

# Manual Técnico

## Clase Form1:

### **Compilar(): void:**

Se usa este método para instanciar la clase de Autómata que realiza todo el proceso para que en el cuadro de texto donde se va a escribir el código, cada palabra sea identificada por su léxico.

## Clase OpcionesArchivo:

### **abrir(RichTextBox): void**

Este método es para que se abra una ventana que nos muestre los archivos que tenemos en nuestra computadora y que podemos abrir solo los archivos con extensiones .gt y podemos modificarlo en el RichTextBox que aparece en el programa.

### **guardar(RichTextBox): void**

Este método sirve para que podamos guardar todo lo que hicimos en el RichTextBox guardándose con una extensión .gt para que el

### **exportarLog(LinkedList) : void**

Es un método que ayuda a guardar los errores que aparecen en el ListBox que debajo del RichTextbox que hace hace el mismo proceso que el metodo de guardar solo que ahora lo guarda con una extensión .gtE y que lo guarda tomando los datos de una LinkedsList y no de un RichTextBox.

# Clase Automata:

## **palabrasReservadas() : void**

Con este método es cuando el estado del autómata esta en H y buscamos si la palabra que esta analizando tiene algún color específico, si escribió una palabra reserva como el mismo método se llama.

## **cambiarcolor(int ,RichTextBox): void**

Lo que este método hace es que selecciona la cadena o palabra que fue aceptada que ya fue evaluada en el método de cambiarEstado() y cambia el color dentro del RichTextBox.

## **cambiarEstado(char ,RichTextBox, int): void**

Este es el método que verifica el carácter que estamos evaluando y revisa en que estado corresponde, es decir, usamos este método para evaluar el carácter con todas las funciones de transición para cambiar el estado a otro.

## **verificar(): void**

Verifica si el estado donde estamos es aceptable o no para ir a cambiar el color de la cadena o la del carácter.

## **BuscaColor() : void**

Si estamos en un estado aceptable por medio del método verificar entonces usamos este método y por medio del estado buscamos su color correspondiente para que así sea cambiado por ese color a la cadena o carácter de se evaluó.

## **errores(char, char) : void**

Sirve para conocer si el carácter evaluado esta dentro del abecedario o si este esta fuera del léxico que se esta manejando.

## **resultado(char): String**

Este método es para ver como es que mueve un carácter por medio de la tabla de transición creada, es decir, muestra en que estado estábamos al evaluarlo y que carácter se estaba evaluando y hacia que estado fue dirigido este carácter.

## **funcionesTransicion(): void**

Aquí es donde se colocaban todas las funciones de transición que tiene este autómata.

## **excepciones(): void**

En este método se controlaba algunos estados que no podían ser reiniciados correctamente por la estructura que tenía la tabla de transición ya que eran estados que tenían otro comportamiento que los demás podían manejarlo, con lo cual hace que al caer en ciertos estados puedan volver al estado principal o el estado inicial para que no hubiera ningún problema.

## **reiniciar(): void**

Hacíamos que cambiara el estado actual al estado inicial con el cual el autómata iniciaba siempre, y borramos todo lo que estaba dentro de la cadena que se usa para seleccionarlo y que cambiara de color.

## **getResult(): LinkedList<String>**

Con esto devolvíamos el listado de errores que tenía el código que fue escrito en el RichTextBox.