

Universidad Rafael Landívar

Facultad de Ingeniería

Lenguajes formales y autómatas, sección 01

Catedrático: Ing. Moises Alonso

Proyecto LFA

“Fase III”

Derly Alejandro Rodas Pérez

Carné: 1177318

Guatemala, 06 de Mayo del 2020

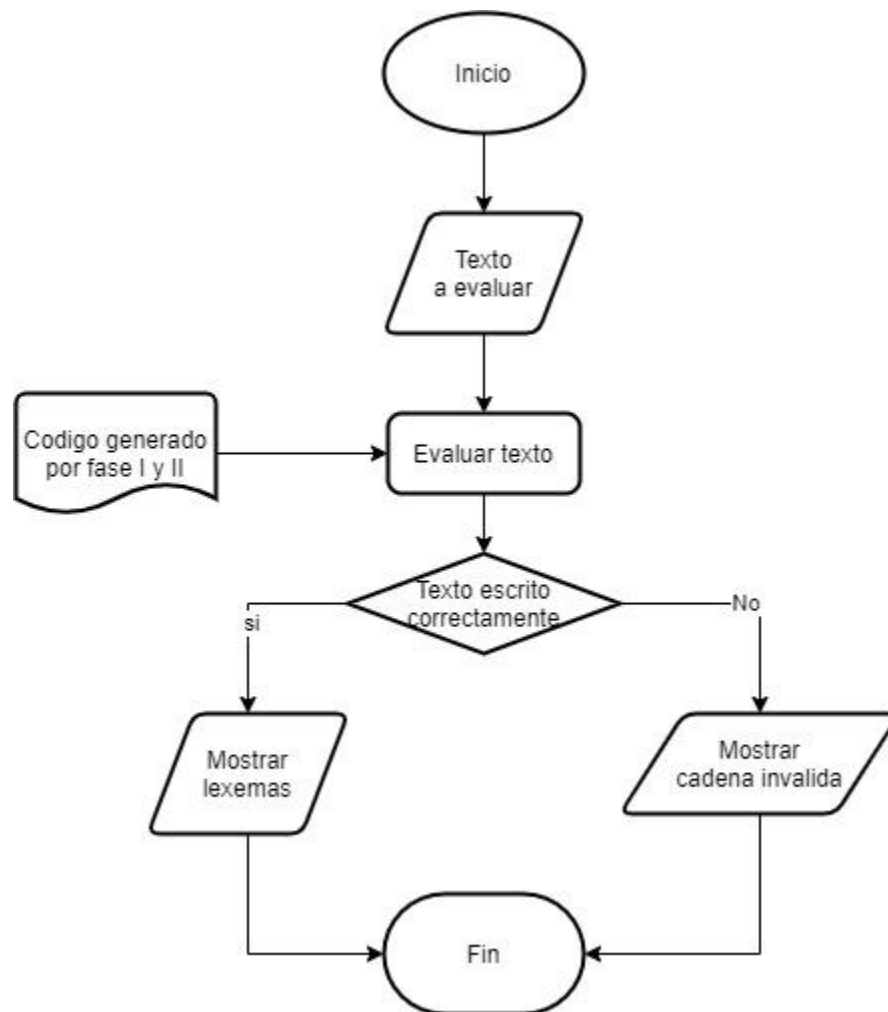
Análisis:

- **Entradas:**
 - Documento de texto o cadena de caracteres a evaluar en el escáner. También documento de texto con las normativas para generar el escáner.
- **Procesos:**
 - Se evalúa el documento de texto que contiene la gramática y se construye el autómata
 - Se procede a extraer las reglas importantes que condicionarán el código del escáner
 - Reconocer los lexemas, durante la evaluación del texto ingresado.
- **Salidas:**
 - Proyecto que contiene el scanner
 - Mensaje que muestra si la cadena ingresa es correcta y sus respectivos lexemas.

Métodos para utilizar:

- **Fase I y II:**
 - **Transportar Actions:**
 - **Parámetros:**
 - Actions recibidos por parte de la lectura del archivo gramática
 - **Salida:**
 - Lista de palabras reservadas
 - **Transportar Tokens**
 - **Parámetros:**
 - Tokens recibidos por parte de la lectura del archivo gramática
 - **Salida:**
 - Lista de tokens con su respectivo numero
 - **Asignar caminos a estados:**
 - **Parámetros:**
 - Tabla de transiciones creada en la fase I y II
 - **Salida:**
 - Estructura de IF y Switch para fase III
 - **Crear métodos utilizados en fase III:**
 - **Parámetros:**
 - Terminales y lista de tokens
 - **Salida:**
 - Código utilizado para fase III.
- **Fase III:**
 - **Lectura:**
 - **Parámetros**
 - Cadena de caracteres o documento de texto a evaluar.
 - **Salida**
 - Lexemas correspondientes o si la cadena es incorrecta.
 - **Evaluar**
 - **Parámetros**
 - Int letra
 - **Salida**
 - Si es carácter es valido para el alfabeto manejado.
 - **Verificar Tokens**
 - **Parámetros**
 - String t
 - **Salida**
 - A que token pertenece la cadena ingresada (lexema)

