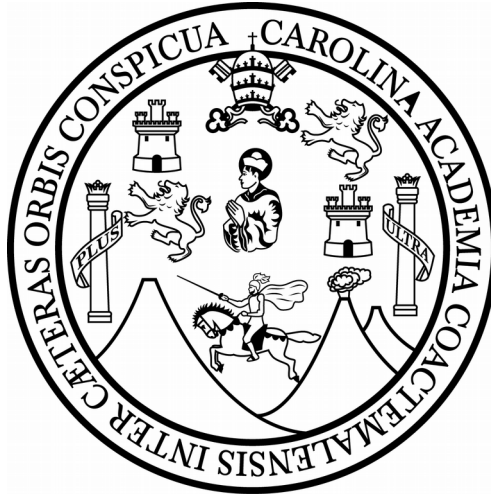


CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA CIENCIAS Y SISTEMAS



LABORATORIO DE ORGANIZACIÓN LENGUAJES Y COMPILADORES 1

“QUINTO SEMESTRE”

ING.: JOSÉ MOISÉS GRANADOS GUEVARA

AUX: EDVIN TEODORO GONZALEZ RAFAEL

ESTUDIANTE: Bryan René Gómez Gómez – 201730919

PROYECTO: “Manual Técnico - PROYECTO FINAL - ANIMATIONS”

FECHA: 27 de Mayo de 2,020

Analizador Léxico (Lienzo, Colores, Tiempo)

TOKEN	TIPO
LIENZOS	Palabra Reservada
COLORES	Palabra Reservada
TIEMPOS	Palabra Reservada
nombre	Palabra Reservada
tipo	Palabra Reservada
Fondo	Palabra Reservada
Red	Palabra Reservada
Blue	Palabra Reservada
Green	Palabra Reservada
tamaño	Palabra Reservada
cuadro	Palabra Reservada
dimension_x	Palabra Reservada
dimension_y	Palabra Reservada
HEX	Palabra Reservada
inicio	Palabra Reservada
fin	Palabra Reservada
imagenes	Palabra Reservada
id	Palabra Reservada
duracion	Palabra Reservada
“	Signo Comillas
:	Signo Dos Puntos
,	Signo Coma
{	Signo Corchete Abierto
}	Signo Corchete Cerrado
[Signo Llave Abierto
]	Signo Llave Cerrado
(Letra “_”) → (Letra “_” Dígito)*	Expresión Identificador
(1)? (Dígito (1 -99))	Expresión Número RGB
(2)? (Dígito (1 – 55))	Expresión Número RGB
Dígito+	Expresión Número
Comillas → (“png” “gif”) → Comillas	Expresión Extensión
Comillas → (Letra Símbolo Dígito) → Comillas	Expresión Cadena
“#” → (([a-fA-F] (0-15)) Max. 6)	Expresión Número Hexadecimal

Analizador Sintáctico
Gramática
(Lienzo, Colores, Tiempo)
G = [N, T, P, S]

Terminal (T) :

Lienzos, Colores, Tiempos, Nombre, Tipo, Fondo, Red, Blue, Green, TamañoR, Tamaño, Cuadro, DimensionX, DimensionY, Hex, Inicio, Fin, Imagenes, Id, Duracion, Comillas, DosPuntos, Coma, CorcheteA, CorcheteC, LlaveA, LlaveC, Identificador, NumeroRGB_1, NumeroRGB_2, Numero, Extension, NombreLienzo, Hexadecimal

No Terminal (N) :

inicio, estructura_lienzo, estructura_colores, estructura_tiempo, declaracion_lienzo, struct_lienzo, declaracion_fondo, fondo_rgb, fondo_hex, declaracion_tamaño, estructura_tamaño, declaracion_colores, struct_id_color, declaracion_tiempo, struct_id_tiempo, estructura_imagenes, declaracion_duracion

S = inicio

Producciones (P) :

inicio

- estructura_lienzo
- estructura_colores
- estructura_tiempo

estructura_lienzo

- CorcheteA Lienzos DosPuntos CorcheteA declaracion_lienzo CorcheteC CorcheteC

declaracion_lienzo

- Identificador DosPuntos CorcheteA struct_lienzo Coma struct_lienzo Coma struct_lienzo Coma struct_lienzo CorcheteC
- declaracion_lienzoComa Identificador DosPuntos CorcheteA struct_lienzo Coma struct_lienzo Coma struct_lienzo Coma struct_lienzo CorcheteC

struct_lienzo

- Nombre DosPuntos NombreLienzo
- Tipo DosPuntos Extension
- declaracion_fondo
- declaracion_tamaño

