

Universidad Rafael Landívar
Facultad de Ingeniería
Lenguajes formales y autómatas
Sección 01
Ing. Moises Alonso

PROYECTO PRÁCTICO FASE III GENERADOR DE SCANNER

Brenner Hernandez 1023718

Guatemala de la asunción, 06 de mayo del 2020

Análisis

Entradas

Se aceptará un archivo de texto que contiene una gramática que será analizada.

Procesos

Después de haber analizado la gramática según la Fase I y Fase II, se procederá a analizar la tabla de transiciones para así poder definir qué casos y condiciones para las mismas se deberán crear para el scanner a generar.

Salidas

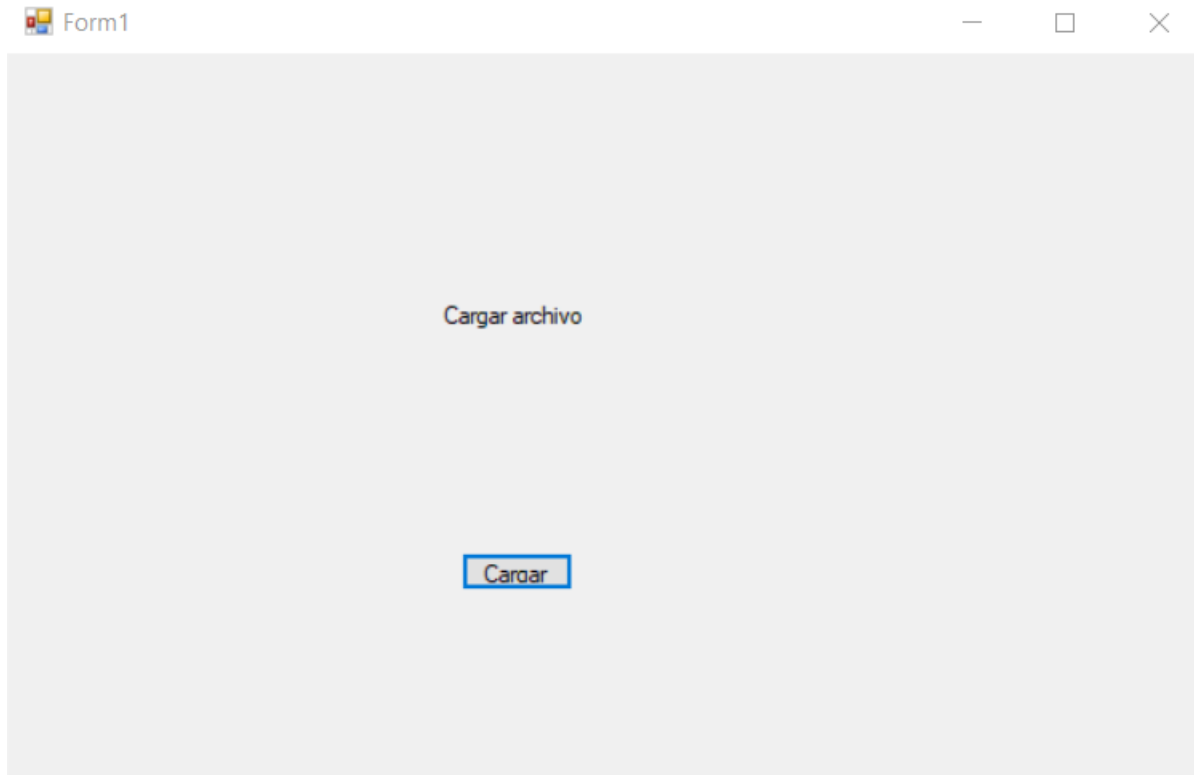
Un programa en lenguaje C# que contendrá un scanner que analizará cadenas según la gramática ingresada, y este mismo tendrá salidas del estado de esa cadena, si fue aceptada o no, y además los tokens o palabras reservadas que se contienen en la misma.

Restricciones

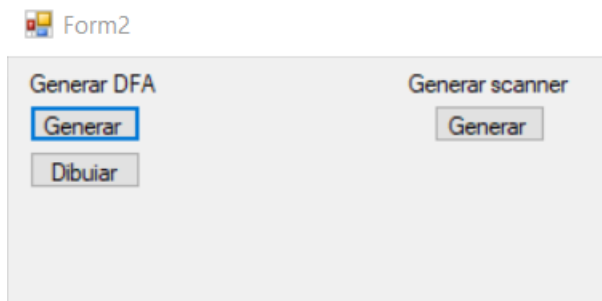
El archivo de entrada tendrá que tener un formato .txt y no estar vacío.

