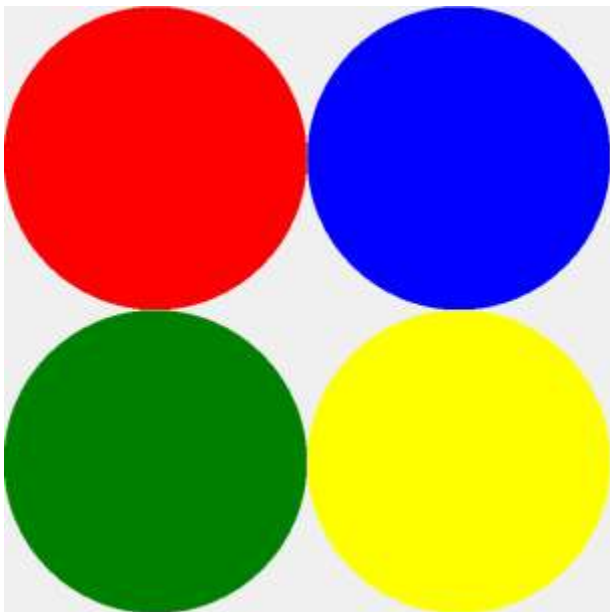


MANUAL TÉCNICO: VP 502



Nombre del Sistema: Vallas 502

Versión del Sistema: 1.0

Tipo de Manual: Manual Técnico

Fecha de elaboración: Agosto, 2017

Área: Ciudad de Guatemala

PRESENTACIÓN

- **Antecedentes**
La empresa Valla 502 se dedica a la creación de vallas publicitarias y requiere de un software que reconozca el lenguaje propio y genere el diseño de la valla solicitada.
 - **Objetivos**
 - Generar el diseño de una valla solicitada a partir de un lenguaje.
 - Obtener resultados de análisis (Componentes léxicos y errores).
 - **Introducción**
Manual dirigido a miembros de la empresa Vallas 502 y a desarrolladores que requieren conocer la aplicación.
-

DESCRIPCIÓN ACTORES DEL SISTEMA

ACTOR DEL NEGOCIO	DESCRIPCION
Usuario de empresa	El miembro de la empresa tendrá acceso al editor de texto el cual utilizará para generar la valla publicitaria deseada a partir del lenguaje establecido de forma previa.

ESPECIFICACIÓN DE REQUISITOS

- I. **Requisitos funcionales**
 - Generar valla publicitaria a través de un lenguaje propio de la empresa.
 - Obtener análisis de componentes léxicos y errores si existen.
 - Se debe de poder abrir, crear y guardar archivos.
 - Se pueden abrir y crear n pestañas dentro del editor.
 - El análisis léxico debe realizarse a través de un autómata finito determinista.
- II. **Requisitos no funcionales**
 - El nombre de la pestaña activa debe tener el nombre del archivo si existe ya ha sido creado o guardado anteriormente.
 - Los archivos de salida deben ser mostrados a través de una página HTML o a través de un reporting services.

LENGUAJE PARA CONSUTRICCIÓN DE VALLA

INSTRUCCION	UTILIDAD
<code><valla></code> ... <code></valla></code>	Es la etiqueta inicial del archivo, sirve para que el analizador sepa que se está iniciando el diseño de una valla publicitaria
<code><empresa></code> ... <code></empresa></code>	Esta etiqueta sirve para indicarle al analizador el nombre de la empresa para la cual se hace el diseño de la valla. (Puede ser cualquier nombre, incluso con dígitos y caracteres especiales como: '_', '@', '#', '&').
<code><fondo></code> ... <code></fondo></code>	Esta etiqueta sirve para indicarle al analizador toda la información que necesita para crear el fondo de la valla.
<code><tamano></code> ... <code></tamano></code>	Esta etiqueta sirve para indicarle al analizador el tamaño del tablero
<code><horizontal></code> ... <code></horizontal></code>	Esta etiqueta sirve para indicarle al analizador la cantidad del tamaño horizontalmente.
<code><vertival></code> ... <code></vertical></code>	Esta etiqueta sirve para indicarle al analizador la cantidad del tamaño verticalmente.
<code><color></code> ... <code></color></code>	Esta etiqueta sirve para indicarle al analizador el color que se utiliza para el pixel o fondo.
<code><pixel></code> ... <code></pixel></code>	Esta etiqueta sirve para indicarle al analizador la información del pixel que se pintará en el tablero Esta etiqueta podrá venir en el archivo de entrada tantas veces como pixeles sean necesarios para pintar el diseño.
<code><posicionx></code> ... <code></posicionx></code>	Esta etiqueta sirve para indicarle al analizador la posición en X del pixel que formará la palabra de la valla.
<code><posiciony></code> ... <code></posiciony></code>	Esta etiqueta sirve para indicarle al analizador la posición en Y del pixel que formará la palabra de la valla