declaracion_fondo

- Fondo DosPuntos CorcheteA fondo_rgb Coma fondo_rgb Coma fondo_rgb CorcheteC
- Fondo DosPuntos CorcheteA fondo_hex CorcheteC

fondo_rgb

- Red DosPuntos NumeroRGB_1
- Red DosPuntos NumeroRGB_2
- Blue DosPuntos NumeroRGB_1
- Blue DosPuntos NumeroRGB_2
- Green DosPuntos NumeroRGB_1
- Green DosPuntos NumeroRGB_2

fondo_hex

- Hex DosPuntos Hexadecimal

declaracion_tamaño

- Tamaño DosPuntos CorcheteA estructura_tamaño Coma estructura_tamaño Coma estructura_tamaño CorcheteC
- TamañoR DosPuntos CorcheteA estructura_tamaño Coma estructura_tamaño Coma estructura_tamaño CorcheteC

estructura_tamaño

- Cuadro DosPuntos Numero
- Cuadro DosPuntos NumeroRGB_1
- Cuadro DosPuntos NumeroRGB_2
- DimensionX DosPuntos Numero
- DimensionX DosPuntos NumeroRGB_1
- DimensionX DosPuntos NumeroRGB_2
- DimensionY DosPuntos Numero
- DimensionY DosPuntos NumeroRGB_1
- DimensionY DosPuntos NumeroRGB_2

estructura_colores

- CorcheteA Colores DosPts CorcheteA declaracion_colores CorcheteC CorcheteC

declaracion_colores

- Identificador DosPuntos CorcheteA CorcheteC
- Identificador DosPuntos CorcheteA struct_id_color CorcheteC
- declaracion_colores Coma Identificador DosPuntos CorcheteA struct_id_color CorcheteC
- declaracion_colores Coma Identificador DosPuntos CorcheteA CorcheteC

struct_id_color

- Identificador DosPuntos CorcheteA fondo_rgb Coma fondo_rgb Coma fondo_rgb CorcheteC
- Identificador DosPuntos CorcheteA fondo_hex CorcheteC
- struct_id_color Coma Identificador DosPuntos CorcheteA fondo_rgb Coma fondo_rgb Coma fondo_rgb CorcheteC
- struct_id_color Coma Identificador DosPuntos CorcheteA fondo_hex CorcheteC

estructura_tiempo

- CorcheteA Tiempos DosPuntos CorcheteA declaracion_tiempo CorcheteC CorcheteC

declaracion_tiempo

→ Identificador DosPuntos CorcheteA struct_id_tiempo Coma struct_id_tiempo Coma struct_id_tiempo CorcheteC
→ declaracion_tiempo Coma Identificador DosPuntos CorcheteA struct_id_tiempo Coma struct_id_tiempo Coma struct_id_tiempo CorcheteC

struct_id_tiempo

→ Inicio DosPuntos NombreLienzo
→ Inicio DosPuntos Comillas Identificador Comillas
→ Fin DosPuntos NombreLienzo
→ Fin DosPuntos Comillas Identificador Comillas
→ Imagenes DosPuntos LlaveA estructura_imagenes LlaveC
→ Imagenes DosPuntos LlaveA LlaveC

estructura_imagenes

→ CorcheteA declaracion_duracion CorcheteC
→ estructura_imagenes Coma CorcheteA declaracion_duracion CorcheteC

declaracion_duracion

→ Id DosPuntos Comillas Identificador Comillas Coma Duracion DosPuntos Numero
→ Id DosPuntos Comillas Identificador Comillas Coma Duracion DosPuntos NumeroRGB_1
→ Id DosPuntos Comillas Identificador Comillas Coma Duracion DosPuntos NumeroRGB_2
→ Duracion DosPuntos Numero Coma Id DosPuntos Comillas Identificador Comillas
→ Duracion DosPuntos NumeroRGB_1 Coma Id DosPuntos Comillas Identificador Comillas
→ Duracion DosPuntos NumeroRGB_2 Coma Id DosPuntos Comillas Identificador Comillas

ANALIZADOR LÉXICO (PINTAR)

TOKEN	TIPO
VARs	Palabra Reservada
INSTRUCCIONES	Palabra Reservada
PINTAR	Palabra Reservada
int	Palabra Reservada
String	Palabra Reservada
boolean	Palabra Reservada
if	Palabra Reservada
else	Palabra Reservada
while	Palabra Reservada
true	Palabra Reservada
false	Palabra Reservada
AND	Palabra Reservada
OR	Palabra Reservada
<	Signo Menor

