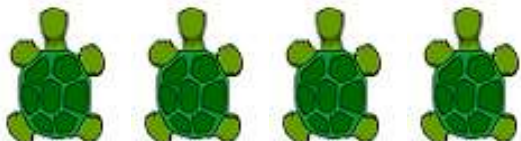
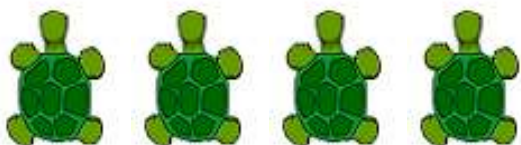


MANUAL TÉCNICO: KTURTLE



Nombre del sistema: KTURTLE

Versión del sistema: 1.0



Tipo de Manual: Manual de técnico

Fecha de elaboración: septiembre, 2017.



Área: Ciudad de Guatemala

PRESENTACIÓN

- Antecedentes

La comunidad de Edulibre requiere de un software, KTURTLE, para incluir en su nueva versión de Edulibre, S.O, el cual consista en un juego que sirva a los niños para aprender a programar.

- Objetivos

-Facilitar el proceso de enseñanza y aprendizaje de programación para niños de cortas edades.

-Realizar cálculos aritméticos así como dibujos sencillos mediante trazos generados por el movimiento de la tortuga.

-Obtener resultados de análisis (Componentes léxicos y errores.

- Introducción

Manual dirigido a miembros de la empresa Edulibre y a desarrolladores que requieren conocer la aplicación.

DESCRIPCIÓN DE ACTORES DEL SISTEMA

ACTOR DEL NEGOCIO	DESCRIPCIÓN
Usuario de la comunidad Edulibre	El usuario tendrá acceso a la aplicación en la cual a través de comandos podrá hacer que la tortuga realice acciones.

ESPECIFICACIONES DE REQUERIMIENTOS

I. Requisitos funcionales

- Realizar cálculos aritméticos así como dibujos sencillos mediante trazos generados por el movimiento de la tortuga.
- Obtener análisis de componentes léxicos y errores si existen.
- Se debe de poder abrir, crear y guardar archivos.
- El análisis léxico debe realizarse a través de un autómata finito determinista.

II. Requisitos no funcionales

- Tiempo prudente en que se tarda en realizar el proceso.
- Tolerante a errores léxicos y de estructura.
- Uso intuitivo y dinámico para el usuario.

INSTRUCCIONES PARA INTERACTUAR CON LA TORTUGA

SINTAXIS	ATAJO	UTILIDAD
Asg variable = operación;	-	El lenguaje soporta la asignación de variables para operaciones entre valores escalares
Escribir(variable);	-	Esta función imprime en pantalla el contenido de una variable.
Tamaño lienzo x,y;	tl x,y;	Puede modificar las dimensiones del lienzo. Requiere de X e Y como parámetros de entrada, donde X es el ancho en píxeles del nuevo lienzo, e Y es la altura del nuevo lienzo también en píxeles
Color lienzo color;	cl color;	Fija el color del lienzo. El parámetro de entrada pueden ser tres distintos colores: blanco, celeste y amarillo.
Avanzar X;	Avz X;	Desplaza la tortuga hacia delante X píxeles (X puede ser una variable o un número escalar). Cuando el pincel esté abajo, la tortuga deja a su paso un trazo.
Retroceder X;	Ret X;	Hace retroceder a la tortuga X píxeles (y puede ser una variable o un número escalar). Cuando el pincel esté abajo, la tortuga deja a su paso un trazo.

GirarIzq X;	Izq X;	Le ordena a la tortuga que rote X grados (X puede ser una variable o un número escalar) hacia la izquierda.
GirarDer x;	Der X;	Le ordena a la tortuga que rote X grados (X puede ser una variable o un número escalar) hacia la derecha.
Centrar;	-	Mueve la tortuga hacia el centro del lienzo;
Ir X,Y;	-	Le ordena a la tortuga que se desplace a una posición concreta del lienzo. Está posición está a X pixeles del margen izquierdo del lienzo y a Y pixeles del margen superior del lienzo;
irX X;	IX x;	Le ordena a la tortuga que vaya a una posición que está a X pixeles (X puede ser una variable o un número escalar) del margen izquierdo del lienzo, sin modificar su altura.
irY y;	IX y;	Le ordena a la tortuga que vaya a una posición que está a Y pixeles (Y puede ser una variable o un número escalar) del margen izquierdo del lienzo, sin modificar su altura.
subirPincel;	spl;	Alza el pincel del lienzo. Cuando el pincel esté arriba la tortuga no deja ningún trazo cuando se desplaza.
bajarPincel;	Bpl;	Presiona el pincel sobre el lienzo. Cuando el pincel esté abajo la tortuga deja un trazo a su paso.
colorPincel;	Cpl;	Fija el color del pincel. El color de entrada puede ser tres distintos colores, rojo azul y negro.

