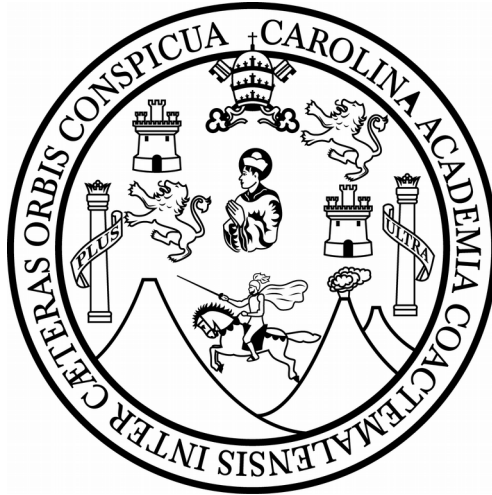


CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA CIENCIAS Y SISTEMAS



LABORATORIO DE ORGANIZACIÓN LENGUAJES Y COMPILADORES 1

“QUINTO SEMESTRE”

ING.: JOSÉ MOISÉS GRANADOS

AUX: EDVIN TEODORO GONZALEZ RAFAEL

ESTUDIANTE: Bryan René Gómez Gómez – 201730919

PROYECTO: “Manual de Usuario - PROYECTO FINAL - ANIMATIONS”

FECHA: 27 de Mayo de 2,020

ANIMATIONS – GUATEMALA

Es una software dedicado a crear animaciones con gráficos en 8 bits.

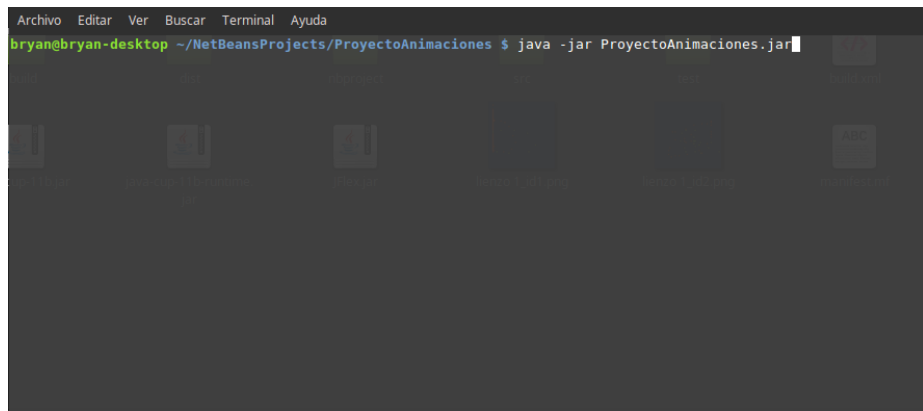
Debido a que el usuario puede editar las animaciones creadas, posee un traductor que lea las instrucciones de un gráfico existente y por medio de una interfaz de usuario se generan modificaciones.

Para poder lograr una animación, el lienzo definido se tendrá que actualizar una cierta cantidad de veces y cada vez que se actualice tendrá que durar un determinado tiempo hasta volverse a actualizar nuevamente, esto deberá especificarse en el archivo tiempo, gracias a esto podremos pintar el lienzo de diferente manera en cada tiempo definido generando de esta manera la animación deseada.

Para iniciar la aplicación debes de tener instalado JDK 8:

Debes de abrir la terminal en donde se encuentra el .jar ejecutable.

Debes de ingresar el siguiente comando: **\$java -jar ProyectoAnimaciones.jar**



Descripción de los archivos (.lnz, .clrs, .tmp, .pnt)

Archivo Lienzos: Este archivo contiene toda la información de los lienzos, sus dimensiones, el formato de la animación, el tamaño, etc. Este es el primer archivo que debe de analizarse para tener la información de los lienzos para que los siguientes archivos puedan utilizar esa información. Extensión “lnz”.

Archivo Colores: Este archivo contiene toda la información acerca de los colores que corresponden a cada lienzo para poder usarlos en la animación. Este es el segundo archivo que debe de analizarse para tener la información de los lienzos. Extensión “clrs”.

Archivo Tiempo: Este archivo contiene toda la información acerca de la duración de la animación como también el tiempo que debe tardar cada imagen del lienzo en actualizarse. Extensión “tmp”.

Archivo Pintar: Este archivo contiene la información de cómo se colorea cada lienzo, indicando la imagen y coordenadas para realizar el dibujo correspondiente. Este es el ultimo archivo que debe analizarse para crear la animación. Extensión “pnt”.

NOTA: Cada lienzo tiene asociada una animación, y cada lienzo es independiente de otros.

