

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y DE PROGRAMACIÓN
PRIMER SEMESTRE 2025



Proyecto 1

AFDGraph

Sección	Catedrático	Tutor Académico
A+	Ing. Otto Amílcar Rodríguez Acosta	Danny Hugo Bryan Tejaxún Pichiyá
A-	Inga. Vivian Damaris Campos González	Luisa María Ortíz Romero
B+	Ing. David Estuardo Morales Ajcót	Herberth Abisai Avila Ruiz
B-	Inga. Zulma Karina Aguirre Ordoñez	Jonatan Leonel García Arana

OBJETIVOS

General

- Desarrollar habilidades en Java para aplicación e implementación de conocimientos adquiridos sobre el análisis.

Específicos:

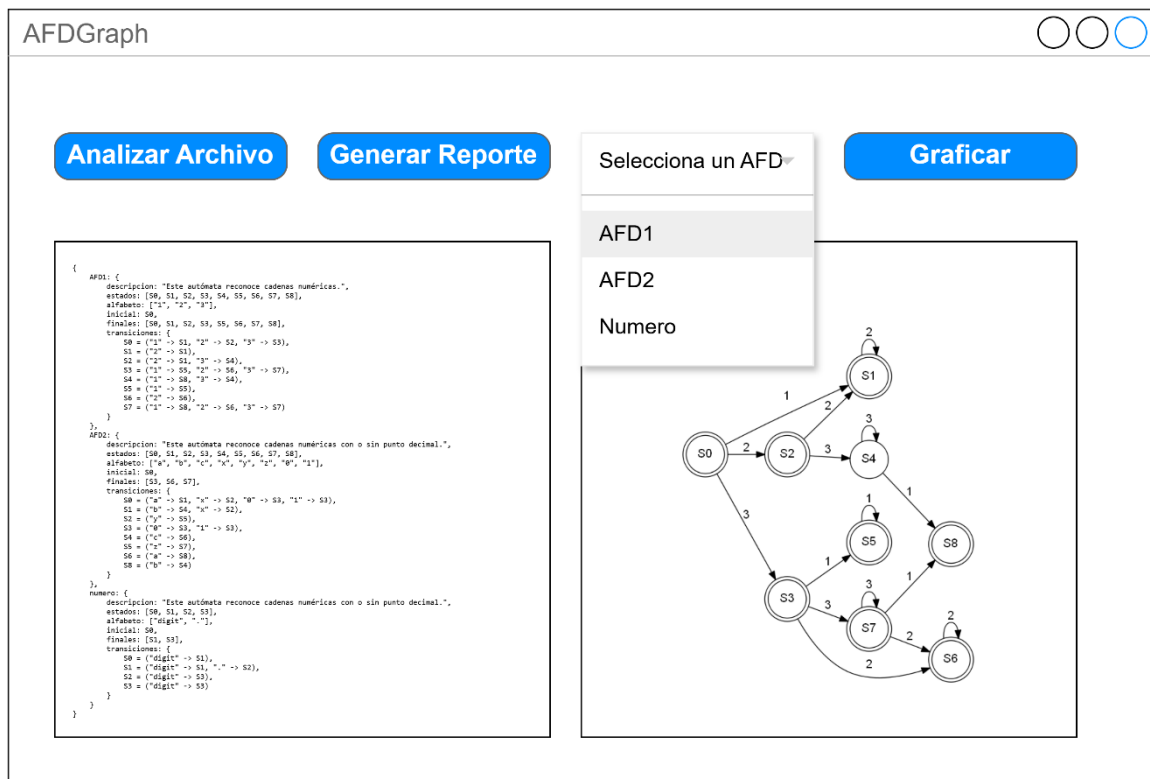
- Comprender el manejo de archivos en Java.
- Implementar un programa en Java que permita aplicar conocimientos sobre analizadores léxicos.
- Desarrollar habilidades y conocimientos en lógica de programación.
- Aplicar conocimientos sobre programación orientada a objetos en Java.
- Aplicar conocimientos sobre conceptos generales como alfabeto, símbolos y cadenas.

ENUNCIADO

En la universidad, los catedráticos del área de Ciencias de la Computación necesitan una herramienta para visualizar **Autómatas Finitos Deterministas (AFD)** a partir de una descripción textual. Actualmente, los estudiantes deben dibujar los autómatas manualmente basándose en tablas de transición, lo que puede llevar a errores y dificultar la comprensión de los conceptos.

Para solucionar esto, se desarrollará un **programa en Java** que lea archivos de entrada con la definición de un AFD y genere una imagen visual del autómata.