Manual de usuario

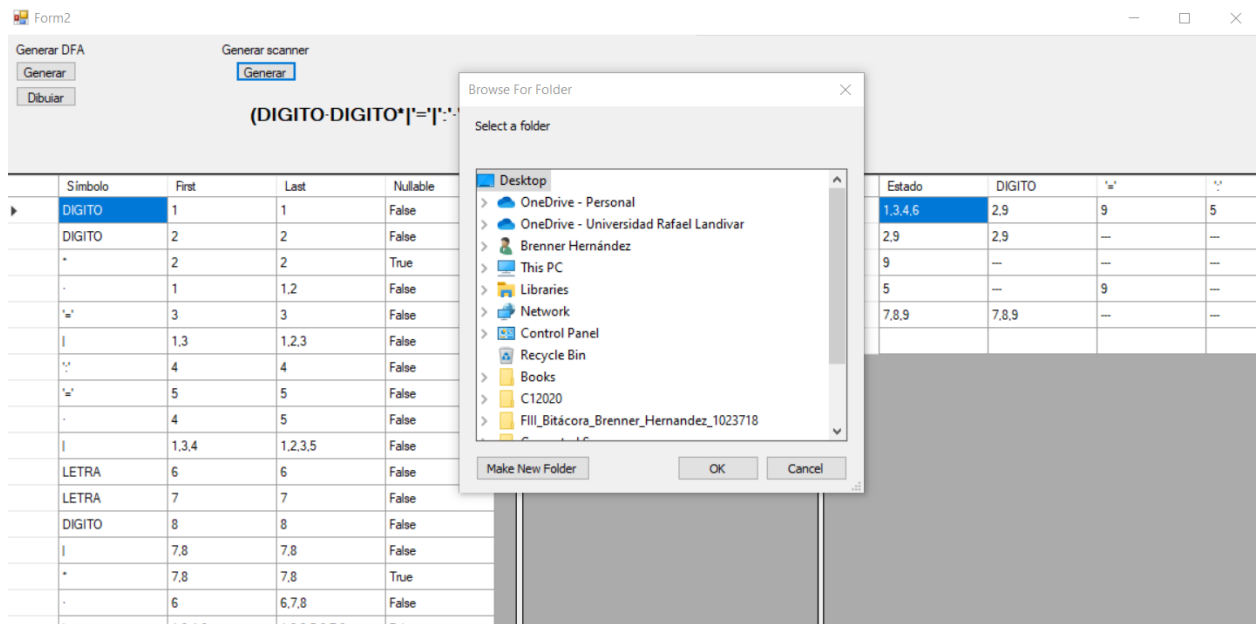
1. Primeramente, se tendrá que cargar un archivo de texto que contenga la gramática a analizar.



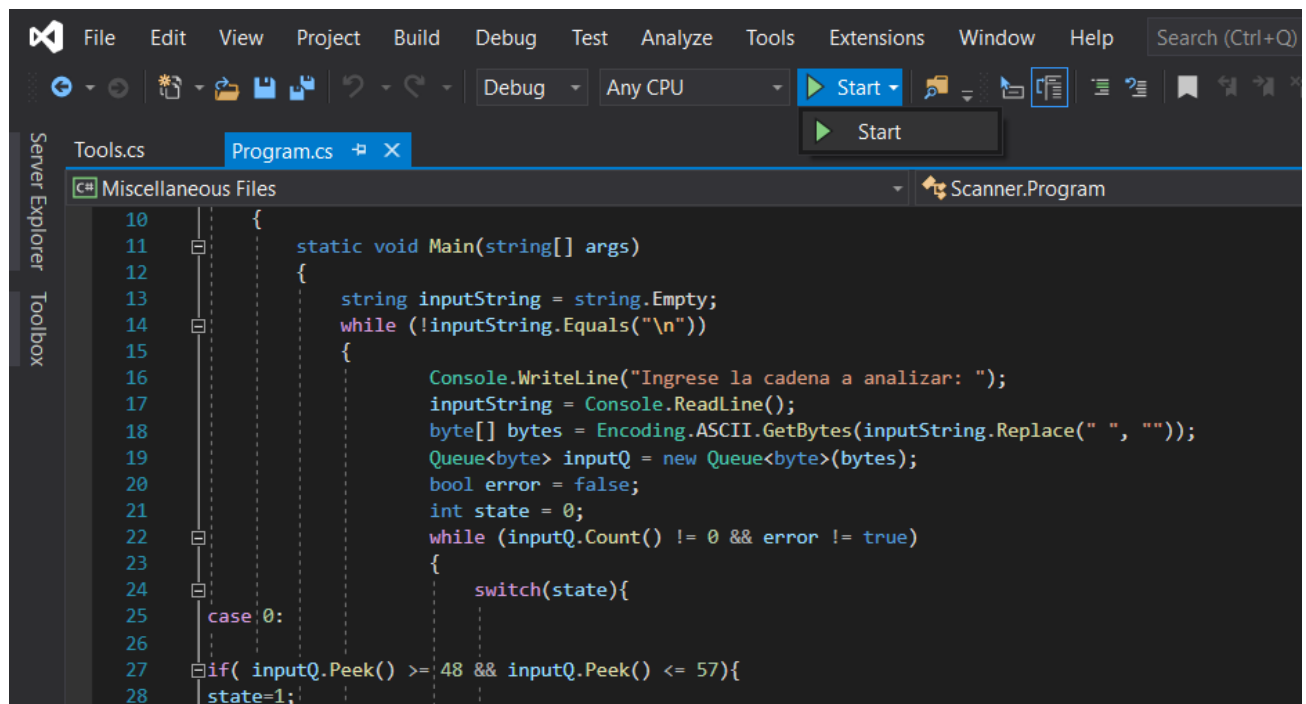
2. Generar las tablas realizadas en la Fase II de este proyecto.