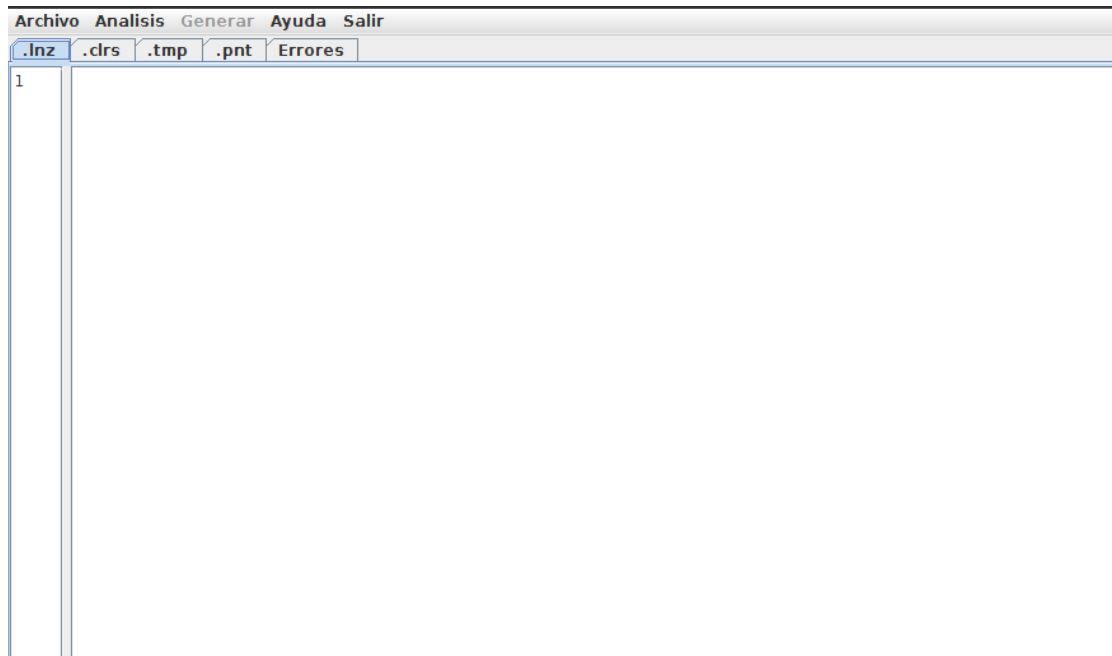
CARACTERÍSTICAS:

La interfaz se divide en dos:

- Editor de texto que facilite la creación o carga de archivos de entrada para la generación de las imágenes.
- Editor gráfico que permite la edición gráfica de las imágenes.

Editor de texto

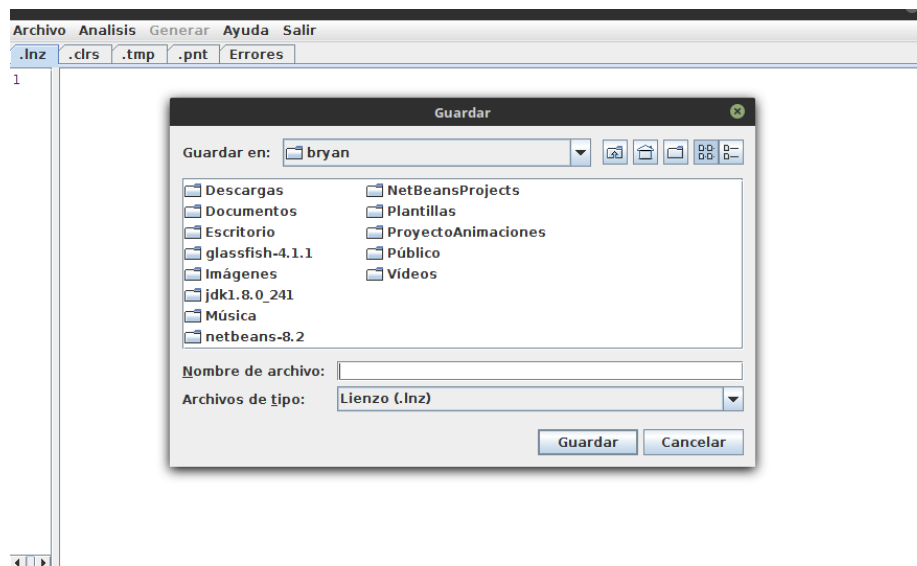
En esa interfaz se pueden abrir los cuatro tipos de archivos que se usan para generar imágenes y se pueden editar para su posterior análisis.