CASOS DE USO

Abrir archivo	Función destinada a abrir un archivo guardado dentro del sistema del usuario
Actores	Usuario
Resumen	El proceso se inicia mediante la pestaña Archivo>Cargar archivo. En dicha función mediante un OpenFileDialog se le da opción al usuario de escoger el archivo a abrir, si la extensión y dirección es la correcta se lee el archivo y se agrega al cuadro de texto activo.

Guardar archivo y guardar como	Función destinada a guardar el archivo del cuadro de texto activo.
Actores	Usuario
Resumen	El proceso se inicia mediante la pestaña Archivo>Guardar Archivo. En dicha función mediante un SaveFileDialog se da opción al usuario de escoger la ruta donde se guardará el archivo si la ruta es correcta mediante un StreamWriter se guarda el archivo en la ruta especificada.

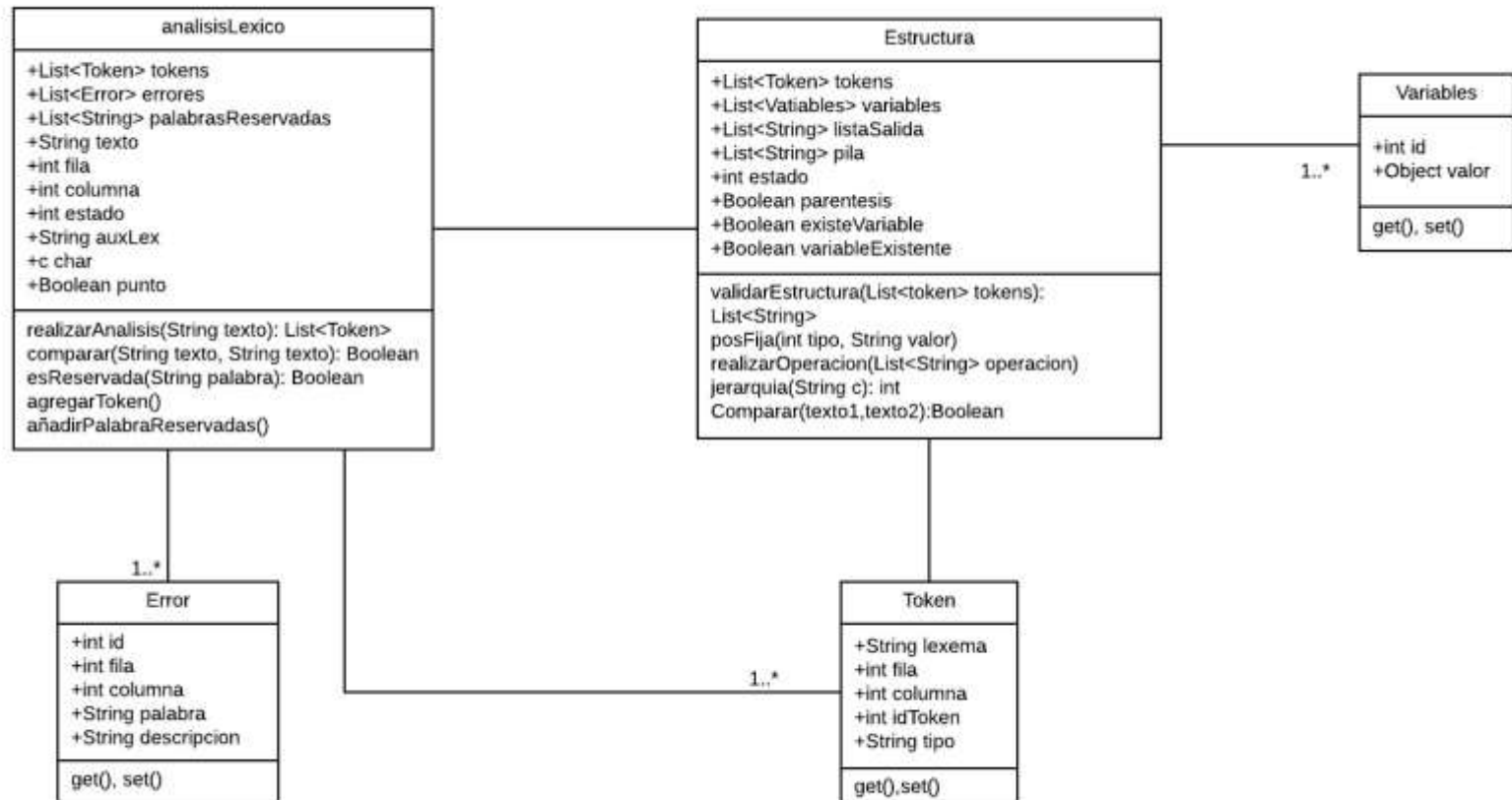
Run	Función destinada a realizar las acciones descritas a través del código fuente.
Actores	Usuario y sistema.
Resumen	El proceso se inicia mediante la pestaña "Run". Dicha función genera una instancia de la clase analisisLexico, convoca al evento de realizarAnalisis que mediante un autómata finito determinista realizará el análisis léxico dividiendo la entrada en tokens. Posteriormente desde la clase principal se invocara al método obtener errores si se encuentran errores léxicos el proceso se detiene sino mediante la lista de tokens se verificara la estructura y se construirán las variables correspondientes para posteriormente hacer un recorrido a través de la lista en el cual se comparan los tokens encontrados con las palabras reservadas y dependiendo del token encontrado se producirá la acción correspondiente.

