

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS Y SISTEMAS  
LENGUAJES FORMALES Y DE PROGRAMACION (A)  
ING. LUIS ESPINO  
AUX. ALEJANDRO LEON

---

*MANUAL TECNICO*

*PROYECTO: AVATAR*

---

EDUARDO ANTONIO GARCIA FRANCO  
2012-12961  
15/06/15

---

## INTRODUCCION

---

Avatar es un analizador léxico bastante simple en el cual podremos ingresar unas líneas de código y posteriormente se podrán analizar para generar en un cuadro ciertos aspectos de una persona, estos aspectos estarán regidos por las palabras reservadas que están definidas en la aplicación.

El uso correcto de las etiquetas es el que permitirá que el código se ejecute correctamente y muestre así lo que el usuario ingreso, también aparte de eso se generaran dos archivos con la extensión .HTML los cuales se abrirán en el navegador que se tenga determinado, el primero de ellos contendrá los lexemas de las líneas de código escritas correctamente, por el contrario el otro contendrá los errores que hayan sido encontrados en las líneas de código erróneas.

El uso de esta aplicación es bastante simple no requiere grandes conocimientos sobre programación o solo comprender como es la estructura de las líneas de código fuera de eso no hay gran complejidad al momento de hacer uso de la aplicación.

---

## DESCRIPCION GENERAL DE LA APLICACIÓN

---

Esta aplicación es bastante sencilla cualquier persona puede manejarla ya que no es demasiado compleja para utilizarla, posee unas palabras reservadas que van acorde a las etiquetas correspondientes siendo las siguientes todas las palabras que reconoce esta aplicación:

Etiquetas	Palabras Reservadas
<avatar></avatar>	Se colocan al inicio y final del código, luego puede venir cualquier etiqueta.
<usuario></usuario>	No hay palabra reservada, solo deberá introducir el texto dentro de comillas.
<edad></edad>	No hay palabra reservada, solo deberá introducir números.
<complexión></complexión>	Gordo, delgado, normal
<personalidad></personalidad>	Enojado, alegre, neutro
<sexo></sexo>	Femenino y masculino

Solo se podrá usar las expresiones regulares dentro de sus etiquetas correspondientes, con la restricción de que solo podrá ser una de lo contrario si no se cumple esta regla se marcara como error.

---

## DESCRIPCION DE CLASES Y METODOS

---

- Clase: Ventana

Esta clase será la interfaz con la que interactuara el usuario, siendo los siguientes los métodos con los que se analiza el código.

Private void Unir (): la función es que va a retornar el texto ingresado en el área para introducir el código en una sola línea de esta forma no importa si el usuario ingresa el código en una sola línea, ejecutara el código igual. La forma de salida del código estaría dividida por el símbolo "%" siendo el siguiente un ejemplo de la salida: <etiqueta>%palabra/texto%<etiqueta>.

Private void Analizar(StringCodigo): la función es que analiza el código previamente unido en una línea con el método "unir" verifica primeramente que existan las etiquetas de "avatar" en el código de lo contrario no funcionara seguidamente verifica si existen las etiquetas respectivas de las palabras reservadas, si estuvieran mal escritas se ignorara la línea y pasara a la siguiente por el contrario si la etiqueta de abrir esta correcta lo siguiente en verificar seria la etiqueta de cerrado si es incorrecta se ignorara la línea, pero si es correcta se pasara a revisar el contenido dentro de las etiquetas ver si pertenece a las palabras reservadas o si existen símbolos necesarios para el cumplimiento de las mismas, cuando encuentre un error o este todo o algunas etiquetas correctas se generara un archivo HTML en el cual estarán tabuladas y especificados los errores y las palabras correctas.

Parámetro: es el código unido en una sola línea, por el método "unir".

Boolean EtiquetaAbre (String Dato) y boolean EtiquetaCierre (String Dato):

Ambos métodos funcionan de la misma manera uno es para verificar que la etiqueta con la que abre la línea de código este correcta, el otro es para verificar que la etiqueta con la que cierra la línea de código este correcta.

Parámetro: ambas reciben la etiqueta a analizar.

Retorno: retornaran "true" si es correcta y "false" si no lo es.

#### Boolean PalabraReservada (String Dato, String Etiqueta):

Verifica el contenido entre las etiquetas dentro de las palabras que están reservadas para esta aplicación.