Menús:

Nuevo (Archivo):

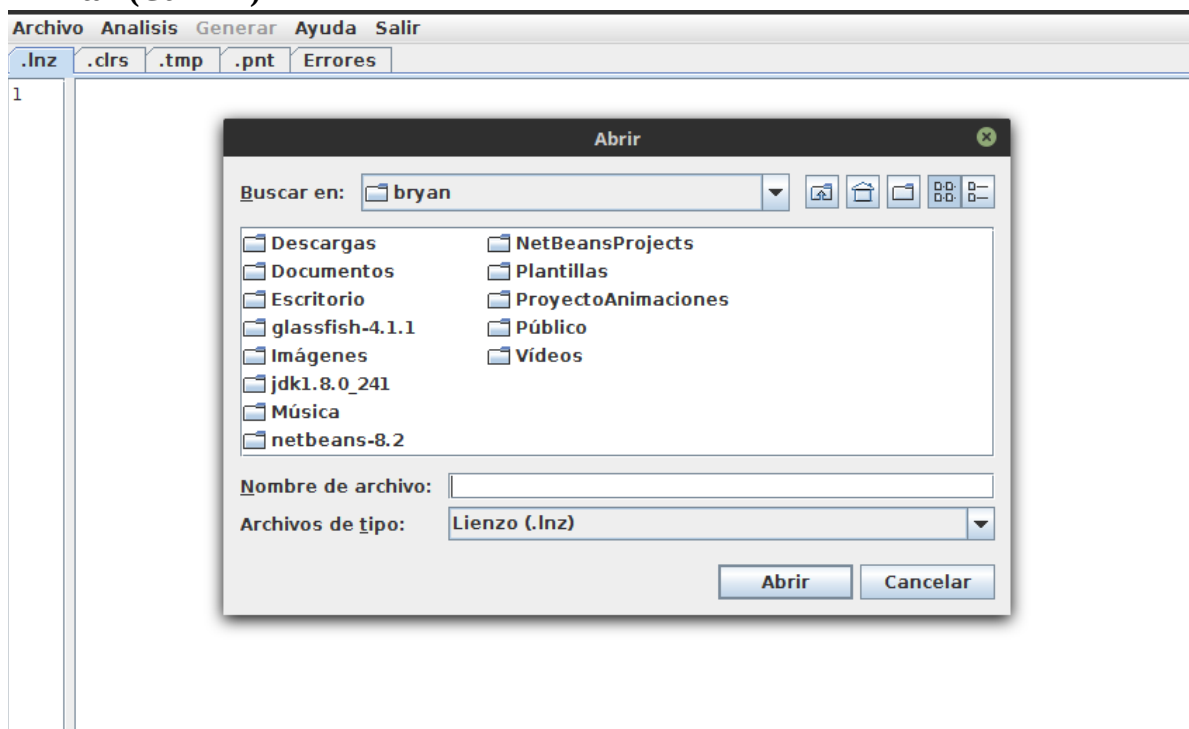
- **Lienzos (Alt + L):** Crea un nuevo archivo con extensión “lnz” y sustituye cualquier archivo abierto en la pestaña de lienzos.
- **Colores (Alt + C):** Crea un nuevo archivo con extensión “clrs” y sustituye cualquier archivo abierto en la pestaña de colores.
- **Tiempo (Alt + T):** Crea un nuevo archivo con extensión “tmp” y sustituye cualquier archivo abierto en la pestaña de tiempo.
- **Pintar (Alt + P):** Crea un nuevo archivo con extensión “pnt” y sustituye cualquier archivo abierto en la pestaña de pintar.



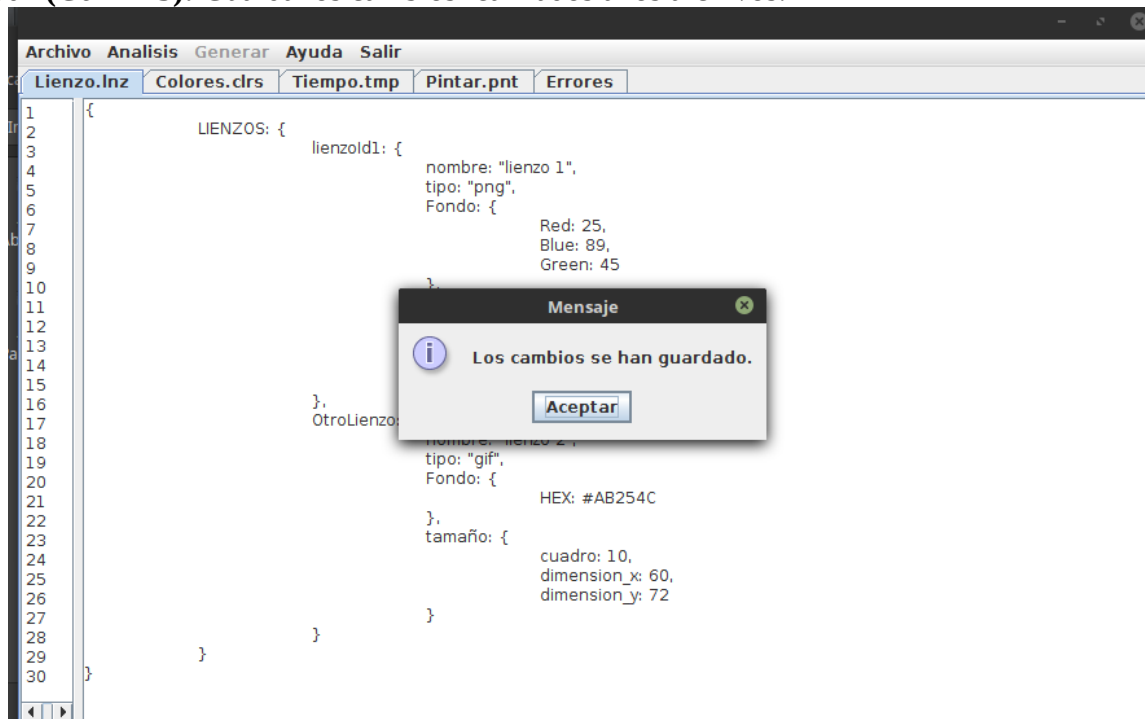
Abrir: abre algún archivo que se use en el programa. El cuadro de diálogo de búsqueda del archivo solo debe muestra archivos con las extensiones del programa (**.lnz**, **.clrs**, **.tmp**, **.pnt**) y se debe sustituir cualquier archivo abierto con anterioridad. Eso para garantizar que solo un archivo de cada tipo esté abierto en el sistema en cualquier momento.

