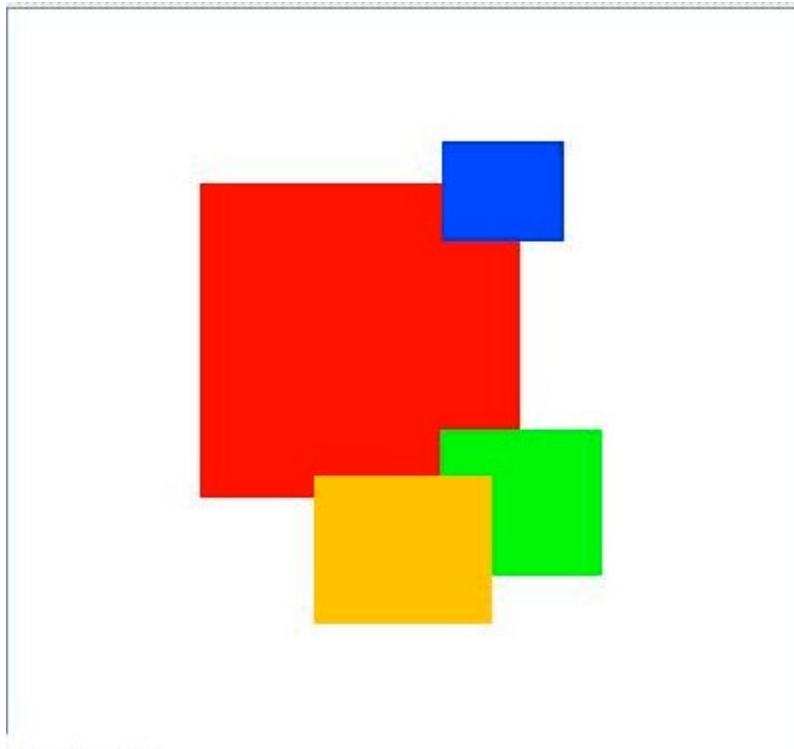


MANUAL TÉCNICO

Software de dibujo de imágenes



Jhonny García Hernández

Versión 1.0

OBJETIVOS

- General

Brindar la información necesaria para poder realizar la instalación y configuración del aplicativo.

- Específicos

1. Representar la a funcionalidad técnica de la estructura, diseño y definición del aplicativo.
2. Definir claramente el procedimiento de instalación del aplicativo.
3. Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
4. Describir las herramientas utilizadas para el diseño y desarrollado del prototipo.

INTRODUCCIÓN

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación del aplicativo creado para el diseño de animaciones (gif) o imágenes con extensión png.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación del aplicativo.

REQUISITOS

Requisitos para el IDE (Opcional):

Configuraciones de Hardware Mínimas

- Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:
- Procesador: Intel Pentium III o equivalente a 800 MHz
- Memoria: 512 MB
- Espacio en disco: 750 MB de espacio libre en el disco
- Ubuntu 9.10:
- Procesador: Intel Pentium III o equivalente a 800 MHz
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco
- Solaris OS versión 10 (SPARC):
- Procesador: UltraSPARC II a 450 MHz
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco
- Solaris OS versión 10 (x86/x64 Platform Edition):
- Procesador: AMD Opteron serie 1200 a 1,8 GHz
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco
- Macintosh OS X 10.5 Intel:
- Procesador: Intel Dual-Core (32 o 64 bits)
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco

Configuraciones de Hardware Recomendadas

- Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:

- Procesador: Intel Pentium IV o equivalente a 2,6 GHz

- Memoria: 2 GB

- Espacio en disco: 1 GB de espacio libre en el disco

- Ubuntu 9.10:

- Procesador: Intel Pentium IV o equivalente a 2,6 GHz

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

- Solaris OS versión 10 (SPARC):

- Procesador: UltraSPARC IIIi a 1 GHz

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

- Solaris OS versión 10 (x86/x64 Platform Edition):

- Procesador: AMD Opteron serie 1200 a 2,8 GHz

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

- OpenSolaris 2010.03 (x86/x64 platform edition):

- Procesador: AMD Opteron serie 1200 a 2,8 GHz

- Memoria: 2 GB

- Espacio en disco: 650 MB de espacio libre en el disco

- Macintosh OS X 10.6 Intel:

- Procesador: Intel Dual-Core (32 o 64 bits)

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

Configuraciones de Hardware Mínimas (Compilador)

