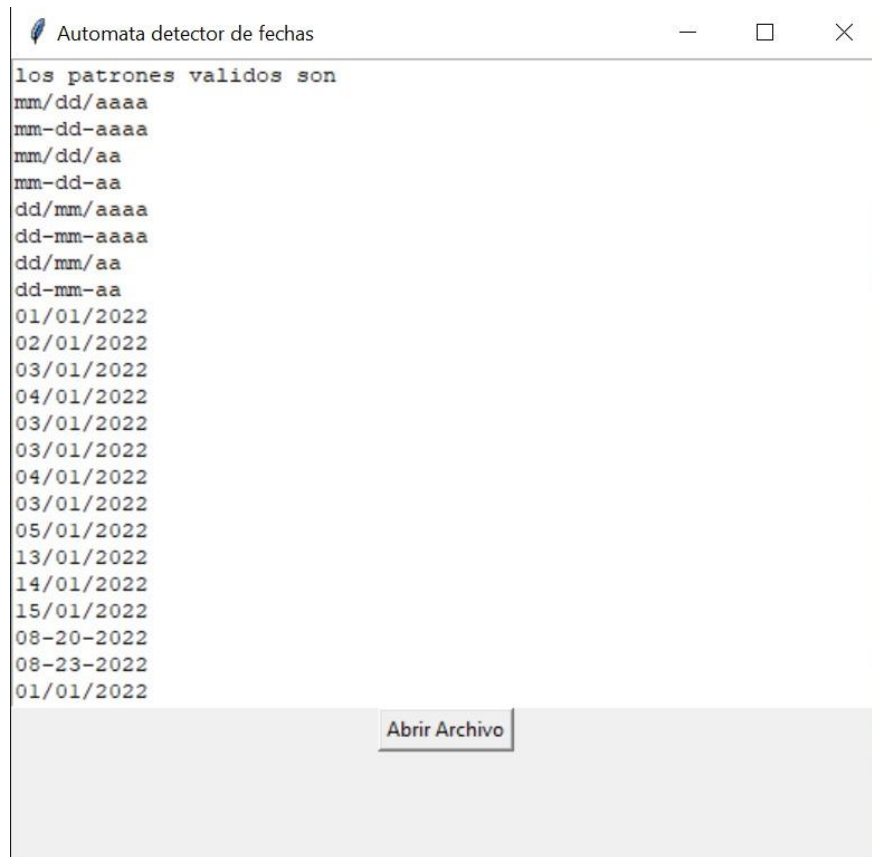
Ilustración 4. Archivo convertido a TXT.

Posteriormente el archivo hecho txt, adaptamos el programa para que hiciera reconocimiento de cada cadena de palabras hasta identificar el primer estado de nuestro autómata para a continuación seguir leyendo los siguientes estados de todas las cadenas dentro del txt e irlos almacenando los que cumplieran con las condiciones que planteamos. Para ellos usamos mensajes en consola que nos ayudaran a comprender el comportamiento del autómata y si algún método cometía error durante la lectura de todo el archivo

```
estado actual: 1 transicion: 1
estado actual: 2 transicion: 2
estado actual: 3 transicion: 3
estado actual: 4 transicion: 4
estado actual: 5 transicion: 5
estado actual: 6 transicion: 6
estado actual: 52 transicion: 52
estado actual: 53 transicion: 53
estado actual: 54 transicion: 54
estado actual: 55 transicion: 55
estado final: 56
cadena valida: 08-20-2022
estado actual: 1 transicion: 1
estado actual: 2 transicion: 2
estado actual: 1 transicion: 1
estado actual: 1 transicion: 1
estado actual: 1 transicion: 1
estado actual: 2 transicion: 2
estado actual: 1 transicion: 1
estado actual: 1 transicion: 1
estado actual: 1 transicion: 1
estado actual: 1 transicion: 1
['01/01/2022', '02/01/2022', '03/01/2022', '04/01/2022', '03/01/2022', '03/01/2022', '04/01/2022', '03/01/2022', '05/01/2022', '13/01/2022',
'02/01/2022', '03/01/2022', '04/01/2022', '03/01/2022', '03/01/2022', '04/01/2022', '05/01/2022', '13/01/2022', '14/01/2022',
'08-22-2022', '08-22-2022', '06-03-2022', '06-20-2022', '09-21-2021', '08-22-2022', '08-20-2022']
```

Ilustración 5. Mensajes en consola.

Y por último todas aquellas cadenas de palabras que eran validadas por el autómata las iba almacenando en un array para posteriormente mostrarla en la vista principal con únicamente los formatos de fecha válidos.



*Ilustración 6. Visualización de las fechas validadas.*

## Discusión

Al Iniciando con el análisis del *Autómata*, el equipo pudo percatarse que este tenía 3 estados finales {54, 56, 58} lo cual generó curiosidad de que las transiciones únicamente podían acabar ahí al tener en cuenta que hay muchas posibilidades de fechas, sin embargo, tomando en cuenta el formato fue establecido para que lo leyera el autómata y validara, no habría de otra forma que no terminara en esos 3. Gracias a los ejercicios y análisis de Autómatas, fue más sencillo desarrollar esta actividad, pues nuestra mayor concentración se centró en el modelado correcto del autómata para asegurarse de que este lograba solventar la problemática planteada.

## Conclusión

Concluyendo el proyecto, el equipo rescata que los *Autómatas*, son de gran utilidad para ser aplicados en diferentes situaciones, ya que, en este caso en específico, nos ayuda a definir dentro de un algoritmo la forma más práctica de resolver una problemática, es decir, gracias a un correcto modelado, es más simple enfocarse a cumplir con la funcionalidad solicitada.

A pesar de encontrarnos con diferentes dificultades, como la de investigar y asegurarse de que la información que fue presentada es válida y oficial, hasta la de desarrollar el autómata y describirlo con el debido cuidado, el equipo puede concluir que se ha logrado llegar al objetivo. Además, el haber trabajado anteriormente con actividades de practica que implican el diseño, modelado e implementación de *Autómatas*, ha permitido un gran desarrollo y entendimiento hacia el tema.

## Referencias Bibliográficas

“Introducción a los Autómatas.” [Online]. Available: <https://ccc.inaoep.mx/~emorales/Cursos/Automatas/AutomatasFinitos.pdf>.

Ganesh Gopalakrishnan, Automata and computability: programmer's perspective. Boca Raton, FL: Crc Press, Taylor & Francis Group, 2019.

J. E. Hopcroft, Rajeev Motwani, and J. D. Ullman, Introducción a la teoría de autómatas, lenguajes y computación (3a. ed.). Madrid: Pearson Educación, 2007.

“SEP | SIGED,” Sep.gob.mx, 2016. <https://www.siged.sep.gob.mx/SIGED/escuelas.html> (accessed Sep. 30, 2021).