Combinaciones de teclas:

- **Lienzo (Ctrl + L)**
- **Colores (Ctrl + Q)**
- **Tiempo (Ctrl + T)**
- **Pintar (Ctrl + P)**



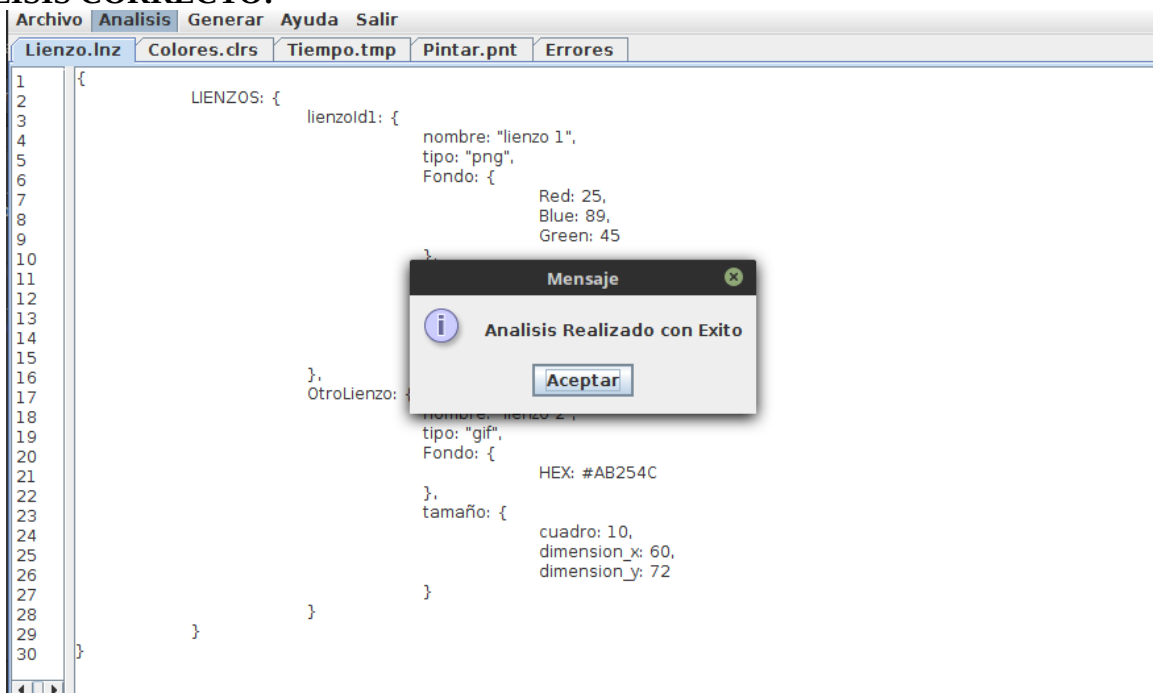
Guardar (Ctrl + S): Guarda los cambios realizados a los archivos.