Pentium III o equivalente, 1 ghz o superior

- 512 MB RAM, se recomienda 1 GB o más
- 1,5 GB NTFS o 3 GB FAT o más de espacio disponible en disco
- Windows 2000 SP4 o superior

INSTALARCIO (IDE)

Para instalar el software:

Microsoft Windows, Solaris OS y Linux

1. Una vez finalizada la descarga, ejecute el instalador.
 - El archivo del instalador de Windows tiene la extensión .exe. Haga doble clic en él para ejecutarlo.
 - El archivo del instalador de las plataformas Solaris y Linux tiene la extensión .sh. En estas plataformas debe convertir los archivos del instalador en ejecutables mediante el siguiente comando: `chmod +x <nombre-archivo-instalador>`
2. Para seleccionar las herramientas y tiempos de ejecución que se van a instalar, siga estos pasos en la página de bienvenida del asistente de instalación:
 - Haga clic en Personalizar.
 - En el cuadro de diálogo Personalizar la instalación, realice las selecciones.
 - Haga clic en Aceptar.
3. En la página de bienvenida del asistente para la instalación, haga clic en Siguiente.
4. En la página del contrato de licencia, revise el contrato, marque la casilla de aceptación y haga clic en Siguiente.

5. En la página de instalación de NetBeans IDE, siga estos pasos:

- Acepte el directorio de instalación predeterminado de NetBeans IDE o especifique otro directorio. Nota: el directorio de instalación debe estar vacío y el perfil de usuario que utilice para ejecutar el instalador debe disponer de permisos de lectura/escritura en dicho directorio.

6. Haga clic en Instalar para comenzar la instalación.

Nota: si surgen problemas para instalar el software correctamente, consulte Resolución de problemas para obtener descripciones y sugerencias para solucionar problemas no resueltos que pueden afectar al proceso de instalación.

Mac OS X

Para instalar el software:

- Una vez finalizada la descarga, ejecute el instalador. La extensión del archivo del instalador es .dmg.
- En los paneles que se abren, haga clic en el icono del paquete. La extensión del paquete es .mpkg. El asistente de instalación comienza.
- En la página de bienvenida del asistente para la instalación, haga clic en Continuar.
- Revise el contrato de licencia y haga clic en Continuar. Haga clic en Aceptar en la ventana emergente para aceptar la licencia.
- En la página de selección de destino, elija la unidad y haga clic en Continuar.
- Si ha descargado los paquetes de "All" (Todo), Java o Ruby, puede personalizar la instalación. En el último panel del asistente de instalación, pulse el botón Personalizar de la parte inferior izquierda del panel. Aparece el árbol de productos.
- Seleccione los productos que desea instalar.
- Especifique el nombre y la contraseña del administrador del sistema y haga clic en Aceptar para comenzar la instalación.

INSTALACIÓN (JDK11)

Para la instalación de la maquina virtual, para este aplicativo se uso el JDK11 por lo que se recomienda, dependiendo de su sistema operativo puede descargarlo en el siguiente link:

<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

Asi mismo al momento de descargarlo, si su computador tiene un software de instalación como es windows, o linux para archivos .deb , puede proceder a instalarlo de manera facil, de lo contrario leer la documentación en el archivo descargado.

CONFIGURACIÓN DEL IDE PARA CREAR LAS GRAMÁTICAS

JLEX (Análisis léxico):

JLex es un generador de analizadores léxicos, escrito en Java, para Java. JLex fue desarrollado por Elliot Berk en la Universidad de Princeton. Para más información visitar la página oficial de JLex:

<https://www.cs.princeton.edu/~appel/modern/java/JLex/>

La principal tarea de un analizador léxico es leer los caracteres de entrada del programa fuente, agruparlos en lexemas y producir como salida una secuencia de tokens.

- Un token es un par que consiste en un nombre de token y un valor de atributo opcional.
- Un lexema es una secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token y que el analizador léxico identifica como una instancia de este token.
- Un patrón es una descripción de la forma que pueden tomar los lexemas de un token.