>	Signo Mayor
=	Signo Igual
==	Signo Igual - Igual
<=	Signo Menor Igual
>=	Signo Mayor Igual
<>	Signo Menor Mayor
+	Operador Más
-	Operador Menos
*	Operador Por
/	Operador División
,	Signo Coma
;	Signo Punto Coma
{	Signo Corchete Abrir
}	Signo Corchete Cerrar
[Signo Llave Abrir
]	Signo Llave Cerrar
(Signo Paréntesis Abrir
)	Signo Paréntesis Cerrar
(Letra “_”) → (Letra “_” Dígito)*	Expresión Identificador
(Dígito)+	Expresión Número
Comillas → (Letra Símbolo Dígito) + → Comillas	Expresión Cadena
“/*” → (Letra Símbolo Dígito) + → “*/”	Expresión Comentario Completo
“//” → (Letra Símbolo Dígito) +	Expresión Comentario de Línea

ANALIZADOR SINTÁCTICO PINTAR

G = [N, T, P, S]

Terminal (T):

Vars, Instrucciones, Pintar, True, False, Int, String, Boolean, If, Else, While, And, Or, Menor, Mayor, Igual, MenorIgual, MayorIgual, IgualIgual, MenorMayor, Punto, Numero, Mas, Menos, Por, Div, Coma, PuntoComa, CorcheteA, CorcheteC, LlaveA, LlaveC, ParentesisA, ParentesisC, String Identificador, Cadena

No Terminal (N):

inicio, declaracion_vars, estructura_instrucciones, estructura_vars, estructura_int, estructura_string, estructura_boolean, expresion_id_img_pintar, expresion_id_instr_wh, declaracion_int, declaracion_string, declaracion_boolean, expresiones_logicas, declaracion_asignaciones_if, expresiones_logicas_p_wh, expresion_int_p_wh, expresion_int, expresion_int_p, expresion_string, expresion_boolean, expresiones_logicas_p, declaracion_estructuras_if_while,

declaracion_asignaciones, declaracion_instr, declaracion_instrucciones, declaracion_estructuras,
declaracion_instrucciones_if_while, estructura_if, estructura_while, estructura_pintar,
expresion_id_instr, instr_id, expresion_id_pintar, declaracion_instrucciones_else,
declaracion_instrucciones_while, declaracion_estructuras_while, declaracion_instr_wh,
estructura_pintar_iw

S → inicio

Producciones (P):

inicio

- estructura_vars estructura_instrucciones
- expresion_boolean
- Pintar expresion_int_p
- declaracion_instr

estructura_vars

- Vars LlaveA declaracion_vars LlaveC
- Vars LlaveA LlaveC

declaracion_vars

- estructura_int
- estructura_string
- estructura_boolean
- declaracion_vars estructura_int
- declaracion_vars estructura_string
- declaracion_vars estructura_boolean

estructura_int

Int declaracion_int PuntoComa

estructura_string

String declaracion_string PuntoComa

estructura_boolean

Boolean declaracion_boolean PuntoComa

declaracion_int

- Identificador Igual expresion_int
- Identificador
- declaracion_int Coma Identificador Igual expresion_int
- declaracion_int Coma Identificador

expresion_int

- expresion_int Mas expresion_int
- expresion_int Menos expresion_int
- expresion_int Por expresion_int
- expresion_int Div expresion_int
- Numero
- Identificador

declaracion_string

- Identificador Igual expresion_string
- Identificador
- declaracion_string Coma Identificador Igual expresion_string
- declaracion_string Coma Identificador

expresion_string

- expresion_string Mas expresion_string
- Cadena
- Identificador

declaracion_boolean

- Identificador
- Identificador Igual expresion_boolean
- declaracion_boolean Coma Identificador Igual expresion_boolean
- declaracion_boolean Coma Identificador

expresion_boolean

- expresion_boolean And expresion_boolean
- expresion_boolean Or expresion_boolean
- expresiones_logicas

expresiones_logicas

- expresiones_logicas Menor expresiones_logicas
- expresiones_logicas Mayor expresiones_logicas
- expresiones_logicas Igual Igual expresiones_logicas
- expresiones_logicas Menor Igual expresiones_logicas
- expresiones_logicas Mayor Igual expresiones_logicas
- expresiones_logicas Menor Mayor expresiones_logicas
- expresion_int
- True
- False

declaracion_instr

- Identificador Igual expresion_id_instr PuntoComa

declaracion_instr_wh

- Identificador Igual expresion_id_instr_wh PuntoComa

expresion_id_instr

- expresiones_logicas_p And expresiones_logicas_p
- expresiones_logicas_p Or expresiones_logicas_p
- expresiones_logicas_p

expresiones_logicas_p

- expresiones_logicas_p Menor expresiones_logicas_p
- expresiones_logicas_p Mayor expresiones_logicas_p
- expresiones_logicas_p Igual Igual expresiones_logicas_p
- expresiones_logicas_p Menor Igual expresiones_logicas_p