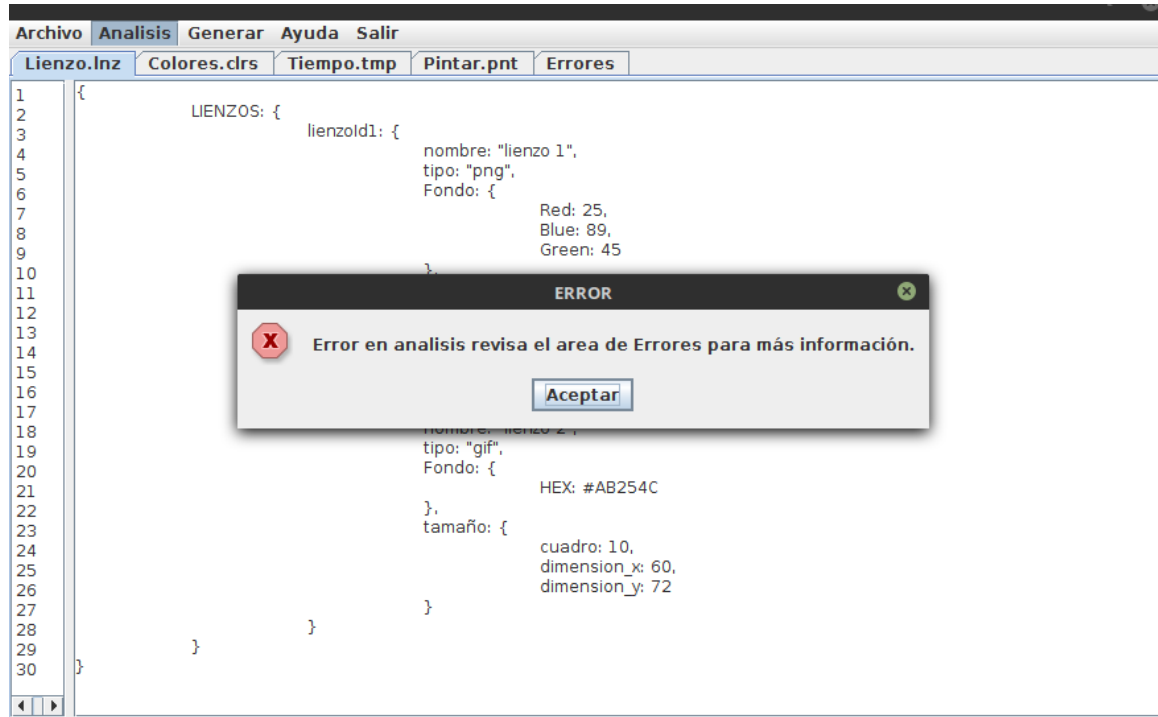
Salir: Cierra la aplicación

Análisis: Realiza un análisis léxico y sintáctico de cada archivo abierto para poder generar las imágenes de la animación posteriormente. Si el análisis detecta algún error entonces las opciones del menú “Generar” se desactivan, de lo contrario las opciones están activas. Si existen errores en el análisis, estos se deben mostrar en la pestaña “Error” indicando tipo, linea, columna, símbolo o lexema de error y una descripción del mismo.

ANÁLISIS CORRECTO:



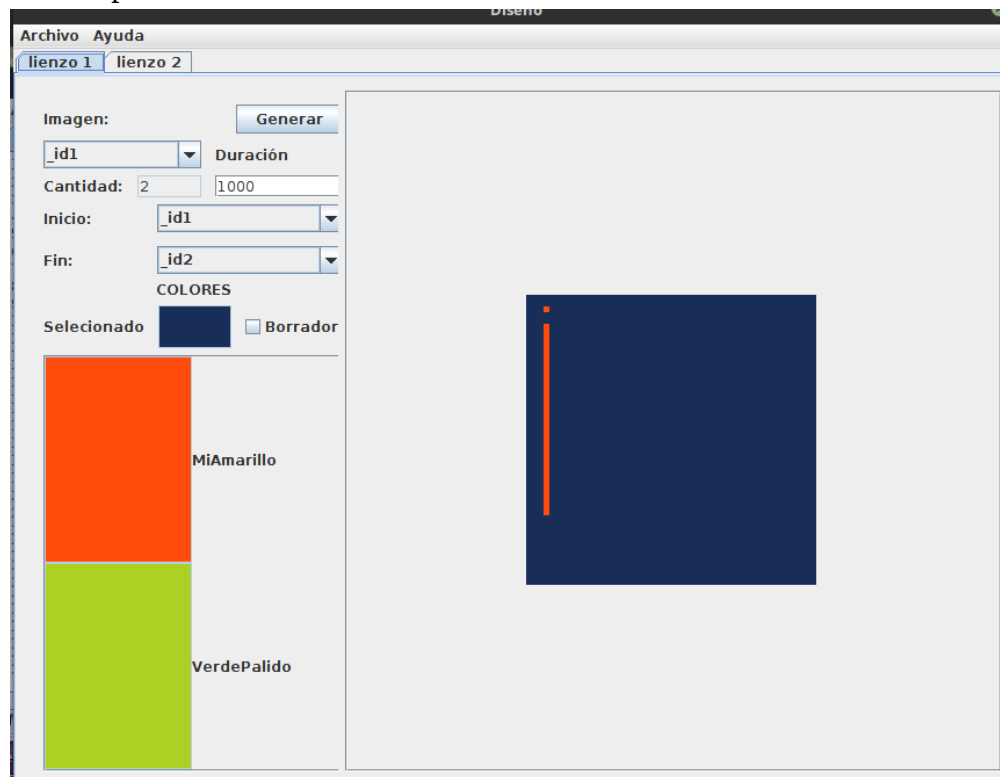
ANÁLISIS CON ERRORES:



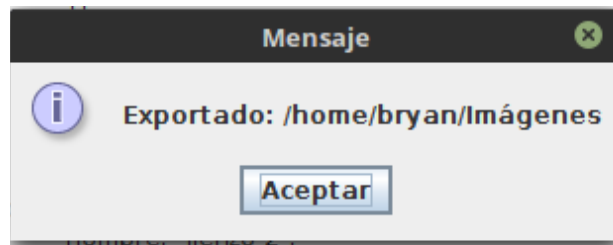
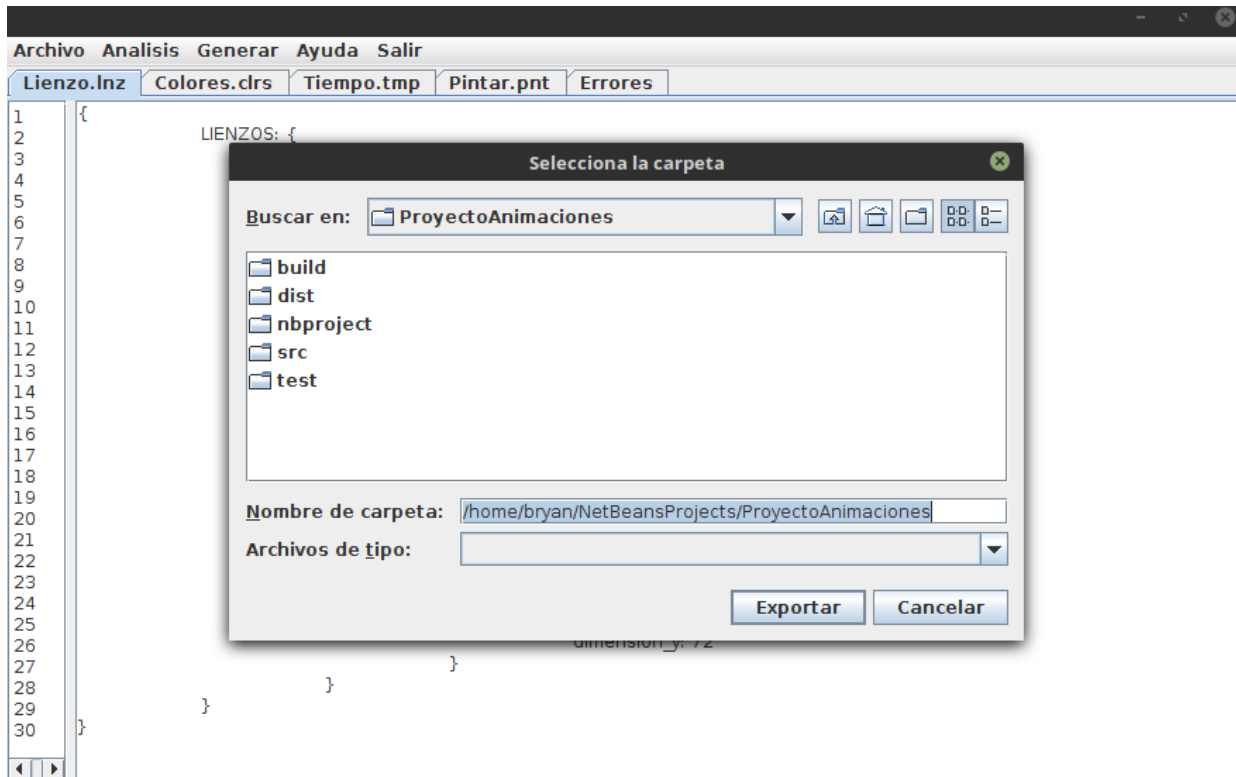
Generar

Las opciones de este menú se activarán sólo si se ha realizado un análisis exitoso con anterioridad.

- **Editor gráfico:** Mostrará la ventana de edición con el fin de poder editar cada imagen correspondiente a la animación esto se hace para poder modificar la animación y generar los cambios en sus respectivo archivo de entrada. El editor gráfico basa su contenido en los códigos de los cuatro tipos de archivos de entrada.