CASOS DE USO

NUEVA PESTAÑA	Función destinada a crear una nueva pestaña dentro del editor de texto.
Actores	Usuario
Resumen	El proceso se inicia mediante la pestaña Archivo>Nueva Pestaña. En dicha función se llama a crearPestaña() la cual añade un nuevo RichTextBox dentro de la lista de cuadros de textos y añade una nueva pestaña al tab control. A ambos componentes se les anida los métodos característicos: a la pestaña el evento de cuando el usuario cambie de pestaña para distinguir a cuál se dirige y al cuadro de texto para llevar control de qué líneas se trabajan y cómo cambia el texto conforme se trabaja.

CARGAR 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 la 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	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.

ANALIZAR LEXICAMENTE	Función destinada a analizar el texto del cuadro de texto activo para crear la valla publicitaria.
Actores	Usuario
Analizar léxicamente	<p>El proceso se inicia mediante la pestaña Compilar> Analizar léxicamente. Dicha función crea una instancia de la clase <code>analisiLexico</code> y convoca a dos eventos: <code>realizarAnalisis</code> y <code>obtenerErrores</code>.</p> <p>La función <code>realizarAnaliss</code> recibe como parámetro el texto del cuadro de texto activo y mediante el autómata finito determinista (Ver AUTOMATA FINITO DETERMINISTA y ALGORITMO AFD) crea los componentes léxicos del cual se compone el texto ingresado.</p> <p>El evento <code>obtenerErrores</code> devuelve la lista de errores encontrados en la fase del análisis.</p> <p>Si no se encontraron errores léxicos en la fase del primer análisis se procede a realizar un análisis estructural, en este mediante un autómata sintáctico (Ver AUTOMATA SINTACTICO) se verifica la estructura del texto ingresado, además se utiliza una pila para verificar que cada instrucción tenga un final. El análisis de estructura se hace mediante el identificador asignado a cada componente léxico, se devuelven los errores encontrados.</p> <p>Si no se encontraron errores léxicos ni estructurales se lanza una alerta de que la valla puede ser creada por otro lado si se encuentran errores se lanza una alerta en la terminal de dónde se encuentran estos.</p> <p>Al realizar el análisis léxico se cambia de valor a la bandera <code>Analisis</code> que indica si ya se ha realizado el análisis léxico o no.</p>

GENERAR VALLA PUBLICITARIA	Función que permite la creación de la valla publicitaria en torno al texto ingresado por el usuario en el editor.
Actores	Usuario
Resumen	El proceso se inicia mediante la pestaña Compilar> Generar Valla Publicitaria. Para poder realizar la valla publicitaria al estar verificada la estructura se procede a verificar que valor pertenece a qué atributo, se compara el token encontrado con las palabras reservadas si coincide con alguna, el token que esté dos posiciones después (en concordancia con la estructura) será el valor del atributo dado a la palabra reservada. Una vez verificados todos los tokens, se añade a la valla los atributos encontrados para proceder a su construcción.
Pre-condiciones	Haber realizado el análisis léxico previamente (Compilar> Analizar Lexicamente) y que no existen errores léxicos ni estructurales.