- expresiones_logicas_p MayorIgual expresiones_logicas_p
- expresiones_logicas_p MenorMayor expresiones_logicas_p
- expresion_int_p
- True
- False

expresion_int_p

- expresion_int_p Mas expresion_int_p
- expresion_int_p Menos expresion_int_p
- expresion_int_p Por expresion_int_p
- expresion_int_p Div expresion_int_p
- expresion_int_p Punto Numero
- expresion_int_p Punto Identificador
- Numero
- Identificador
- Cadena

expresion_id_instr_wh

- expresiones_logicas_p_wh And expresiones_logicas_p_wh
- expresiones_logicas_p_wh Or expresiones_logicas_p_wh
- expresiones_logicas_p_wh

expresiones_logicas_p_wh

- expresiones_logicas_p_wh Menor expresiones_logicas_p_wh
- expresiones_logicas_p_wh Mayor expresiones_logicas_p_wh
- expresiones_logicas_p_wh IgualIgual expresiones_logicas_p_wh
- expresiones_logicas_p_wh MenorIgual expresiones_logicas_p_wh
- expresiones_logicas_p_wh MayorIgual expresiones_logicas_p_wh
- expresiones_logicas_p_wh MenorMayor expresiones_logicas_p_wh
- expresion_int_p_wh
- True
- False

expresion_int_p_wh

- expresion_int_p_wh Mas expresion_int_p_wh
- expresion_int_p_wh Menos expresion_int_p_wh
- expresion_int_p_wh Por expresion_int_p_wh
- expresion_int_p_wh Div expresion_int_p_wh
- expresion_int_p_wh Punto Numero
- expresion_int_p_wh Punto Identificador
- Numero
- Identificador
- Cadena

expresion_id_pintar

- expresion_int_p Mas expresion_int_p
- Identificador
- Cadena

expresion_id_img_pintar

- expresion_id_img_pintar Mas expresion_id_img_pintar
- Identificador
- Cadena

estructura_pintar

- Pintar ParentesisA expresion_id_pintar Coma expresion_id_img_pintar Coma expresion_int_p Coma expresion_int_p ParentesisC PuntoComa

estructura_pintar_iw

- Pintar ParentesisA expresion_id_pintar Coma expresion_id_img_pintar Coma expresion_int_p_wh Coma expresion_int_p_wh ParentesisC PuntoComa

estructura_while

- While ParentesisA declaracion_asignaciones ParentesisC CorcheteA declaracion_instrucciones_while CorcheteC

estructura_if

- If ParentesisA declaracion_asignaciones_if ParentesisC CorcheteA declaracion_instrucciones_if_while CorcheteC
- If ParentesisA declaracion_asignaciones_if ParentesisC CorcheteA declaracion_instrucciones_if_while CorcheteC Else CorcheteA declaracion_instrucciones_else CorcheteC

declaracion_asignaciones_if

- expresion_id_instr

declaracion_asignaciones

- expresion_id_instr_wh

declaracion_estructuras

- estructura_if
- estructura_while
- estructura_pintar
- declaracion_instr

declaracion_estructuras_if_while

- estructura_if
- estructura_while
- estructura_pintar_iw
- declaracion_instr_wh

declaracion_estructuras_while

- estructura_if
- estructura_while
- estructura_pintar_iw
- declaracion_instr_wh

declaracion_instrucciones_while

- declaracion_estructuras_while
- declaracion_instrucciones_while declaracion_estructuras_while

declaracion_instrucciones_if_while

- declaracion_estructuras_if_while
- declaracion_instrucciones_if_while declaracion_estructuras_if_while

declaracion_instrucciones_else

- declaracion_estructuras_if_while
- declaracion_instrucciones_else declaracion_estructuras_if_while

declaracion_instrucciones

- declaracion_estructuras
- declaracion_instrucciones declaracion_estructuras

estructura_instrucciones

- Instrucciones ParentesisA instr_id ParentesisC LlaveA declaracion_instrucciones LlaveC
- estructura_instrucciones Instrucciones ParentesisA instr_id ParentesisC LlaveA declaracion_instrucciones LlaveC
- Instrucciones ParentesisA instr_id ParentesisC LlaveA LlaveC
- estructura_instrucciones Instrucciones ParentesisA instr_id ParentesisC LlaveA LlaveC