En Jlex se definen los patrones de los diferentes tokens que se desean reconocer, estos patrones pueden definirse a través de expresiones regulares. Además Jlex cuenta con múltiples opciones, una muy importante es su capacidad para integrarse con generadores de analizadores sintácticos como Cup

Cup:

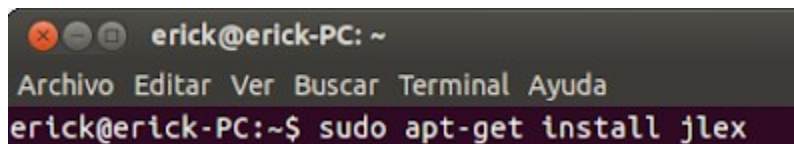
Cup es un generador de analizadores sintácticos de tipo LALR para Java. Para más información visitar la página oficial de Cup:

<http://www2.cs.tum.edu/projects/cup/>

El analizador sintáctico obtiene una cadena de tokens del analizador léxico y verifica que dicha cadena pueda generarse con la gramática para el lenguaje fuente. Una gramática proporciona una especificación precisa y fácil de entender de un lenguaje de programación.

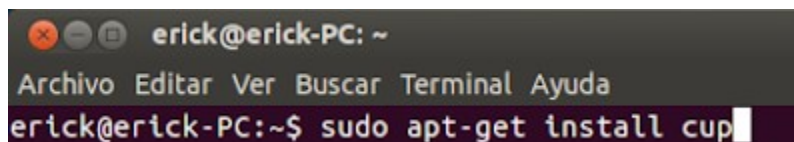
Instalación de jlex y cup:

Lo primero que haremos será instalar JLex, para ello abrimos una terminal, en Ubuntu puede hacerse con la combinación de teclas Ctrl+Alt+t o en Aplicaciones → Accesorios → Terminal, una vez abierta la terminal ingresamos el comando “sudo apt-get install jlex”, autenticamos ingresando nuestra contraseña y aceptamos la descarga e instalación, con esto quedará instalado JLex.

A screenshot of a terminal window titled 'erick@erick-PC: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command prompt shows 'erick@erick-PC:~\$' followed by the command 'sudo apt-get install jlex' which is currently being typed.

```
erick@erick-PC:~$ sudo apt-get install jlex
```

Luego instalamos cup, ejecutando en la terminal el comando “sudo apt-get install cup”, autenticamos ingresando nuestra contraseña y aceptamos la descarga e instalación, con esto quedará instalado Cup.

A screenshot of a terminal window titled 'erick@erick-PC: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command prompt shows 'erick@erick-PC:~\$' followed by the command 'sudo apt-get install cup' which is currently being typed.

```
erick@erick-PC:~$ sudo apt-get install cup
```

Para mas información sobre la creacion de archivos jflex e incorporar cup, asi mismo como ejecutar estos componentes para realizar el analisis de una entrada:

<http://e-navarro.blogspot.com/2015/09/mi-primer-proyecto-utilizando-jlex-y-cup.html>
<https://openfecks.wordpress.com/jlex-y-cup/>

ANÁLISIS DE REQUERIMIENTOS:

- Crear archivos con las extensiones:
 1. pnt
 2. clr
 3. lz
 4. tmp
- Realizar un análisis sintáctico, semántico y léxico de cada archivo.
- Gestión de archivos, para abrir un archivo por cada tipo.
- Abrir archivos, solo con las extensiones mencionadas anteriormente.
- Guardar un archivo abierto.
- Realizar un editor de texto para los archivos, (CRUD)
- Realizar un editor grafico, si los archivos están correctamente estructurado.
- Pintar sobre un plano en 2D en base a los datos recavados de los archivos.
- Tener un orden de las imágenes.
- Exportar lienzos a formato Gif
- Exportar lienzos a formato Png
- Exportar lienzo en base a su descripción.

CASOS DE USO ALTO NIVEL:

Numero	CU01
Caso de Uso:	Crear nuevo archivo
Actores:	Usuario
Descripción:	Un usuario, estando en el aplicativo se dirige a la barra de menú superior, en el menú archivo existe un Item, crear nuevo archivo, donde al momento de darle click escoge la configuración del archivo para poder visualizarlo.
Tipo:	Primario

Numero	CU02
Caso de Uso:	Realizar análisis sintáctico, léxico y semántico
Actores:	Usuario
Descripción:	Un usuario, cuando un archivo esta abierto, en la barra de menú, existe un menú de analizar que permite analizar cada tipo de archivo en base a su extensión.
Tipo:	Primario