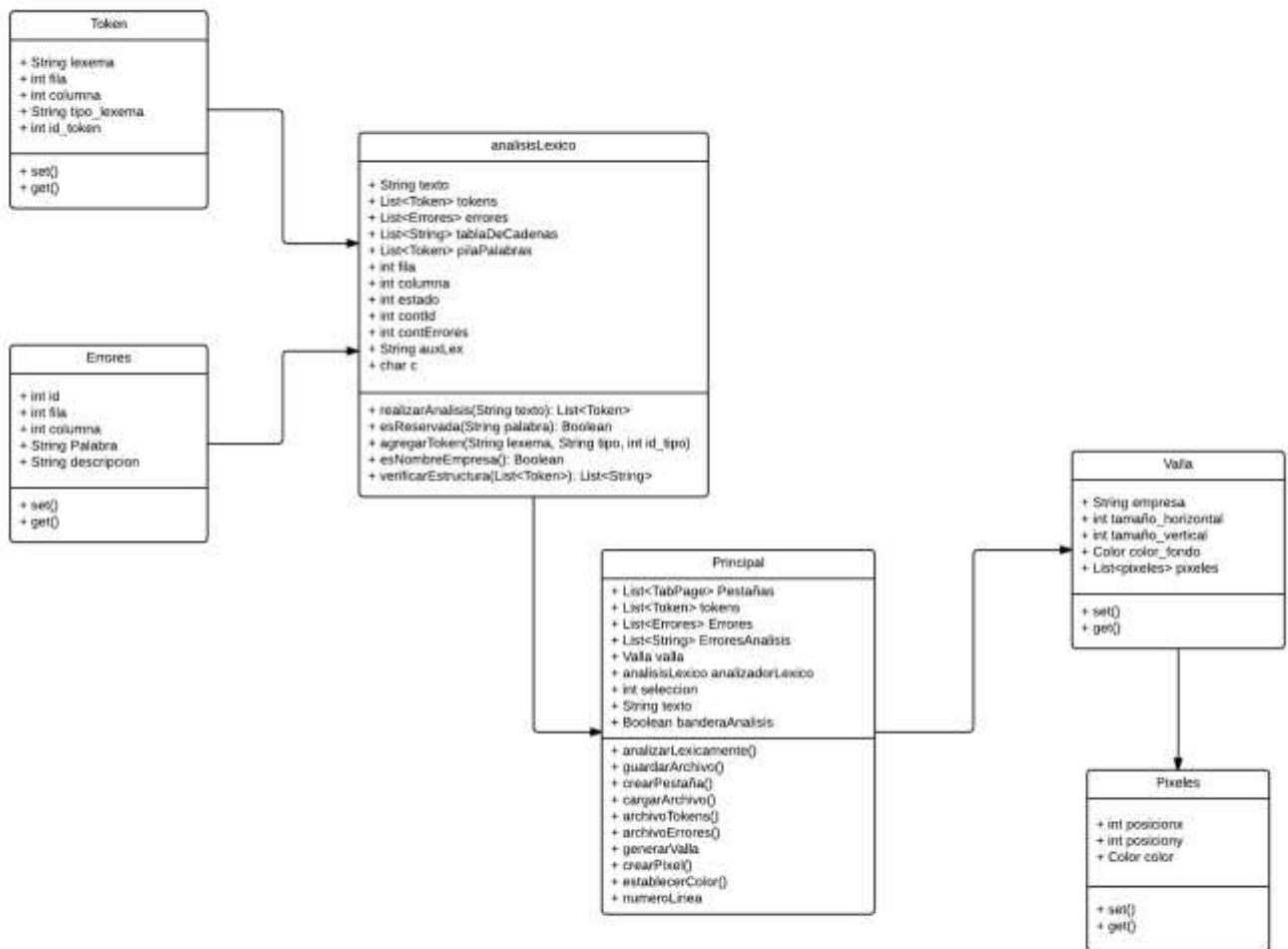
ARCHIVOS DE SALIDA	Función que muestra mediante un Reporting Services los componentes léxicos y errores encontrados.
Actores	Usuario
Resumen	El proceso se inicia mediante la pestaña Compilar> Archivos de salida. Para esta función se hace una instancia de la forma donde se mostrará la salida pasando como parámetro la lista correspondiente En la forma correspondiente hay un componente ReportViewer que se conecta con su reporte correspondiente (LocalReport). Dicho reporte tiene como fuente de datos un Objeto (en caso de los tokens, la clase Token y en caso de Errores la clase Errores). Que da o fija la estructura de cómo estará compuesto el reporte En la forma, mediante el evento Load se establece limpiar el reporte y se brinda una

	nueva entrada de datos. Posteriormente se muestra el reporte realizado.
Pre-condiciones	Haber realizado previamente el análisis léxico del texto trabajado.
Requisitos especiales	<p>Procesador i3 en adelante ya que la creación del Reporting services consume recursos.</p> <p>De parte del desarrollador Ingresar en el Nuget Package Manager Console: Install-Package Microsoft.ReportingServices.ReportViewer Control.WinForms para tener el componente de ReportViewer (Para más información ver bibliografía: I. Report Viewer)</p> <p>Y tener instalado el componente: "Diseñador de informes RLDC de Microsoft para Visual Studio".</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	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.