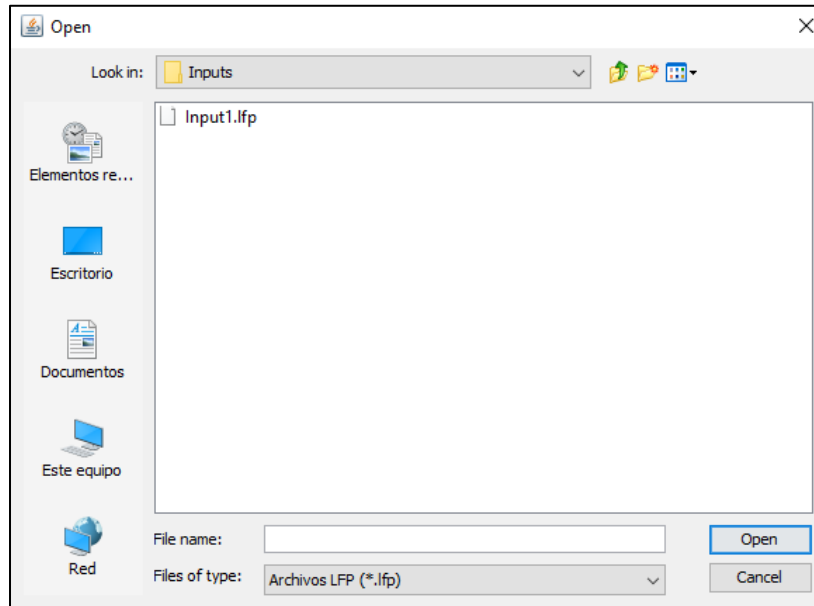
PROPUESTA DE INTERFAZ



FUNCIONES DEL SISTEMA

1. Analizar Archivo

- a. Muestra una ventana emergente que permite al usuario seleccionar un archivo “.lfp” que contiene información de los automatas



- b. Esta función permitirá guardar en memoria (en un HashMap) los datos de los autómatas y construir la estructura del autómata usando POO.
- c. Los autómatas tienen un nombre propio, por lo que si vienen dos o más autómatas con el mismo nombre en un mismo archivo se tomará en cuenta el último.

2. Graficar Autómata

- a. Es posible recorrer la estructura en la que se almacenaron los autómatas y seleccionar un autómata en específico para graficar.
- b. Los autómatas deben de mostrarse en un selectbox y luego dar click en el botón de “Graficar”.

3. Generar Reportes

Hay 3 reportes principales que se podrán generar.

a. Reporte de Tokens.

Token	Lexema	Línea	Columna
Llave Izquierda	{	1	2
Identificador	AFD1	2	9
Dos Puntos	:	2	10
Llave Izquierda	{	2	12
Reservada	descripcion	3	20
Dos Puntos	:	3	21
Cadena	"Este autómatá reconoce cadenas numéricas."	3	65
Coma	,	3	66
Reservada	estados	4	16

b. Reporte de Errores Léxicos.

Caracter	Línea	Columna
@	1	1
\$	1	2
~	2	6
+	2	8

SCRIPT DE AUTÓMATAS

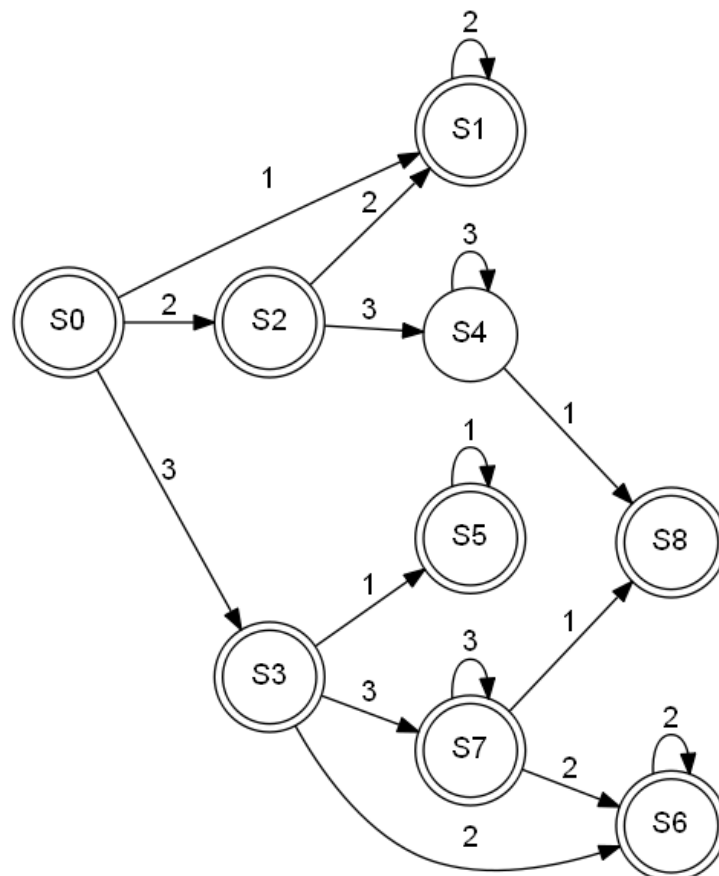
Ejemplo de Archivo de Autómatas

```
1 {
2   AFD1: {
3     descripcion: "Este autómata reconoce cadenas numéricas.",
4     estados: [S0, S1, S2, S3, S4, S5, S6, S7, S8],
5     alfabeto: ["1", "2", "3"],
6     inicial: S0,
7     finales: [S0, S1, S2, S3, S5, S6, S7, S8],
8     transiciones: {
9       S0 = ("1" -> S1, "2" -> S2, "3" -> S3),
10      S1 = ("2" -> S1),
11      S2 = ("2" -> S1, "3" -> S4),
12      S3 = ("1" -> S5, "2" -> S6, "3" -> S7),
13      S4 = ("1" -> S8, "3" -> S4),
14      S5 = ("1" -> S5),
15      S6 = ("2" -> S6),
16      S7 = ("1" -> S8, "2" -> S6, "3" -> S7)
17    }
18  },
19  AFD2: {
20    descripcion: "Este autómata reconoce cadenas numéricas con o sin punto decimal.",
21    estados: [S0, S1, S2, S3, S4, S5, S6, S7, S8],
22    alfabeto: ["a", "b", "c", "x", "y", "z", "0", "1"],
23    inicial: S0,
24    finales: [S3, S6, S7],
25    transiciones: {
26      S0 = ("a" -> S1, "x" -> S2, "0" -> S3, "1" -> S3),
27      S1 = ("b" -> S4, "x" -> S2),
28      S2 = ("y" -> S5),
29      S3 = ("0" -> S3, "1" -> S3),
30      S4 = ("c" -> S6),
31      S5 = ("z" -> S7),
32      S6 = ("a" -> S8),
33      S8 = ("b" -> S4)
34    }
35  },
36  numero: {
37    descripcion: "Este autómata reconoce cadenas numéricas con o sin punto decimal.",
38    estados: [S0, S1, S2, S3],
39    alfabeto: ["digit", "."],
40    inicial: S0,
41    finales: [S1, S3],
42    transiciones: {
43      S0 = ("digit" -> S1),
44      S1 = ("digit" -> S1, "." -> S2),
45      S2 = ("digit" -> S3),
46      S3 = ("digit" -> S3)
47    }
48  }
49 }
```

