

Universidad Rafael Landívar  
Lenguajes Formales y Autómatas  
Área de ingeniería en informática y sistemas  
Ing. Juan Carlos Soto Santiago



## PROYECTO #1: “LECTOR DE ADF”

Diego Jeancarlo Cosillo Ramos 1136222

José Daniel Alvarado Zapata 1047222

Guatemala 13 de abril de 2024

## **INTRODUCCION**

Este manual está diseñado para proporcionarle toda la información necesaria para utilizar esta herramienta de manera efectiva y comprender su funcionamiento.

Este Validador de Cadenas con Autómatas Finitos Deterministas (DFA) es una aplicación de consola que te permite validar cadenas de entrada de acuerdo con las reglas definidas en un autómata finito determinista. Este programa es útil para verificar si una cadena cumple con las especificaciones de un autómata dado.

## ANÁLISIS Y DISEÑO

Este código en C# es un programa que lee un archivo de texto que define un autómata finito determinista (DFA) y luego valida cadenas de entrada según las reglas definidas por el autómata. Aquí está el análisis y diseño del código:

- Análisis:

1. Lectura del archivo de definición del autómata:

La función `ReadAutomatonFile` toma la ruta del archivo como entrada y devuelve un arreglo de strings que contiene el nombre del archivo, el estado inicial y el estado final del autómata. Esta función verifica si el archivo existe, si tiene el formato esperado y luego extrae la información necesaria.

Validación de la cadena de entrada:

La función `ValidateInput` toma la cadena de entrada y la ruta del archivo como entrada. Lee las líneas del archivo para obtener información sobre el número de estados y las transiciones. Luego, utiliza estas transiciones para simular el autómata, moviéndose de un estado a otro según los símbolos de la cadena de entrada. Finalmente, verifica si el estado final alcanzado es uno de los estados finales definidos en el archivo.

2. Main:

En el método `Main`, se maneja la lógica principal del programa. Se solicita al usuario que ingrese una cadena de entrada y se valida utilizando la función `ValidateInput`.

- Diseño:

1. División de responsabilidades:

El código está dividido en funciones que realizan tareas específicas: lectura del archivo, validación de la cadena de entrada y la lógica principal en el método `Main`. Esto facilita la comprensión y el mantenimiento del código.

2. Manejo de errores:

Se han implementado mecanismos para manejar errores como archivos faltantes o formatos incorrectos. Esto mejora la robustez del programa.

3. Uso de estructuras de datos:

Se utiliza una lista de arreglos de strings para almacenar las transiciones del autómata antes de convertirlas en una matriz. Esto simplifica la manipulación de las transiciones y facilita la verificación del formato.

## MANUAL DE USUARIO

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Ruta del archivo
        string filePath = @"C:\Users\Diego\Desktop\ProyectoAutomatas\Automata #1.txt";
    }
}
```

Antes de iniciar el programa debemos especificarle la dirección en la que se encuentra guardado el archivo .txt con el autómata que queremos ingresar y validar mediante el path donde esta almacenado, en este caso queremos analizar y validar el Autómata #1.txt: Ejemplo

“C:\Users\Diego\Desktop\ProyectoAutomatas\Automata #1.txt”



Una vez iniciado el programa lo primero que veremos será una ventana de consola indicándonos lo siguiente del autómata indicado en el archivo .txt:

1. El nombre del archivo .txt que incluye el autómata con el formato dado.  
Ejemplo:

```
AUTOMATA #1.txt - Notepad
File Edit Format View Help
4
q0
q3
q0, 1, q0
q0, 0, q1
q1, 0, q2
q2, 0, q3
q3, 1, q3
q3, 0, q1
```

De referencia se toma:

# de estados (n)  
# estado inicial (1..n)  
Conjunto de estados finales separados por comas  
Una línea por cada *transición* separando por comas: Estado Inicial (1..n), Cadena Leída, Estado

2. Se mostrará el estado inicial indicado en el archivo
3. Se mostrará el estado final indicado
4. El usuario debe ingresar una cadena para que el programa lo valide con el autómata subido, de no tener ninguna cadena para ingresar simplemente deberá ingresar la letra “q” para finalizar y cerrar el programa por completo

Una vez ingresada una cadena el programa mostrara si esta es valida o no lo es según se calcule:

```
C:\Users\Diego\Desktop\ProyectoAutomatas\Proyecto1_JoseAlvarado_DiegoCosillo\bin\Debug\net6.0\Proyecto1_JoseAlvarado_DiegoCosillo.exe
Nombre del archivo: Automata #1.txt
Estado inicial: q0
Estado final: q3
Ingresa la cadena a validar (o 'q' para salir): 10001
La cadena es aceptada por el autómata.
Ingresa la cadena a validar (o 'q' para salir):
```

Una vez mostrado el mensaje el programa automáticamente pedirá al usuario una nueva interacción ya sea una nueva cadena o el comando para finalizar el programa.