MODEO LÓGICO DE DATOS

DIAGRAMA DE CLASES



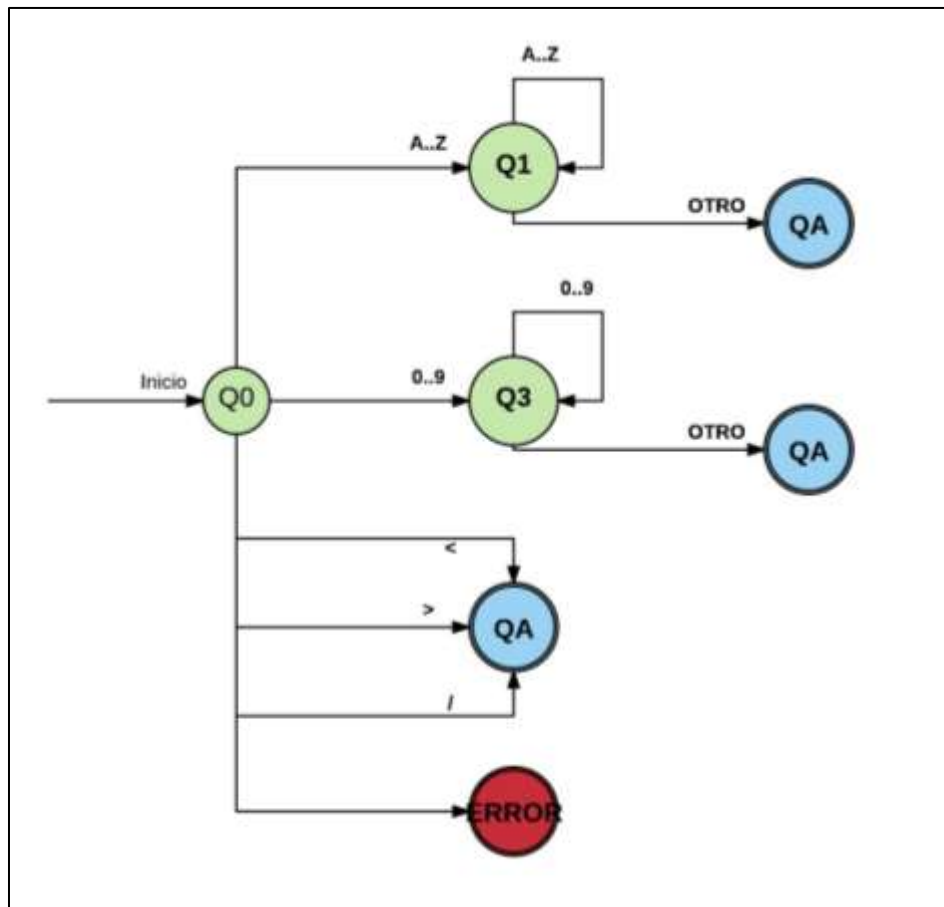
DEFINICIÓN DE CLASES

- 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.
 - analizadorLexico: Realiza el análisis léxico y de estructura en base al texto a analizar. En esta clase están declaradas las palabras reservadas del sistema que le dan forma a la valla. El análisis léxico se realiza mediante un autómata finito determinista mientras el análisis estructural se realiza mediante un autómata finito sintáctico. Se hacen las validaciones correspondientes para el nombre de la empresa.
 - Principal: Esta clase es la encargada del traspaso de datos entre la interfaz y las clases que componen al sistema. Convoca a las distintas funciones para que el sistema realice sus funciones específicas. Además en la generación de la valla, establece los valores para cada atributo en la creación de la valla.
 - Valla: Clase destinada a dar la estructura de los datos de cómo se compondrá la valla publicitaria. Al ser establecidos los valores de cada atributo por la clase principal se redirige a una nueva forma en donde mediante Graphics se construye la valla establecida a través de código.
 - Píxeles: Clase que da la estructura de los píxeles a pintar que contendrá la valla. Esta clase contiene la ubicación en x y y respecto al número total del tamaño y el color que contendrán. Los píxeles son añadidos a una lista de píxeles en la clase Valla para que en la construcción de la Valla se recorra dicha lista y se pinten los espacios correspondientes.
-