Paquetes:

- proyectoanimaciones
- proyectoanimaciones.backed.analizador
- proyectoanimaciones.backed.analizador.pintar
- proyectoanimaciones.backed.manejadores
- proyectoanimaciones.backed.manejadores.gif_png
- proyectoanimaciones.backed.objetos
- proyectoanimaciones.backed.gui

Métodos:**GeneradorArchivos.java****Constantes:**

```
public static final String PARSER = "-parser";  
public static final String SYM = "sym.java";
```

Constantes para generar archivos del analizador léxico y sintáctico de las estructuras: lienzo, colores, tiempo

```
public static final String PATH_FLEX = "src/proyectoanimaciones/backed/analizador/Lexer.flex";  
public static final String PATH_CUP = "src/proyectoanimaciones/backed/analizador/Syntax.cup";
```

```

    public static final String PATH_SYM = "src/proyectoanimaciones/backed/analizador/sym.java";
    public static final String PATH_SINTAX =
"src/proyectoanimaciones/backed/analizador/Sintax.java";
    public static final String NOMBRE_SINTAX = "Sintax";
    public static final String NOMBRE_CLASE_SINTAX = "Sintax.java";

```

Constantes para generar archivos del analizador léxico y sintáctico de las estructuras: lpintar

```

    public static final String PATH_FLEX_PINTAR =
"src/proyectoanimaciones/backed/analizador/pintar/LexerPintar.flex";
    public static final String PATH_CUP_PINTAR =
"src/proyectoanimaciones/backed/analizador/pintar/SintaxPintar.cup";
    public static final String PATH_SYM_PINTAR =
"src/proyectoanimaciones/backed/analizador/pintar/sym.java";
    public static final String PATH_SINTAX_PINTAR =
"src/proyectoanimaciones/backed/analizador/pintar/SintaxPintar.java";
    public static final String NOMBRE_SINTAX_PINTAR = "SintaxPintar";
    public static final String NOMBRE_CLASE_SINTAX_PINTAR = "SintaxPintar.java";

```

Constructores:

```

    private GeneradorArchivos() {}

```

Instancia de la Clase:

```

    public static GeneradorArchivos getGeneradorArchivos()

```

public void genearar(String rutaFlex, String rutaCup, String rutaSym, String rutaSintaxJava, String nombreSintax, String nombreClase) throws Exception

Método para generar las rutas de los archivos cup y flex.

public void generador(String rutaLexicoProyecto, String[] rutaSintax, String rutaSymTXT, String nombreSym, String rutaSint, String nombreSin) throws IOException, Exception

Método para generar los archivos de cup y flex.

ProyectoAnimaciones.java

public static void main(String[] args) throws Exception

Método principal al momento de realizar la ejecución del sistema realiza una instancia de la ventana principal.

ManejadorEscritura.java

Constantes para la escritura de los archivos: lnz, clrs, tmp, pnt.

```

    public static final String LIENZOS = "LIENZOS";
    public static final String NOMBRE = "nombre";
    public static final String TIPO = "tipo";
    public static final String FONDO = "Fondo";
    public static final String RED = "Red";
    public static final String BLUE = "Blue";
    public static final String GREEN = "Green";

```

```

public static final String TAMANIO = "tamaño";
public static final String DIMENSION_X = "dimension_x";
public static final String DIMENSION_Y = "dimension_y";
public static final String CUADRO = "cuadro";
public static final String HEX = "HEX";
public static final String COLORES = "COLORES";
public static final String TIEMPOS = "TIEMPOS";
public static final String INICIO = "inicio";
public static final String FIN = "fin";
public static final String IMAGENES = "imagenes";
public static final String ID = "id";
public static final String DURACION = "duracion";
public static final String VARS = "VARS";
public static final String COMENTARIO = "Sección de Variables";
public static final String INSTRUCCIONES = "INSTRUCCIONES";
public static final String PINTAR = "PINTAR";
private static ManejadorEscritura manejadorEscritura;

```

Constructor de la clase

```
private ManejadorEscritura() {}
```

Instancia de la Clase

```
public static ManejadorEscritura getInstancia() {
```

```
    public void escribirArchivoSalida(String path, String textoSalida) throws IOException {
```

Crea el archivo, en la dirección seleccionada, mediante la clase **File**.

```
    public String archivoLienzo(List<Lienzo> lienzo)
```

Realiza la escritura del archivo lienzo, mediante la lista de lienzos que fueron creados.

```
    public String archivoColores(List<Lienzo> lienzo)
```

Realiza la escritura del archivo colores, mediante la lista de colores de cada lienzo creado.

```
    public String archivoTiempo(List<Lienzo> lienzo)
```

Realiza la escritura del archivo tiempo, mediante la lista de duración de cada imagen creada por cada lienzo creado.

```
    public String archivoPintar(List<Lienzo> lienzo)
```

Realiza la escritura del archivo pintar, mediante las instrucciones realizadas en el editor por cada lienzo modificado.