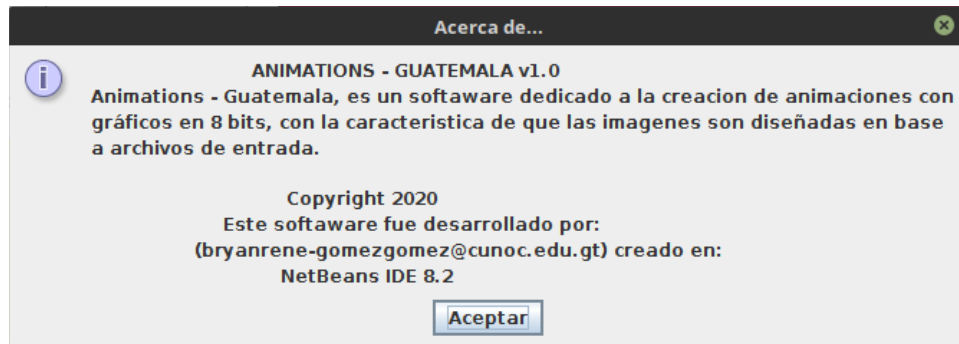


- **Generar:** Genera la animación que ha sido diseñada desde los archivos de entrada.



Ayuda

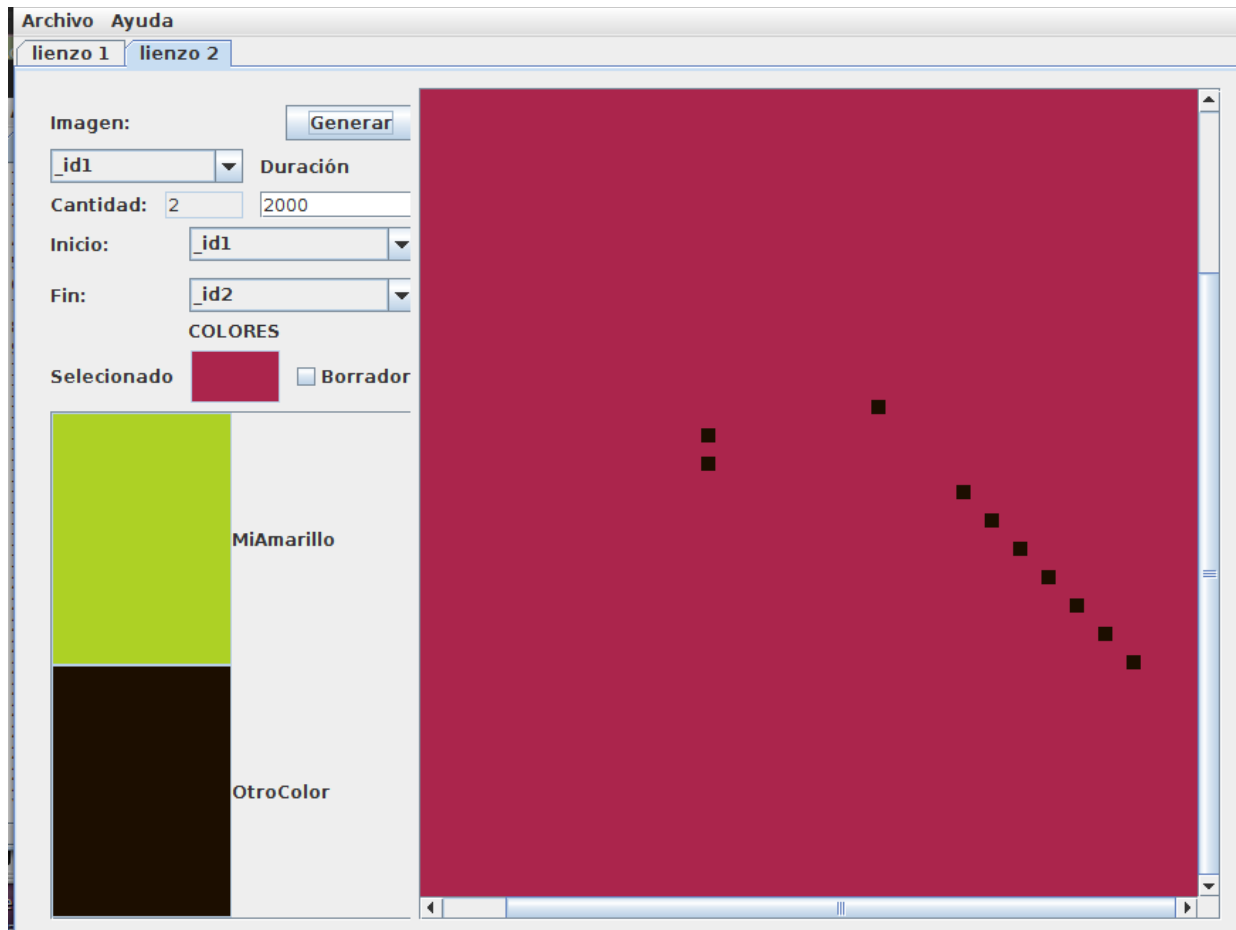
- **Manual de usuario:** abre el manual de usuario
- **Manual técnico:** abre el manual técnico
- **Acerca de:** Muestra información del desarrollador.



Editor gráfico

Basado en los códigos de los archivos de entrada se deberá generar gráficamente los lienzos indicados, reflejando en la interfaz cada característica indicada por los archivos de entrada.