Reportes	Función destinada a producir los archivos de salida correspondiente.
Actores	Usuario
Resumen	<p>El proceso se inicia mediante la pestaña Reportes>Reportes tokens o Reportes>Reporte Errores.</p> <p>Para ellos se crea una plantilla para una tabla de salida HTML y se recorre la lista de tokens o errores dando el formato de estructura correspondiente para la tabla. Al terminar de producirse el archivo se abre al usuario a través de su navegador predeterminado.</p>

Ayuda	Compuesta de las opciones: Manual de usuario, manual técnico y acerca de, da acceso a la documentación propia del sistema
Actores	Usuario
Resumen	<p>El proceso se inicia mediante la pestaña Ayuda. Mediante dichas funciones se abren los documentos específicos donde se brinda la documentación necesaria para comprender el sistema.</p>

MODELO LÓGICO DE DATOS

DIAGRAMA DE CLASES

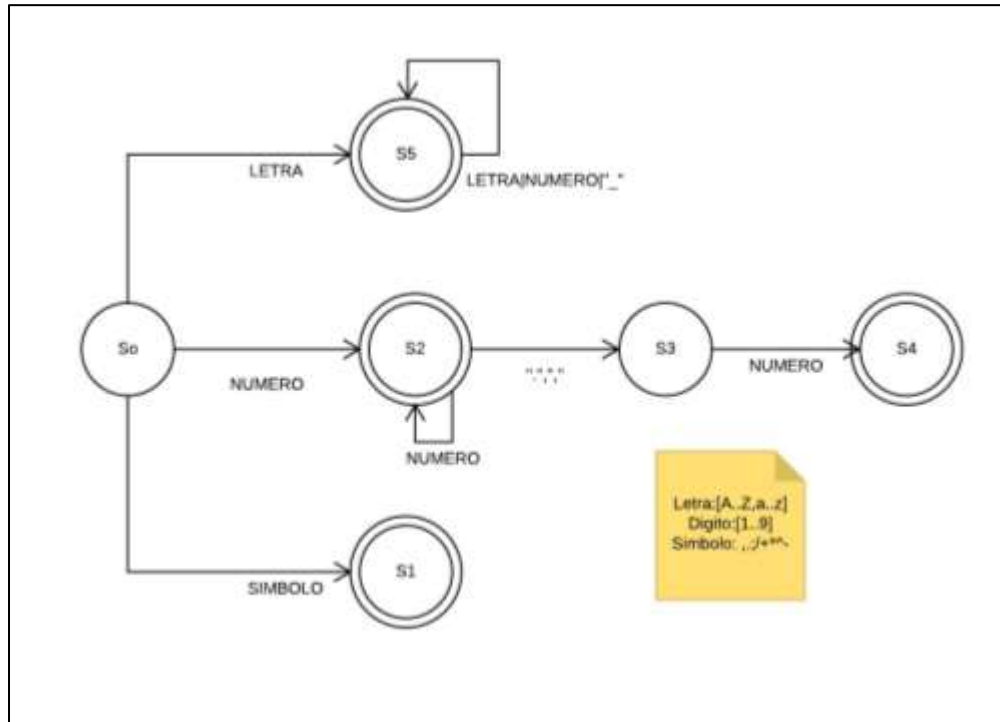


Descripción de clases

- Token: Clase que estructura la información de los componentes léxicos, se conecta al reporte de TablaSimbolos para brindar los datos de las columnas correspondientes. El analizador léxico establecerá los componentes léxicos propios del texto a analizar.
- Errores: Clase que estructura la información de los errores léxicos a encontrar por el proceso realizado a través del analizador léxico, se conecta al reporte de Errores para brindar los datos de las columnas correspondientes.
- Variables: Estructura la información de las variables para luego ser añadida como tal a la tabla de símbolos. Esta clase incluye tanto el identificador como el nombre de la variable.
- analisisLexico: Realiza el análisis léxico y de estructura en base al texto a analizar. En esta clase están declaradas las palabras reservadas del sistema que incluye la lista de instrucciones que harán los movimientos e interactividad entre tortuga y usuario. El análisis léxico se realiza mediante un autómata finito determinista.
- Estructura: Esta clase se encarga de en base a la lista de tokens encontrada mediante la clase de análisis léxico, valida la sintaxis de las variables para construirlas ya que estas funcionaran a todo lo largo del programa. Las operaciones aritméticas son realizadas mediante notación posfija, para pasar la notación infija a posfija se utiliza una cola y lista de salida. Una vez realizado esta operación se van añadiendo cada valor a una nueva pila que al encontrar un operador, hará la operación de los últimos dos valores ingresados en la pila.

