## Kevin Andrés Ortiz Ramírez

1242018

## Proyecto Lenguajes Formales & Autómatas

**Introducción**

Se realizó un programa en Visual Studio en el lenguaje de programación c# de tipo Forms para un analizador gramatical a partir de una expresión regular y un árbol de expresión.

El programa recibe un archivo de texto y este valida si el archivo cumple con la gramática o no. Si este no cumple el programa indicara en que línea y columna sucedió el error.

**Autómata**

Es una máquina matemática M formada por 5 elementos M = (Σ, Q, s, F, δ) donde, Σ es un alfabeto de entrada, Q es un conjunto finito de estados, s es el estado inicial, F es un conjunto de estados finales o de aceptación y δ (delta) es una relación de transición.

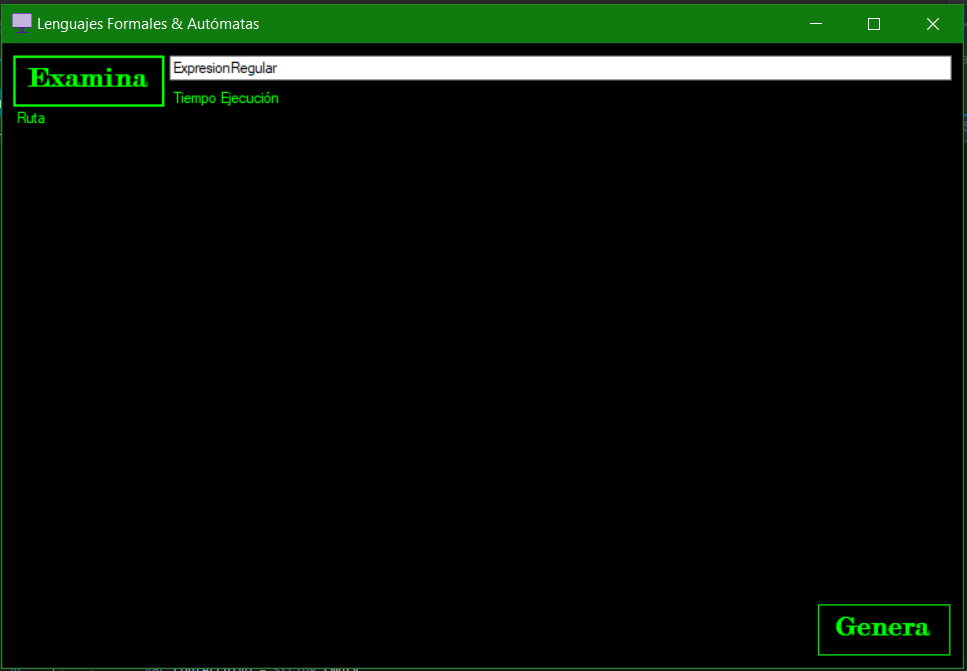
**Autómata Finito Determinista (AFD)**

Es un autómata finito en donde δ (delta) es una función de transición, es decir, que para cada par (estado actual y símbolo de entrada) le corresponde un único estado siguiente.

**Función**

Se realiza un programa de lectura de archivo donde dicho archivo debe contener SET, TOKEN, ACTION y ERROR. Se genera una Expresión regular que genera un Árbol de Expresiones que a su vez genera una los First, Last y Follow de los nodos correspondientes.

**Manual de usuario**

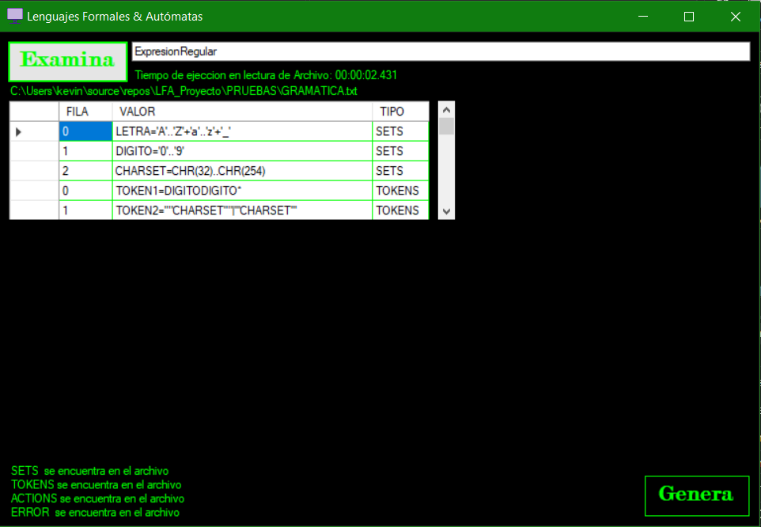
Al ingresar al programa y correrlo aparecerá la siguiente ventana:

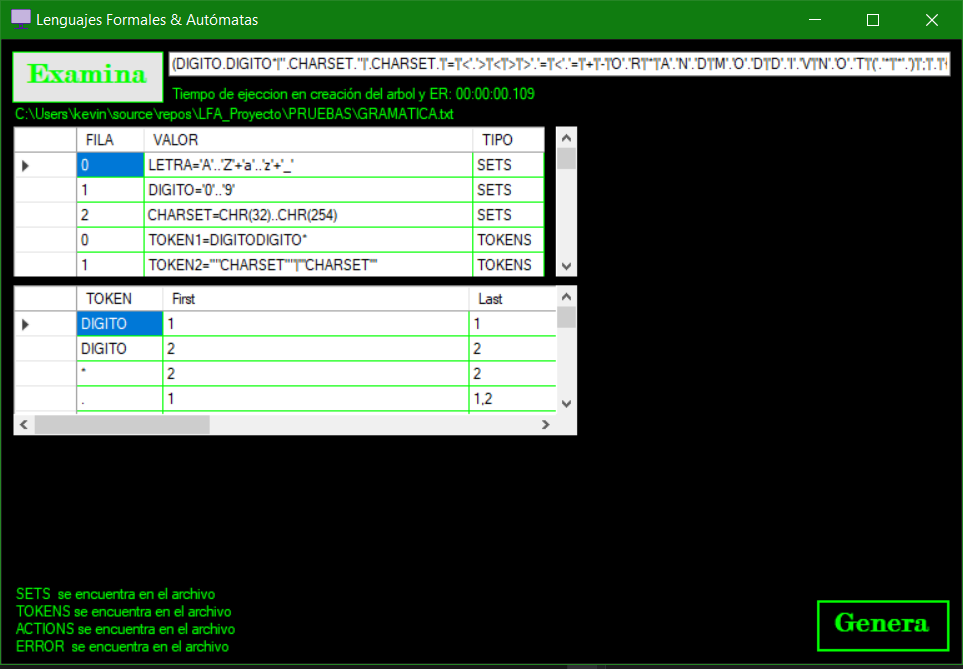
Se debe presionar el botón “Examina” y saltará el explorador de archivos en el directorio “c://PRUEBAS”. El programa tiene restricción implícita que permite únicamente la subida de archivos de texto(.txt).

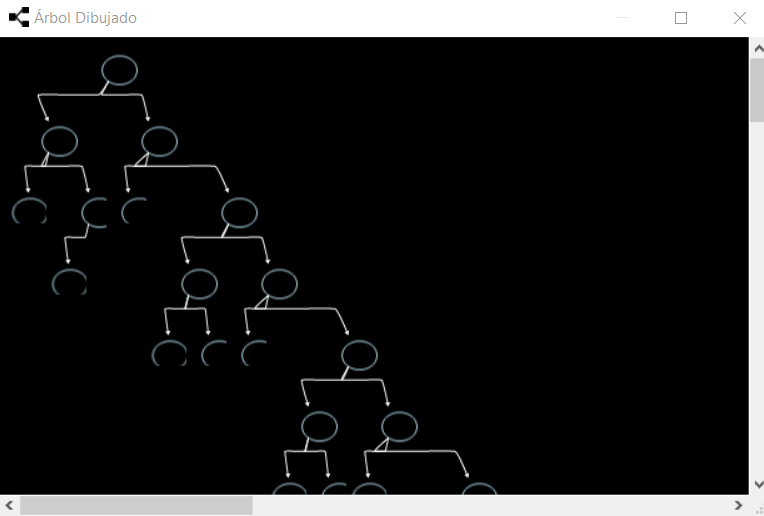
Captura de pantalla de un celular

Descripción generada automáticamenteAl ingresar el archivo deseado para evaluar la gramática, se confirmará que el archivo contiene el formato correcto

Y se mostrarán en pantalla los SETs, TOKENs, ACTIONs y ERRORs agregados en el archivo



Al presionar “Genera” desplegará otra ventana con el Árbol de Expresiones creado y se despliegan los First, Last, Follow, Null que contenga en el árbol. 



## Pseudocódigo

Teniendo un archivo de entrada en formato .txt

1. Comprobar que exista los requisitos mínimos, siendo estos
   1. SETS(No obligatorio a no ser que TOKEN lo requiera)
      1. Se comprueba el formato de escritura como [Nombre\_SET] = [TIPO\_SET]
         1. Se agrega a lista de SETs
   2. TOKEN(Obligatorio)
      1. Se comprueba el formato de escritura como [TOKEN\_#] = [‘TIPO\_TOKEN’]
         1. Se agrega a lista de Símbolos Terminales
      2. En caso [‘TIPO\_TOKEN’] no contenga comillas, se comprueba que exista en la lista de SETs de SETs
         1. Se agrega a lista de símbolos Terminales
      3. Si no
         1. Error de sintaxis en TOKEN
   3. ACTIONS/RESERVADAS(Obligatorio)
      1. Se comprueba el formato de escritura como [#] = [‘TIPO\_ACTION’]
         1. Se agrega a lista Actions
   4. ERROR(Opcional)
      1. Se comprueba el formato de escritura como [ERROR] = [#]
         1. Se agrega a lista Errors
2. Con la lista de símbolos terminales se genera un árbol de expresiones con pseudocódigo echo por Moises Alonso. **ER to ET**
3. Se procede a generar las tablas correspondientes
   1. Obtener Simbolo
      1. Si es |
         1. Es nulo si su hijo izquierdo o su hijo derecho es nulo
      2. Si es .
         1. Es nulo si el hijo izquierdo y el hijo derecho es nulo
      3. Si es \*, + o ?
         1. Son nulos
   2. Por cada nodo donde sus hijos sean nulos
      1. First = conteo
      2. Last = conteo
      3. Conteo + 1
   3. Si es |
      1. Su First es igual a la unión de los first del hijo derecho con los first del hijo izquierdo
   4. Si es .
      1. Si el hijo izquierdo es nullo
         1. First es igual a la unión de los first del hijo izquierdo con los del hijo derecho
      2. Si no
         1. First es igual al first del hijo derecho
   5. Si es \*, + ó ?
      1. First es igual al first del hijo izquierdo
   6. Generar los Follow de cada nodo
      1. Si el simbolo es **|**, **+**  ó **?** no contienen follow
      2. En caso sea .
         1. Para todo hijo izquierdo, le corresponde el First del hijo derecho
      3. En caso sea \*
         1. Para todo hijo izquierdo, le corresponde el First del hijo izquierdo