Diagrama de flujo:



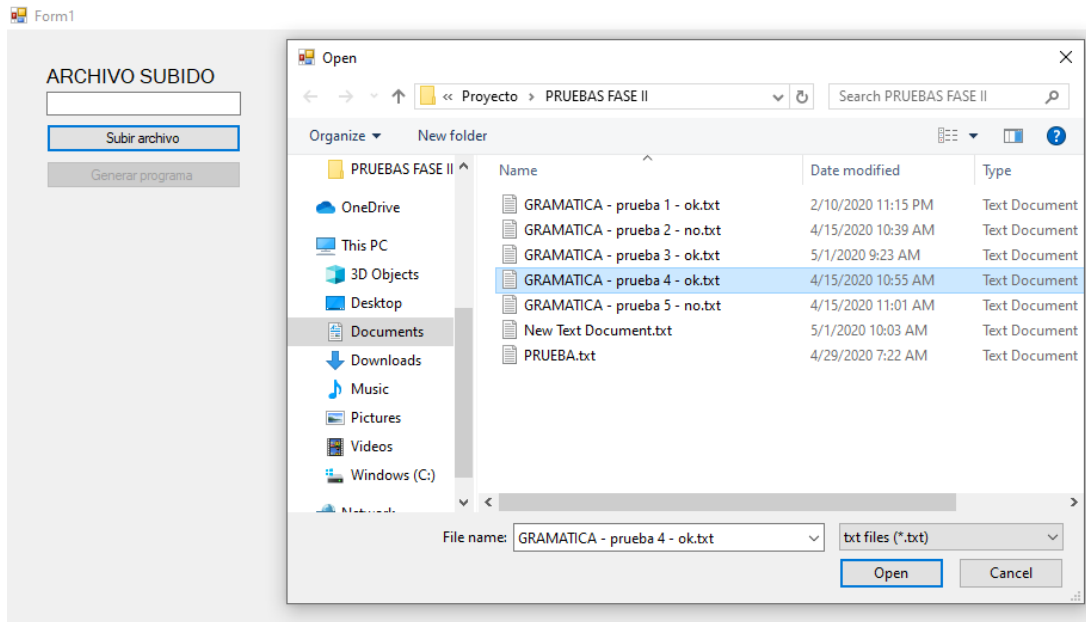
Pruebas:

- ER
 - $\text{DIGITO} \mid \text{DIGITO} * \mid \text{LETRA} * \mid \text{LETRA} \mid '-' \text{DIGITO}$
- Reservadas:
 - 18 = 'PROGRAM'
 - 19 = 'INCLUDE'
 - 20 = 'CONST'
 - 21 = 'TYPE'
 - 22 = 'VAR'
 - 23 = 'RECORD'
 - 24 = 'ARRAY'
 - 25 = 'OF'
 - 26 = 'PROCEDURE'
 - 27 = 'FUNCTION'
 - 28 = 'IF'
 - 29 = 'THEN'
 - 30 = 'ELSE'
 - 31 = 'FOR'
 - 32 = 'TO'
 - 33 = 'WHILE'
 - 34 = 'DO'
 - 35 = 'EXIT'
 - 36 = 'END'
 - 37 = 'CASE'
 - 38 = 'BREAK'
 - 39 = 'DOWNTON'
- Token
 - 1 = $\text{DIGITO} \mid \text{DIGITO} * \mid \text{LETRA} *$
 - 3 = $\text{LETRA} \mid '-' \text{DIGITO}$