GRAFO DEL AUTÓMATA

```
{
  AFD1: {
    descripcion: "Este autómata reconoce cadenas numéricas.",
    estados: [S0, S1, S2, S3, S4, S5, S6, S7, S8],
    alfabeto: ["1", "2", "3"],
    inicial: S0,
    finales: [S0, S1, S2, S3, S5, S6, S7, S8],
    transiciones: {
      S0 = ("1" -> S1, "2" -> S2, "3" -> S3),
      S1 = ("2" -> S1),
      S2 = ("2" -> S1, "3" -> S4),
      S3 = ("1" -> S5, "2" -> S6, "3" -> S7),
      S4 = ("1" -> S8, "3" -> S4),
      S5 = ("1" -> S5),
      S6 = ("2" -> S6),
      S7 = ("1" -> S8, "2" -> S6, "3" -> S7)
    }
  }
}
```

Grafo para el autómata del archivo de ejemplo. El grafo se generará a partir de las transiciones especificadas en el archivo de entrada.



ENTREGABLES

- Manual de Usuario
- Manual Técnico
- Código fuente

CONSIDERACIONES

- Se debe desarrollar de forma individual.
- El proyecto será anulado si el estudiante utiliza herramientas externas para el analizador léxico o el escáner.
- No se permite el uso de herramientas como ANTLR, JFlex, Lex, Yacc o cualquier generador automático de analizadores léxicos.
- El análisis léxico debe realizarse mediante un autómata finito determinista (AFD), con una implementación que procese carácter por carácter y maneje los estados del analizador manualmente.
- En el repositorio privado en Github con el nombre [LFP]<carnet> creado anteriormente debe crear una carpeta llamada Proyecto1 que contenga los entregables requeridos.
- Para la interfaz gráfica, es obligatorio utilizar Java Swing.
- No se permitirá el uso de JavaFX u otros frameworks gráficos externos.
- Agregar al auxiliar a su repositorio de GitHub: Jona1056.
- La entrega se realizará en la plataforma UEDI. Únicamente deberán subir el enlace de su repositorio de GitHub en la plataforma.
- No se aceptan entregas vía correo electrónico u otro medio.
- La calificación será presencial.
- La calificación del proyecto será presencial y durará como máximo 15 minutos, en un horario que posteriormente será establecido.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación (2 días antes) para ceder su lugar a otro estudiante, en caso contrario el estudiante solo obtendrá el 80% de su nota obtenida.
- No se dará prórroga para la entrega de la práctica.
- **COPIA PARCIAL O TOTAL DEL PROYECTO TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ A LA ESCUELA DE SISTEMAS PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES.**
- En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará la práctica; por lo cual, se tendrá una nota de cero puntos.

Fecha de entrega: 26 de marzo de 2025 antes de las 23:59, no se recibirá después de la fecha y hora establecida.