ManejadorIfWhile.java

Constructor de la Clase

```
private ManejadorIfWhile() {}
```

Instancia de la Clase

```
public static ManejadorIfWhile getInstancia()
```

public void realizarIfWhile(TablaSimbolos simbolos)

Comprueba la instrucción a realizar, entre if, else o while para posteriormente realizar cada una de las instrucciones.

private void realizarIfWhile(List<Simbolo> simbolos)

Realiza cada una de las instrucciones indicadas en cada estructura realizando cada una de las validaciones.

private int comprobarTipo(Simbolo simbolo){

Comprueba el tipo de dato de un símbolo de la tabla de simbolos.

ManejadorPestaña.java

Constructor de la clase

private ManejadorPestaña(){}

Instancia de la Clase

public static ManejadorPestaña getInstancia()

public void repintar(JLabel [][] labels, int filas, int columnas, java.awt.Color fondo)

Método para repintar el panel donde se encuentra la imagen cada vez que selecciona una nueva.

public void pintar(JLabel [][] labels, java.util.List<Pintar> pintar)

Método para pintar, un cuadrado seleccionado.

public proyectoanimaciones.backed.objetos.Color search(Lienzo lienzo, java.awt.Color color)

Método para buscar un color, mediante el color de la clase AWT, en el lienzo definido.

public Duracion searchDuracion(Lienzo lienzo, String id)

Método para buscar una imagen en el lienzo definido.

public void comprobarColores(Lienzo lienzo, JLabel[][] comparar, String id)

Método que comprueba si se realizaron modificaciones en el lienzo seleccionado.

public void addNuevaPestaña(JTabbedPane pestaña, String tituloPestaña, Lienzo lienzo, DialogEditorGrafico deg)

Método para agregar una nueva pestaña, por cada imagen que exista en el lienzo.

ManejadorExportar.java

Constructor de la clase

private ManejadorExportar() {}

Instancia de la clase

public static ManejadorExportar getInstancia()

public void pintar(Graphics g, java.util.List<Pintar> pintar, int pixels)

Método que pinta el lienzo mediante la clase Graphics, en base a la lista de pintar.

public Duracion searchDuracion(Lienzo lienzo, String id)

Método que realiza la búsqueda de la duración de una imagen en el lienzo.

private void sourcePint(int dimensionX, int dimensionY, Lienzo lienzo, AnimatedGifEncoder gifEncoder, String id)

Método para realizar una nueva instrucción de pintar una imagen,

public boolean exportarGif(Lienzo lienzo, String directorio)

Método para exportar una serie de imágenes en una sola a formato GIF.

public void expotarPNG(Lienzo lienzo, String directorio) throws IOException

Método para exportar una serie de imágenes a formato PNG.

ManejadorPintar.java

Constructor de la Clase

private ManejadorPintar() {}

Instancia de la clase

public static ManejadorPintar getManejadorPintar() {

public Lienzo buscarLienzo(String id, List<Lienzo> liezo)

Función que busca un lienzo en especifico en una lista de lienzo.

public Color buscaColores(Object id, int left, int right)

Función que realiza la búsqueda de un color en el lienzo definido.

public Duracion buscaImg(Object id, int left, int right)

Función que realiza la búsqueda de una imagen en el lienzo definido.

public String buscaImgId(Object id, Object cadena, int left, int right)

Función que realiza la búsqueda de una imagen en la tabla de símbolos.

private String sDI(String id, int left, int right)

Función que realiza la búsqueda de una imagen en la tabla de símbolos mediante el tipo.

private String sCI(String id, int left, int right)

Función que realiza la búsqueda de una imagen en la tabla de símbolos mediante el id.

public boolean comprobarSiSonMismoTipo(Object a, Object b, int aleft, int aright, int bleft, int bright, int caso)

Método que comprueba si dos variables son del mismo tipo de dato.

public Pintar comprobarPintar(Object a, Object b, Object c, Object d, int aleft, int aright, int bleft, int bright)

Método que comprueba la estructura de un objeto pintar.

public void asignarPintar(Pintar pint)

Método que realiza la asignación de un objeto pintar en el archivo lienzo.

public String comprobarTipoObjeto(Object object)

Método que realiza la comprobación del tipo de dato de un objeto.