Numero	CU03
Caso de Uso:	Abrir un archivo
Actores:	Usuario
Descripción:	Un usuario, cuando un archivo esta abierto, en la barra de menú de archivo, o con la combinación de teclas CTRL+A, desplegará un buscador de archivos, visualizando solo aquellos archivos que cumplan con la extensión antes mencionada, de esa manera puede abrir un archivo previamente creado.
Tipo:	Primario

Numero	CU04
Caso de Uso:	Guardar un archivo
Actores:	Usuario
Descripción:	Un usuario, cuando un archivo esta abierto, en la barra de menú de archivo en guardar, o con la combinación de teclas CTRL+G, se guardara el documento que se esta seleccionado.
Tipo:	Primario

Numero	CU05
Caso de Uso:	Editar gráficamente
Actores:	Usuario
Descripción:	Un usuario, cuando un archivo esta abierto, en la barra de menú de analizar en analizar todos, o con la combinación de teclas CTRL+T, analizara todos los archivos, si esta correctamente desplega el editor grafico.
Tipo:	Primario

Numero	CU06
Caso de Uso:	Pintar en un lienzo
Actores:	Usuario
Descripción:	Luego de analizar los archivos, se muestra un editor grafico donde puede pintar mediante acciones, un listado de colores, lienzos e imágenes de cada lienzo con su determinado tiempo.
Tipo:	Primario

Numero	CU07
Caso de Uso:	Ordenar imágenes
Actores:	Usuario
Descripción:	En el editor grafico, un botón de ordenar, permite establecer la secuencia de las imágenes que se quiere.
Tipo:	Primario

Numero	CU08
Caso de Uso:	Exportar Gif
Actores:	Usuario
Descripción:	En la barra de menú, en la opción de generar en exportar Gif, por cada lienzo los exporta en formato Gif.
Tipo:	Primario

Numero	CU09
Caso de Uso:	Exportar Png
Actores:	Usuario
Descripción:	En la barra de menú, en la opción de generar en exportar Png, por cada lienzo los exporta en formato Png.
Tipo:	Primario

Numero	CU010
Caso de Uso:	Exportar este lienzo
Actores:	Usuario
Descripción:	Cuando esta posicionado en un lienzo, en un botón de exportar este lienzo, exporta solo el lienzo donde esta posicionado, acá se toma el parámetro que se determino en su archivo de configuración, ya sea Png o Gif.
Tipo:	Primario

GRAMATICA PARA ARCHIVO .PNT

Terminales:

Vars	Punto
abreCor	Inst
cierraCor	If
Int	Else
Id	While
String	pintar
Boolean	dobleEquals
puntoComa	And
coma	Or
equals	Res
Sum	Mul
Cadena	Div
Operador	Num
Bool	abreParen
COM	cierraParen
Error	Double
abreLlaves	
cierraLlaves	

No terminales:

A	INSTRUC	STR3
VARIABLES	INSTRUCCIONES	COND2
VALUES	ASIGNACION	CONDICION2
OPCBOOL	ASIG	COND22
OPCSTR	SENTENCIAIF	COND23
OPCINT	SENTENCIAWHILE	CONDICIONAL
K	PAINT	Datas
E	PARAM	ASIGNACION2
T	PARAM2	ASIG2
OP2	SYMBOLS	CONDICIONAL2
OP1	STR2	COND24
F	C	COND233
STR	U	
STR1	X	
CADENA	V	
CONDICION	STRINGX	
COND	OperadorLogico	
COND1	AS	

INICIO {AS}

AS::= A INSTRUCCIONES T22
 SYMBOLS::= Sum | Res | Mul | Div | abreParen | cierraParen;
 OperadorLogico::= And | Or ;
 T22::= INSTRUCCIONES T22 | ;

A::=VARIABLES K 1
 VARIABLES::= Vars abreCor VALUES
 VALUES::= INT OPCIONINT1
 | BOOLEAN OPCIONBOOL
 | STRINGX OPCIONSTRING
 ;

INT::=Int Id OPCINT

BOOLEAN::= Boolean Id OPCBOOL

OPCBOOL::= equals E22

| ;

OPCIONBOOL::= OPCIONBOOL2 OPCIONBOOL

| puntoComa

OPCIONBOOL2::= coma Id

| coma Id equals E22

STRINGX::= String Id OPCSTR

OPCSTR::= equals S1

| ;