Parámetro: recibirá dos parámetros, el primero será la palabra a analizar, la segunda será la etiqueta con la que se está encerrando para comprobar que ambas están usándose correctamente.

Retorno: retornara "true" si la palabra y la etiqueta pertenecen al mismo grupo y "false" si no lo son.

#### Boolean esnumero (String Dato):

Verifica si el contenido de la etiqueta "edad" es realmente un número.

Parámetro: recibirá el contenido dentro de la etiqueta.

Retorno: retornara "true" si se ha ingresado un número y "false" si es cualquier otro carácter.

#### Void Correcto (String Dato, String Etiqueta):

Este método nos servirá para ejecutar lo que está dentro de las etiquetas correctamente escritas.

Parámetro: recibirá dos, el primero es la palabra reservada y el segundo la etiqueta de la cual proviene dicha etiqueta.

#### Void ObtenerRutaDeArchivo ():

La función de este método es la de navegar entre los directorios de nuestra computadora para elegir un archivo en el cual contenga código para la lectura en esta aplicación.

#### Public void actionPerformed (ActionEvent e):

En este método va todo lo relacionado a la ejecución al momento de presionar los botones.

- Clase: Archivo

#### Public String AbrirARCHIVO (String ruta):

Mediante este método se leerá el contenido del archivo que el usuario eligió y se enviara a la clase "ventana" para su posterior analizacion.

Parámetro: será la ruta del archivo que elegimos en el método "obtenerrutadearchivo" en la clase "ventana".

Retorno: devolverá todo el contenido dentro del archivo seleccionado como un String.

#### Public void Guardar (String texto, String Ruta, boolean abierto):

La función es sobrescribir el archivo que haya elegido el usuario abrir o darle la oportunidad de crear un archivo nuevo si eligió ingresar código de manera manual.

Parámetro: el primero es el contenido dentro del área para para ingresar texto, el segundo la ruta donde se encuentra ubicado el archivo y el tercero un booleano para saber si se abrió un archivo o no.

#### Public void GuardarComo (String texto):

La función es después de abrir un archivo si no se desea sobre escribir este archivo se le da la opción al usuario de crear uno Nuevo.

Parámetro: recibirá el contenido en el área de texto.

#### Public void GenerarHTMLLexemas (String datos) y public void GenerarHTMLErrores (String datos):

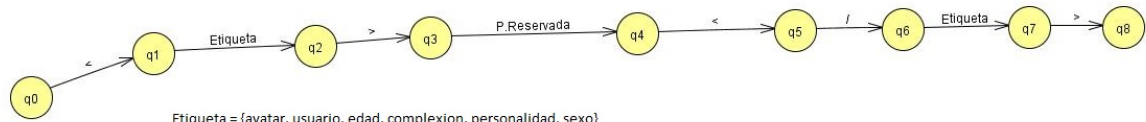
Como su nombre lo indica estos métodos serán los encargados de generar el archivo HTML con la tabla correspondiente, ya sea de errores o de lexemas escritos correctamente.

Parámetro: ambas reciben el contenido del área de texto.

---

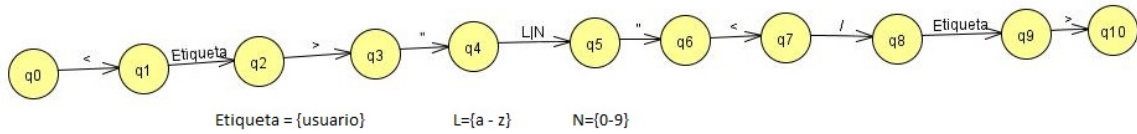
# AUTOMATAS

---



Etiqueta = {avatar, usuario, edad, complexión, personalidad, sexo}

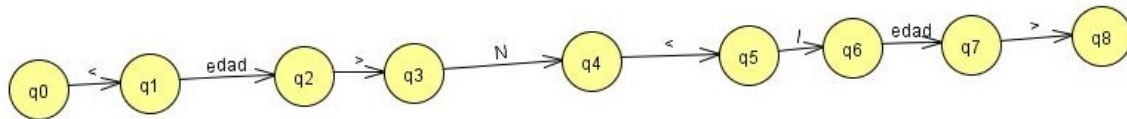
P.Reservada = {enojado, alegre, neutro, masculino, femenino, delgado, normal, gordo, número, "texto"}



Etiqueta = {usuario}

L={a - z}

N={0-9}



N = {0 - 9}