ManejadorSintactico.java

Constructor de la Clase

private ManejadorSintactico() {}

Instancia de la clase

public static ManejadorSintactico getInstancia()

public void clear(){

Método que realiza la limpieza de la lista de símbolos y la tabla de símbolos, para poder realizar un nuevo análisis.

public void errorSyntax(int left, int right, String value, String mensaje)

Método que agrega al área de texto errores la descripción de un error sintáctico.

public void errorSemantico(int left, int right, String valor, String mensaje)

Método que agrega al área de texto errores la descripción de un error semántico.

public Object addRGB(Object objeto, int caso, int left, int right){

Método que agrega la estructura RGB en la tabla de símbolos.

public Integer addCDXY(Object objeto, int caso, int left, int right)

Método que realiza la asignación de una estructura tamaño.

public Fondo addFondo(Object objeto, int left, int right)

Método que realiza la asignación de un nuevo fondo.

public Tamaño addTamaño(Object objeto, int left, int right){

Método que realiza la asignación de un objeto tamaño.

public Object add(Object object, int left, int right, String tipo){

Método que agrega a la tabla de símbolos uno nuevo.

public Fondo comprobarRGB(Object a, Object b, Object c, int al, int ar, int bl, int br, int cl, int cr)

Método que realiza la asignación de un objeto fondo al lienzo indicado.

public Tamaño comprobarTamaño(Object a, Object b, Object c, int al, int ar, int bl, int br, int cl, int cr)

Método que realiza la asignación de un objeto tamaño al lienzo indicado.

public Lienzo addLienzo(Object a, Object b, Object c, Object d, int left, int right, String idLista, int al, int ar, int bl, int br, int cl, int cr, int dl, int dr)

Método que realiza la comprobación de un objeto lienzo

public List<Lienzo> addListLienzo(Object a, Object b, int aleft, int aright, int bleft, int bright)

Método que realiza la asignación de un nuevo lienzo ala lista.

public Color comprobarColor(Object id, Fondo color, int cleft, int cright, int aleft, int aright){

Método que realiza la comprobación de la estructura de un fondo.

public List<Color> listaColores(Object a, Color b, int aleft, int aright, int bleft, int bright){

Método que realiza la asignación de un objeto color al lienzo seleccionado.

public void comprobarAddColores(Object id, Object colores, int ileft, int irect){

Método que realiza la comprobación de la existencia de un color en lienzo.

public List<Duracion> comprobarTimeImg(Object e, int left, int right){

Método que realiza la comprobación de un objeto duración.

public Tiempo comprobarTiempo(Object a, Object b, Object c, int al, int ar, int bl, int br, int cl, int cr){

Método que realiza la comprobación de la estructura de un objeto tiempo.

public void comprobarDuracion(Object id, Tiempo tiempo, int left, int right){

Método que comprueba la existencia de una imagen el lienzo.

public Object realizarOperaciones(Object a, Object b, int aleft, int aright, int bleft, int bright, int caso){

Método que realiza operaciones.

private void asignarInt(String id, Object valor, int left, int right){

Método que realiza la asignación de un dato.

public Object buscarIdEnTabla(String id, int left, int right){

Método que realiza la búsqueda de un lienzo en la tabla.

public Object realizarOperacionesLogicas(Object a, Object b, int aleft, int aright, int bleft, int bright, int caso) {

Método para realizar operaciones lógicas.

public Object realizarOperacionesAndOr(Object a, Object b, int aleft, int aright, int bleft, int bright, int caso) {

Método para realizar operaciones lógicas AND y OR.

public void cargarTablaCorrespondiente(String id, int left, int right)

Método que busca un lienzo en la tabla de símbolos.

public Object buscarSimbolo(String id, String tipo, int left, int right, boolean soloTabla, boolean idSearch, boolean mostrarError)

Método que realiza la búsqueda de un símbolo.

public Object addValorIfWhile(Object add, int left, int right)

Método para realizar agrega una instrucción a la estructura de If o While.

public void agregarNuevaEstructura(boolean ifs, Object booleanoCodicional){

Método que agrega una nueva instrucción If o While a la tabla de simbolos.

