Universidad Rafael Landívar

Facultad de Ingeniería

Lenguajes Formales Y Autómatas, sección 01

Catedrático: Ing. Moisés Alonso

**Proyecto Práctico Fase 2**

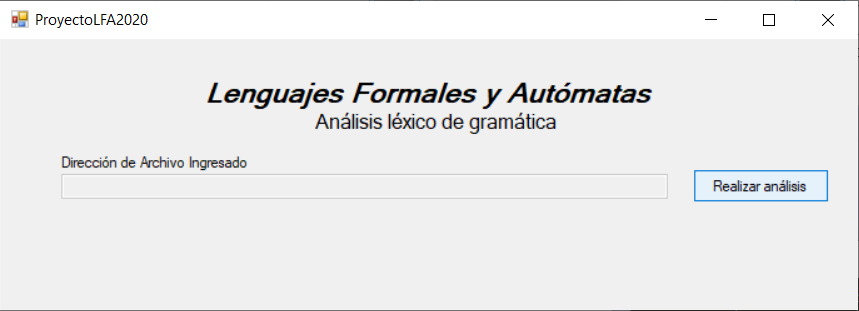
**“Análisis Sintáctico de Gramática”**

José Eduardo Meléndez De la Rosa 1059918

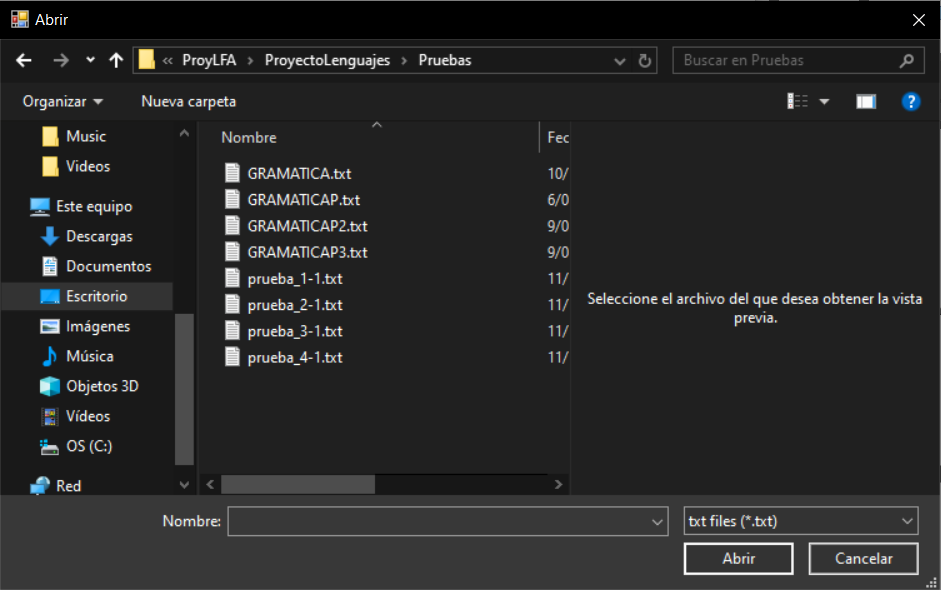
Guatemala, 9 de abril del 2020

# Manual de Usuario

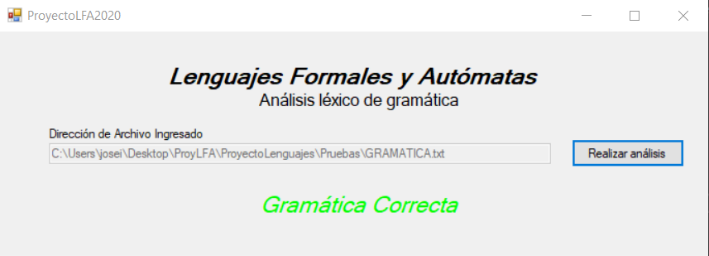
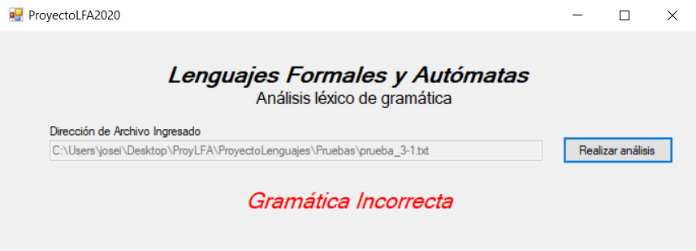
La primera ventana que se abre muestra un botón con el texto “Realizar Análisis”



Al hacer clic en el botón se abre una ventana de dialogo de apertura de archivo en el que solo se muestran archivos con extensión “.txt”, si se da el caso en que la carga del archivo es incorrecta, dicha ventana se abre de nuevo.



Se realizan las verificaciones correspondientes al análisis léxico de la gramática y se muestra una de la ventana inicial cambia a uno de los siguientes estados según corresponda.



Si la gramática ingresada es correcta se abre la ventana que se muestra a continuación



Se debe hacer clic sobre el botón con el texto “Proceso Fase 2”



Una vez realizados los cálculos las tablas se llenan con la información requerida por el análisis sintáctico de la gramática.