DESCRIPCIÓN DETALLADA DE ALGORITMOS

AUTOMATA FINITA DETERMINISTA (AFD)



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 <,>, /

Crear Token
Sino marca error;

Caso 1:

Si c es letra

AuxiliarLexico = AuxiliarLexico + c;

Sino

Crear Token

Estado = 0;

Caso 2:

Si c es digito

AuxiliarLexico = AuxiliarLexico + c;

Sino

Crear Token

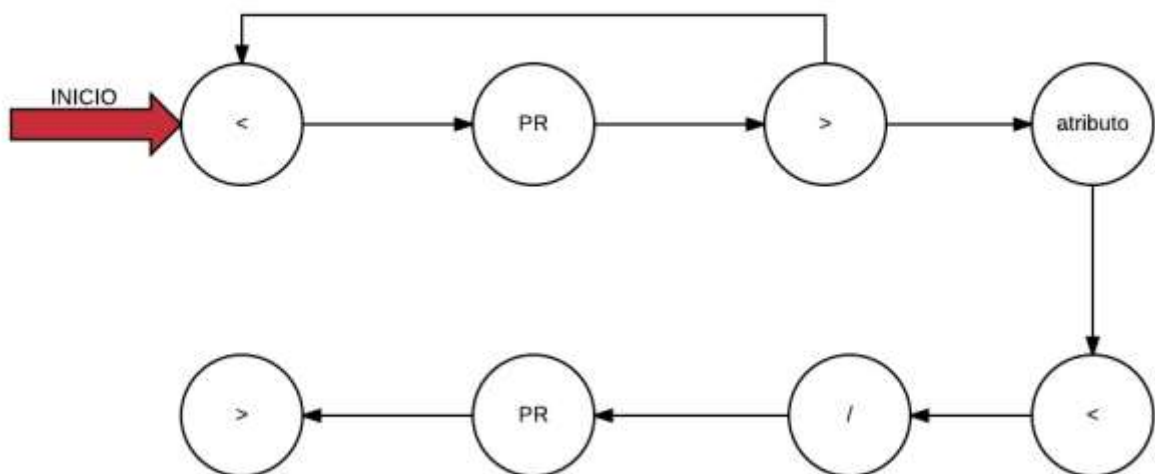
Estado = 0;

Fin Según

Fin Para

Fin

AUTOMATA SINTÁCTICO



DISEÑO DE PANTALLAS Y REPORTE

CONSIDERACIONES

- Se debe de poseer el componente reportViewer.
- Se debe tener instalado el complemento: Diseñador de informes RLDC de Microsoft para Visual Studio que puede ser instalado en el menú de extensiones y actualizaciones de Visual Studio.
- Los reportViewer deben ir conectados con los reportes locales creados.
- Los reportes locales su fuente de datos es un objeto dentro del sistema.
- Los services reporting necesitan de un tiempo prudencial para ser creados.
- En la creación de la valla el tamaño total en x será tamaño_horizontal*20 y el tamaño total en y será tamaño_vertical*20

GLOSARIO DE TÉRMINOS

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.
Reporting services	Plataforma que permite realizar informes empresariales cuyo contenido se extrae de una variedad de orígenes de datos.

FUNDAMENTOS DE LA TECNOLOGÍA UTILIZADA

REPORTING SERVICES

- <https://docs.microsoft.com/en-us/sql/reporting-services/application-integration/integrating-reporting-services-using-reportviewer-controls-get-started>
- <http://joseluisgarciab.blogspot.com/2013/10/reportviewer-y-rdlc-ejemplo-facturacion.html>

ANÁLISIS LÉXICO

- Compiladores, principios, técnicas y herramientas- Alfred V. Aho.

TEORÍA DE AUTÓMATAS

- Teoría de autómatas, lenguajes y computación: John E. Hopcroft
- Teoría de la computación: J. Glenn Brookshear