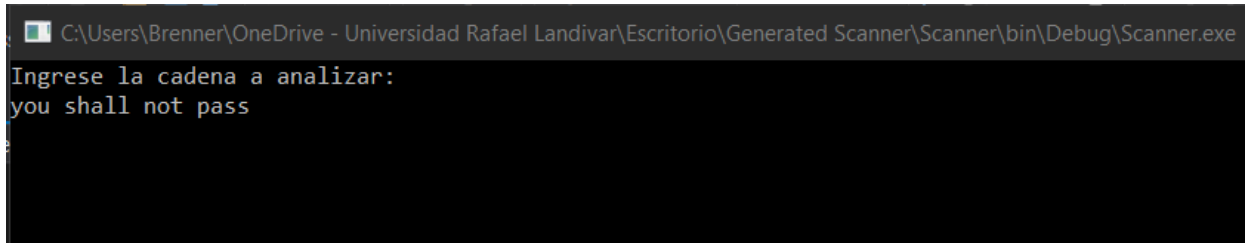
3. Generar el scanner y seleccionar el directorio deseado para almacenar este nuevo programa.



4. Una vez se haya generado el scanner, podrá iniciar el mismo y ejecutarlo.

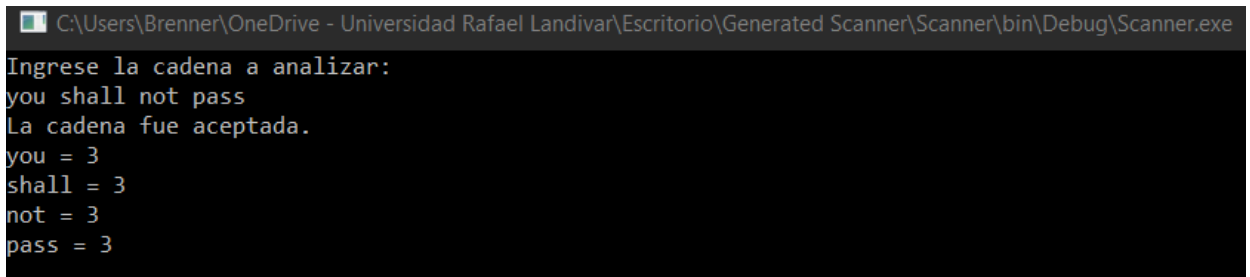


5. Cuando se haya ejecutado el programa del scanner, se deberá ingresar la cadena que se desea analizar.



```
C:\Users\Brenner\OneDrive - Universidad Rafael Landivar\Escritorio\Generated Scanner\Scanner\bin\Debug\Scanner.exe
Ingrese la cadena a analizar:
you shall not pass
```

6. Cuando se haya ingresado la cadena, se deberá presionar enter. Y se obtendrá un resultado así. Con los tokens y las partes de la cadena como se especifica en la gramática.



```
C:\Users\Brenner\OneDrive - Universidad Rafael Landivar\Escritorio\Generated Scanner\Scanner\bin\Debug\Scanner.exe
Ingrese la cadena a analizar:
you shall not pass
La cadena fue aceptada.
you = 3
shall = 3
not = 3
pass = 3
```

Diagramas de clases

DFA
<ul style="list-style-type: none">• TSigns(): este obtiene los Sets agregados en la gramática.• GetSetsRanges(): este método devuelve los sets agregados en el anterior método y los tokens con sus respectivos rangos de aceptación con el código ascii.• GetTokenWords(): se devuelve el número de token o palabra reservada, y el contenido de las mismas.• GetFinalStates(): los estados y su valor en el caso de que sean estados finales o no.

Tools
<ul style="list-style-type: none">• TokenizeText(): se tokeniza la cadena de entrada ingresada por el usuario en el scanner generado.• TokenListToPrint(): se obtiene la lista del valor de los tokens junto con las partes de la cadena al que corresponden según lo que ingresó el usuario.

FileClass
<ul style="list-style-type: none">• CopyFolder(): se copia y pega el folder de la solución del scanner generando en la ubicación seleccionada por el usuario.• DeleteDirectory(): se elimina el folder de la copia de la solución del scanner en caso el usuario seleccione una dirección en la que ya exista un scanner.• RemoveSingleQuotes(): se creó este método para específicamente remover todas las comillas simples en el proyecto.