# Pseudocódigo

## Definición de los valores First, Last de los nodos hoja

Entradas: Árbol de expresión, Cola de NodosHoja, Contador de Nodos

Algoritmo:

1. Si el nodo tiene hijo izquierdo
   1. Visitar hijo Izquierdo
   2. Ir a paso1.
2. Si el nodo tiene hijo derecho
   1. Visitar hijo derecho
   2. Ir a paso 1.
3. Si el nodo no tiene hizo izquierdo y no tiene hijo derecho
   1. Igualar los valores “first” y “Last” del nodo al valor del contador
   2. Agregar el Nodo a la Cola de Nodos
   3. Incrementar el valor del contador en 1.

## Calcular valores First, Last Y Nullable del resto de Nodos del árbol

Entradas: Árbol de expresión, Cola de NodosArbol

Algoritmo:

1. Si el nodo tiene hijo izquierdo
   1. Visitar hijo Izquierdo
   2. Ir a paso1.
2. Si el nodo tiene hijo derecho
   1. Visitar hijo derecho
   2. Ir a paso 1.
3. Si el carácter del nodo es “+”
   1. Igualar los valores first y last a los valores first y last del hijo izquierdo del nodo
   2. Si el hijo es anulable, hacer anulable el nodo
4. Si el carácter del nodo es “\*”
   1. Igualar los valores first y last a los valores first y last del hijo izquierdo del nodo
   2. hacer anulable el nodo.
5. Si el carácter del nodo es “?”
   1. Igualar los valores first y last a los valores first y last del hijo izquierdo del nodo
   2. hacer anulable el nodo.
6. Si el carácter del nodo es “.”
   1. Si el hijo izquierdo es anulable
      1. Igualar First del nodo a la unión del first de los hijos izquierdo y derecho
      2. Si no Igualar first al first del hijo izquierdo.
   2. Si el hijo derecho es anulable
      1. Igualar last del nodo a la unión del last de los hijos izquierdo y derecho
      2. Si no, Igualar last al last del hijo derecho.
   3. Si el hijo izquierdo y derecho son anulables
      1. Hacer anulable el nodo
7. Si el carácter del nodo es “|”
   1. Igualar First del nodo a la unión del first de los hijos izquierdo y derecho
   2. Igualar Last del nodo a la unión del last de los hijos izquierdo y derecho
   3. Si el hijo derecho o el hijo izquierdo es anulable
      1. Hacer anulable el nodo
8. Inicializar nodo auxiliar e igualar los valores first, last y nullable a los del nodo actual.
9. Encolar el nodo en la cola de NodosArbol

## Calcular Follow de los nodos hoja

Entradas: Árbol de expresión, Cola de NodoHoja

Algoritmo:

1. Si el nodo tiene hijo izquierdo
   1. Visitar hijo Izquierdo
   2. Ir a paso1.
2. Si el nodo tiene hijo derecho
   1. Visitar hijo derecho
   2. Ir a paso 1.
3. Si el carácter del nodo es “+” o “\*”
   1. Para cada valor last del hijo izquierdo
   2. Ir al NodoHoja cuyos valores first y last sean iguales al valor que se está buscando
   3. Agregar todos los valores first del hijo izquierdo al follow del nodo actual.
4. Si el carácter del nodo es “.”
   1. Para cada valor last del hijo izquierdo
   2. Ir al NodoHoja cuyos valores first y last sean iguales al valor que se está buscando
   3. Agregar todos los valores first del hijo derecho al follow del nodo actual.

## Calcular las transiciones

Entrada: Lista de enteros con el estado inicial, cola de nodos hoja, lista de símbolos usados, Cola estados nuevos, Cola de estados visitados, Diccionario de Estados Analizados

\*\*La estructura del diccionario de estados utilizado es

|  |  |  |
| --- | --- | --- |
| Diccionario< List<List<int>>,Diccionario<string, List<int>>> | | |
| List<List<int>> | Diccionario<string, List<int>> | |
| string | List<int> |

Algoritmo

1. Encolar estado inicial en cola estados nuevos
2. Mientras la cantidad de estados nuevos sea mayor a cero
   1. Inicializar nuevo diccionario Diccionario<string, List<int>> y agregar todos los símbolos utilizados junto con una lista de enteros vacía y nombrarlo transiciones
   2. Desencolar el primer estado de la cola de estados nuevos y agregar junto con el diccionario de transiciones al diccionario de estados.
   3. Para cada valor en la lista de enteros del estado inicial visitar, obtener valor del nodo hoja.
   4. Ir al nodo hoja cuyo first y last sea igual al valor que se está comparando y obtener su follosw y carácter.
   5. Acceder al diccionario de estados en el estado actual e ir al diccionario de transiciones cuya llave sea igual al carácter del nodo hoja, igualar la lista de enteros al follow del nodo hoja.
   6. Si el follow del nodo hoja no existe en la cola de estados visitados y tampoco en la cola de estados nuevos, encolar el follow en la lista de estados nuevos.