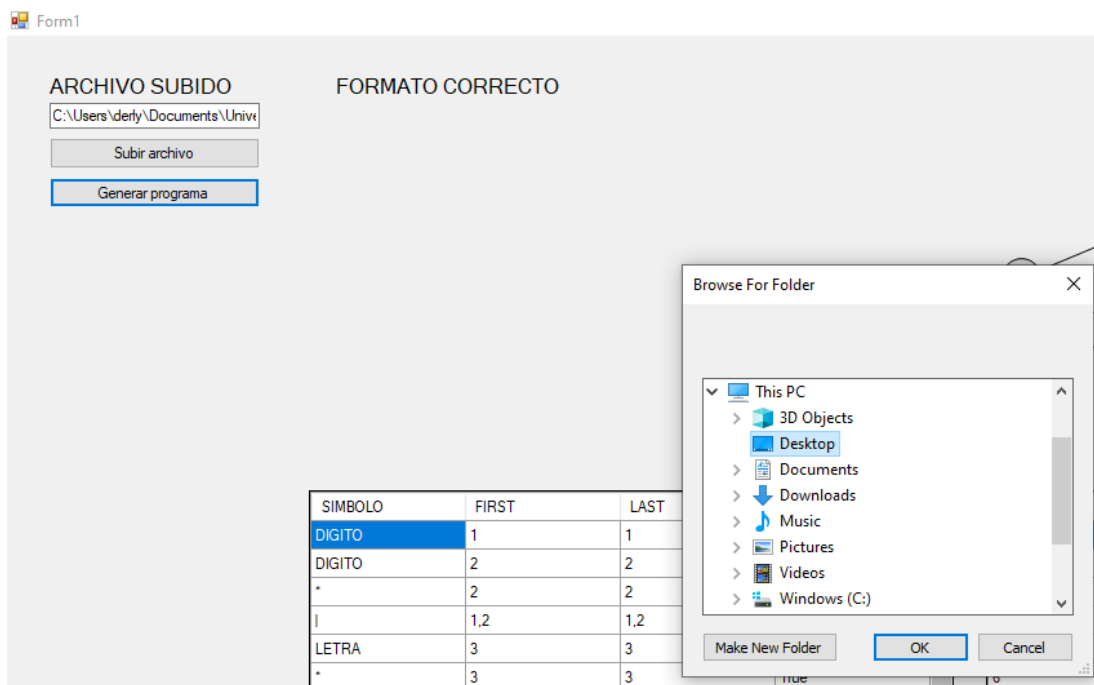
```
C:\Users\derly\Desktop\Generico\Generico\l
Menu
1.Ingresar Archivo de.txt
2.Ingresar cadena
3.Salir
2
Ingrese cadena a evaluar
-5-5-5-5-5-5-5-5 Program Var
-5 = 3
-5 = 3
-5 = 3
-5 = 3
-5 = 3
-5 = 3
-5 = 3
-5 = 3
Program = 18
Var = 22
Presione enter para continuar
```

Manual de usuario:

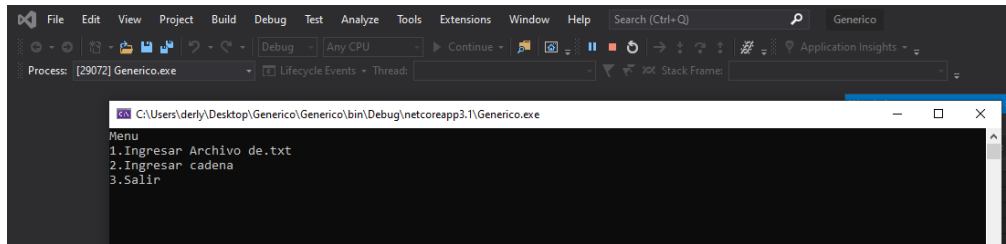
1. Subir archivo de gramática a utilizar



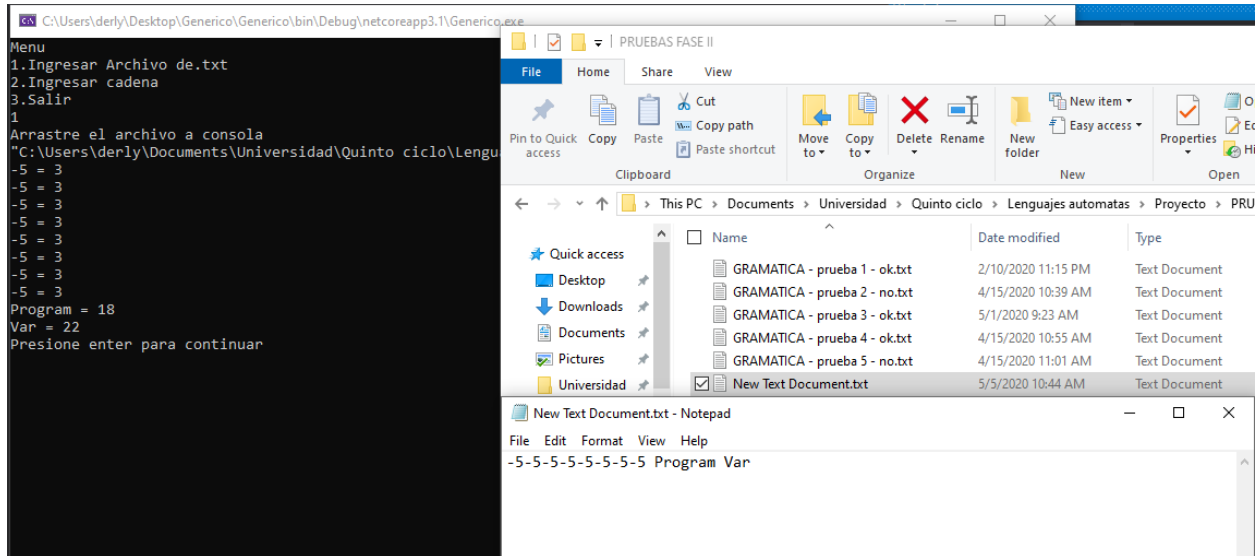
2. Presionar botón generar programa y escoger ruta en donde guardar proyecto de escáner



3. Dirigirse a la carpeta creada y correr la solución



4. En el menú se puede seleccionar dos opciones
 - a. Por archivo de texto: se debe de arrastrar él .txt a la ventana consola



- b. Por cadena: Se ingresa la cadena a evaluar directamente en la ventana consola