OPCIONSTRING::= OPCIONSTRING2 OPCIONSTRING

| puntoComa

OPCIONSTRING2::= coma Id

| coma Id equals S1

OPCINT::= equals E |

OPCIONINT1::= OPCIONINT2 OPCIONINT1

;}

| puntoComa

OPCIONINT2::= coma Id

| coma Id equals E

K::= cierraCor | VALUES K

|error K ;

E::=T OP1 E

| T
T::= F OP2 T
| F

F::= Num
| abreParen E cierraParen
| Id

OP2::= Mul
| Div
OP1::= Sum
| Res

/*
*Condicion string
*/

CADENA::= STRING CADENA
| ;
STRING::= Cadena | Id | Num | SYMBOLS
E22::= C S11
| CONDICION X22

X22::= Operador S1 S11
| S11

S11::= OperadorLogico E22
| ;

CONDICION::= V22 OP1 S1
| V22
;

V22::= DATA22 OP2 S2
| DATA22 { :RESULT=e;: }

| abreParen COND1 X22 cierraParen

COND1::= V22 OP1 S1
| V22 { :RESULT=e; }

C::= abreParen C S11 cierraParen
| Bool

DATA22::=
Num
| Id
| COM CADENA COM

S1::= S2 OP1 S1
| S2
S2::= S3 OP2 S2
| S3
S3::= Num

| abreParen S1 cierraParen

| Id
| COM CADENA COM

INSTRUCCIONES::= Inst abreParen Id cierraParen abreCor INSTRUC ASIG

INSTRUC::= ASIGNACION | PAINT | SENTENCIAIF
| SENTENCIAWHILE

ASIGNACION::= Id equals E22 puntoComa

ASIG::= cierraCor
| INSTRUC ASIG

SENTENCIAWHILE::= While abreParen E22 cierraParen V

PARAM2::= E
| E Punto E

V::= abreLlaves Datas cierraLlaves

Datas::= INSTRUC S22
S22::= INSTRUC S22

C22::= Else V

TOKENS

Token	Lexema	Patrón
Or	OR	Palabra reservada
abreParen	{	Palabra reservada
Sum	+	Palabra reservada
If	if	Palabra reservada
Id	ID	
Operador	> < == != <> > = <=	Palabra reservada
coma	,	Palabra reservada
Else	else	Palabra reservada
Res	-	Palabra reservada
Punto	..	Palabra reservada
While	while	Palabra reservada
Int	int	Palabra reservada
Cadena	Cadena	Sucesión de caracteres, iniciada por letra o guion bajo, seguido de mas letras o números mas simbolos
Inst	INSTRUCCIONES	Palabra reservada
EOF	landa	Palabra reservada
And	AND	Palabra reservada
cierraLlaves	}	Palabra reservada
Div	/	Palabra reservada
Num	Numero	Numero seguido de cero o mas numeros
COM	“\”	Palabra reservada
Vars	VARS	Palabra reservada
puntoComa	;	Palabra reservada
abreLlaves	{	Palabra reservada
String	String	Palabra reservada

Mul	*	Palabra reservada
Bool	true false	Palabra reservada
Boolean	boolean	Palabra reservada
abreCor	[Palabra reservada
cierraCor]	Palabra reservada
pintar	pintar	Palabra reservada
cierraParen)	Palabra reservada

GRAMATICA PARA COLORES:

TERMINAL:

abreLlaves
cierraLlaves
dosPuntos
Colores
coma
Rojo
Azul
Verde
Hex
Letter
Double
Numero
Id
Error

NO TERMINAL: A,DATOS,F,H,J,COLORS,K,DATA;

A
DATOS
F
H
J
COLORS
K
DATA

INICIO= A;

A::= abreLlaves Colores dosPuntos abreLlaves DATOS F cierraLlaves cierraLlaves

DATOS::= Id dosPuntos abreLlaves H J cierraLlaves

F::= coma DATOS | |error coma DATOS;

H::= Id dosPuntos abreLlaves COLORS cierraLlaves

J::= coma H | |error coma H;