DialogEditorGrafico.java

Constructor de la ventana

public DialogEditorGrafico(java.awt.Frame parent, boolean modal, String pathLienzo, String pathColores, String pathTiempo, String pathPintar, List<Lienzo> lista)

private void menuGuardarActionPerformed(java.awt.event.ActionEvent evt)

Método para guardar los cambios realizados en la interfaz grafica.

private void menuAcercaDeActionPerformed(java.awt.event.ActionEvent evt)

Método para mostrar los datos del desarrollador.

private void addPestañas()

Método para agregar una nueva pestaña al panel.

Principal.java

Constructor de la ventana

public Principal()

private void menuGuardarActionPerformed(java.awt.event.ActionEvent evt)

Método para guardar los cambios realizados por cada uno de los archivos en el editor de texto.

private void realizarAnalisisLCT(String comprobacion)

Método para realizar el análisis sintáctico y léxico de un archivo.

private void menuAcercaDeActionPerformed(java.awt.event.ActionEvent evt)

Método que muestra la información del desarrollador.

private void menuExportarActionPerformed(java.awt.event.ActionEvent evt)

Método para exportar los lienzo al formato indicado.

Objetos:

- Blue
- Color
- Dimension
- Duración
- Fondo
- Green
- Int
- Lienzo
- Pintar
- Red
- Simbolo
- TablaSimbolos
- Tamaño
- Tiempo

NOTA: Cada uno con sus respectivo constructor, getter's y setter's.

Analizador Léxico – Sintáctico

- Lexer.flex
- Lexico.java
- Sintax.cup
- Sintax.java
- LexerPintar.flex
- LexicoPintar.java
- SintaxPintar.cup
- SintaxPintar.java
- sym.java

Librerías:

- Jflex.jar
- java-cup-11b-runtime.jar
- java-cup-11b.jar

- Diseño Absoluto – AbsoluteLayout.jar

- AnimatedGifEncoder.java
- GifDecoder.java
- LZWEncoder.java
- NeuQuant.java

REQUISITOS MINIMOS PARA PROGRAMAR EN JAVA: REQUERIMIENTOS DE HARDWARE PARA PROGRAMAR EN JAVA

Para las aplicaciones generadas se debe tener un mínimo de 32MB de RAM, se recomienda que se tengan 48MB o más. Para compilar utilizando el SDK de Microsoft es un mínimo de 32MB de RAM, y para JDK 48MB mínimo. El procesador en principio no es tan crítico como la memoria RAM, pero se recomienda utilizar al menos un Pentium de 133 para compilar/ejecutar las aplicaciones.

Requerimientos de SOFTWARE

Para trabajar con el Generador Java es necesario cierto software adicional para la compilación y ejecución de las aplicaciones generadas.

Java Development Kit - Compilador y Máquina Virtual

Para la compilación y ejecución de los programas es necesario tener el JDK. En cuanto a las versiones se recomienda en general utilizar las últimas liberadas de cada uno de ellos. En caso de que alguna versión sea menos estable que su versión anterior, se avisara.

Se requieren las versiones siguientes de .NET Framework:

SQL Server 2008 en Windows Server 2003 64 bits IA64: .NET Framework 2.0 SP2
SQL Server Express: .NET Framework 2.0 SP2. Todas las demás ediciones de SQL Server 2008: .NET Framework 3.5 SP1, etc.

IDE NetBeans 8.2 requisitos mínimos:

Microsoft Windows Vista SP1 / Windows 7 Professional:

Procesador: Varel Pentium III 800MHz o equivalente
Memoria: 512 MB
Espacio en disco: 750 MB de espacio libre en disco

Ubuntu 9.10:

Procesador: Varel Pentium III 800MHz o equivalente
Memoria: 512 MB
Espacio en disco: 650 MB de espacio libre en disco

Macvarosh OS X 10.7

Varel: Procesador: Varel
Dual-Core Memoria: 2 GB
Espacio en disco: 650 MB de espacio libre en disco

Microsoft Windows 7 Professional / Windows 8 / Windows 8.2 / Windows 10:

Procesador: Varel Core i5 o equivalente
Memoria: 2 GB 32 bits, 4

GB 64 bits

Espacio en disco: 1,5 GB de espacio libre en disco

Ubuntu 15.04 y Superior:

Procesador: Varel Core i5 o

equivalente Memoria: 2 GB 32 bits, 4

GB 64 bits

Espacio en disco: 1,5 GB de espacio libre en disco

OS X 10.10 Varel:

Procesador: Varel Dual-

Core Memoria: 4 GB

Espacio en disco: 1,5 GB de espacio libre en disco