DESCRIPCIÓN DETALLADA DE ALGORITMO

AUTOMATA FINITO DETERMINISTA



ALGORITMO AFD

Estado = 0 AuxiliarLexico = "";

Para i=0 hasta Tamaño_Texto_Entrada hacer

 c = Tamaño_Texto_Entrada(i);

 Según (Estado) hacer:

 Caso 0:

 Si c es letra

 AuxiliarLexico = AuxiliarLexico + c;

 Estado = 1;

 Sino Si c es digito

 AuxiliarLexico = AuxiliarLexico + c

 Estado = 2;

 Sino Si c es Simbolo:

 Crear Token

 Sino marca error;

 Caso 1:

 Si c es letra o digito o " _ "

 AuxiliarLexico = AuxiliarLexico + c;

 Sino

 Crear Token

 Estado = 0;

Caso 2:

Si c es digito

AuxiliarLexico = AuxiliarLexico + c;

Sino si(c es "." Y no se ha encontrado punto antes)

AuxiliarLexico = AuxiliarLexico + c;

Sino

Crear Token

Estado = 0;

Fin Según

Fin Para

Fin

GLOSARIO DE TERMINOS

TÉRMINO	DEFINICIÓN
Compilador	Programa que traduce un programa escrito en código fuente a un programa equivalente en código destino.
Token	Componente léxico con un significado colectivo.
Lexema	Secuencia de caracteres del programa fuente que concuerdan con el patrón de un componente léxico
Reservadas	Lexema que tiene una función previamente definido por el programa por lo cual no puede usarse para otro objeto.
Análisis léxico	Fase del compilador el cual consiste en leer el código fuente carácter a carácter y agruparlos en componentes léxicos
Autómata finito determinista (AFD)	Autómata finito en el cual para cada estado en que se encuentra, existe no más de una transición posible. Es una quintupla formada por: Estados, alfabeto, estado inicial, función de transición y estados de aceptación.
Método del árbol	Método para producir autómatas finitos más ordenados, eficientes y reducidos.
Tabla de símbolos	estructura de datos que usa el proceso de traducción de un lenguaje de programación, por un compilador o un intérprete, donde cada símbolo en el código fuente de un programa está asociado con información tal como la ubicación, el tipo de datos y el ámbito de cada variable,