COLORS::= K dosPuntos Numero 1 coma K dosPuntos Numero 2 coma K

| Hex dosPuntos Letter

K::= Rojo| Verde | Azul

TOKENS

Token	Lexema	Patrón
Rojo	Rojo	Palabra reservada
Azul	Azul	Palabra reservada
coma	,	Palabra reservada
cierraLlaves	}	Palabra reservada
Id	Id	Suscecion de caracteres, iniciada por letra o guion bajo, seguido de mas letras o numeros
Hex	HEX	Palabra reservada
Letter	Color Hexadecimal	Token inicia con # seguido de un numero hexadecimal de 6 parámetros
EOF	Landa	Palabra reservada
abreLlaves	{	Palabra reservada
Verde	Verde	Palabra reservada
Colores	Colores	Palabra reservada
Numero	Numero	Numero seguido de cero o mas numeros
dosPuntos	:	Palabra reservada

GRAMATICA PARA TIEMPOS:

TERMINAL:

abreLlaves	Image
cierraLlaves	Iden
dosPuntos	COM
Time	Com
Start	abreCor
End	cierraCor
Numero	coma
Id	Duracion
Error	

NO TERMINAL:

A
DATOS
F
C
V
K
W
Z
X
J

start with A;

A::= abreLlaves Time dosPuntos abreLlaves DATOS F cierraLlaves cierraLlaves

DATOS::= Id dosPuntos abreLlaves C cierraLlaves

F::= coma DATOS | |error coma DATOS;

C::= V1 coma V coma V3

V::= Start dosPuntos COM Id COM

;

K::= W1 coma Z

| Z coma W1

W::=Iden dosPuntos COM Id COM

Z::=Duracion dosPuntos Numero

X::=abreLlaves K cierraLlaves

J::= coma X J | cierraCor;

TOKENS

Token	Lexema	Patrón
cierraLlaves	{	Palabra reservada
Numero	Numero	Numero seguido de cero o mas números
coma	,	Palabra reservada
Iden	Id	Sucesión de caracteres, iniciada por letra o guion bajo, seguido de mas letras o números
COM	“	Palabra reservada
Start	inicio	Palabra reservada
End	fin	Palabra reservada
Duracion	duración	Palabra reservada
abreCor	[Palabra reservada
Com	“ \”	Palabra reservada
Time	tiempo	Palabra reservada
EOF	landa	Palabra reservada (cadena vacia)
cierraCor]	Palabra reservada
abreLlaves	{	Palabra reservada
Id	ID	Sucesión de caracteres, iniciada por letra o guion bajo, seguido de mas letras o números
Image	imágenes	Palabra reservada
dosPuntos	:	Palabra reservada

GRAMATICA PARA LIENZOS:

A::= abreLlaves Lienzos dosPuntos abreLlaves DATOS F cierraLlaves cierraLlaves

DATOS::= Id dosPuntos abreLlaves C cierraLlaves

F::=coma DATOS

| ;

C::= V coma V coma V coma V

COLORS::= K dosPuntos Numero coma K dosPuntos Numero coma K dosPuntos
Numero

| Hex dosPuntos Letter

D3::= Size dosPuntos abreLlaves Y cierraLlaves

Y::= SIZE coma SIZE coma SIZE

SIZE::= POSX

|POSY

|Cuadro dosPuntos Numero

;

POSX::=Posx dosPuntos Numero

POSY::= Posy dosPuntos Numero

K::= Rojo | Verde | Azul ;

STRING::= Cadena STRING| ;

TOKENS

Token	Lexema	Patrón
cierraLlaves	{	Palabra reservada
Verde	Green	Palabra reservada
Numero	Numero [0-9]	Numero seguido de cero o mas números
Fond	Fondo	Palabra reservada
type	tipo	Palabra reservada
coma	,	Palabra reservada
COM	“, \”	Palabra reservada
Rojo	Red	Palabra reservada
Cadena	Cadena	Sucesión de caracteres, iniciada por letra o guion bajo, seguido de mas letras o números mas simbolos
EOF	Landa	Palabra reservada
Hex	HEX	Palabra reservada
Azul	Blue	Palabra reservada
Id	Id	Sucesión de caracteres, iniciada por letra o guion bajo, seguido de mas letras o números
abreLlaves	{	Palabra reservada
Size	tamaño	Palabra reservada
Image	png O gif	Palabra reservada
Letter	Numero hexadecimal de 6 numeros	Sucesión de # mas 6 números en hexadecimal
dosPuntos	:	Palabra reservada
Posy	dimension_y	Palabra reservada
Posx	dimension_x	Palabra reservada
Cuadro	cuadro	Palabra reservada
Lienzos	LIENZOS	Palabra reservada
Nombre	nombre	Palabra reservada

DIAGRAMA DE CLASES