La interfaz gráfica permite al usuario poder realizar cambios en las imágenes de cada lienzo, para poder guardarlos posteriormente y poder exportar la animación creada.



Área de tiempo

En esta parte se muestra la información respecto al tiempo de cada lienzo: duración, inicio, fin y el tiempo correspondiente a la imagen en edición.

- **Inicio:** Este valor nos va a servir al momento de exportar la animación a un gif, ya que se tomarán en cuenta todas las imágenes a partir de ese valor.
- **Fin:** Este valor define la ultima imagen a tomar en cuenta en la animación al exportarla a un gif.
- **Imagen:** Mediante esta opción podremos navegar entre las imágenes que componen nuestro lienzo, el id de de cada imagen y su duración está definida por el archivo con extensión “tmp”. Al momento de seleccionar el id de una imagen específica, se desplegará la forma en que está pintado es imagen en específico.

Área de colores

Aquí se listan los colores configurados en el archivo “clrs” y el usuario puede seleccionar el color deseado y con un clic en el cuadro del lienzo puede pintar el cuadro.

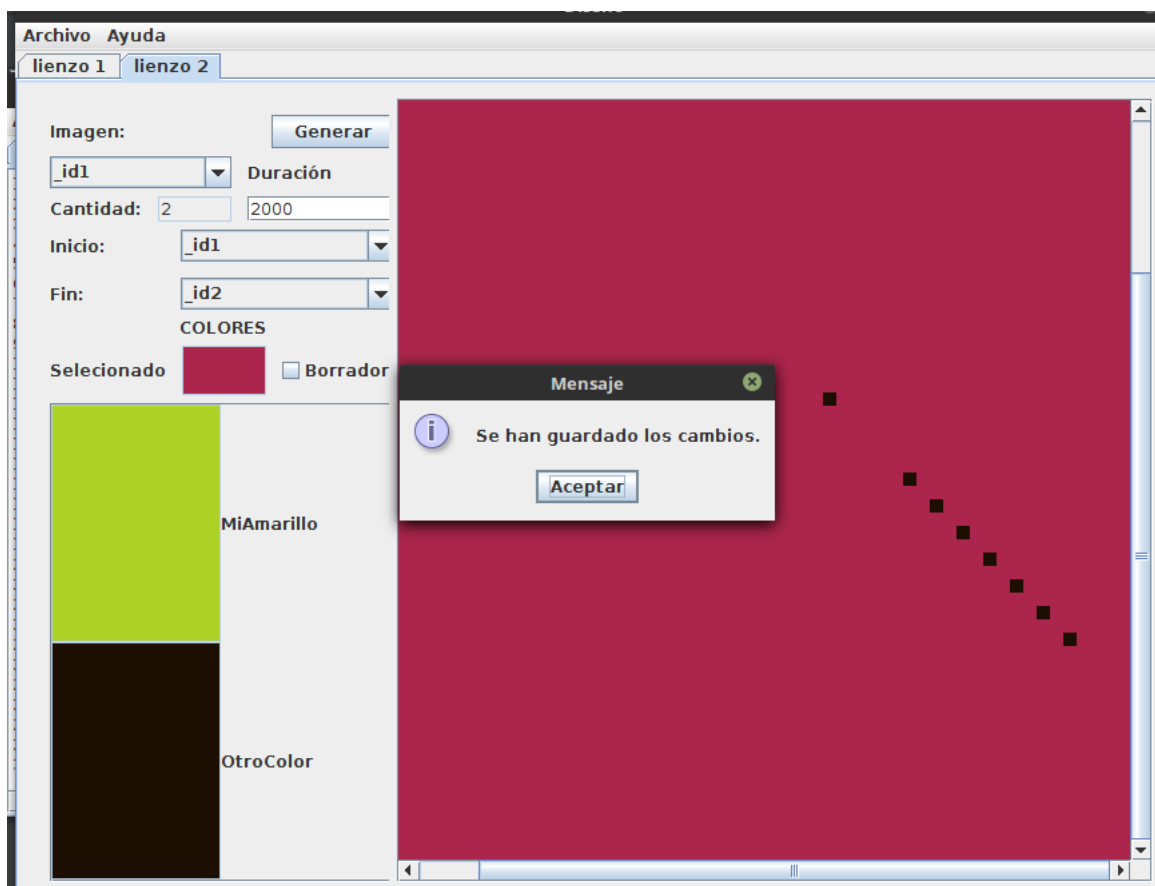
Área de lienzo

Se muestra una grilla del tamaño especificado en el archivo de entrada “lnz”. Cada cuadro puede ser pintado de algún color o también puede ser borrado.

En la parte inferior se muestra el color activo y también se puede activar o desactivar la herramienta de borrado.

La herramienta de borrado remueve el color de un cuadro al hacer un clic sobre dicho cuadro.

Guardar Cambios (Ctrl + S)



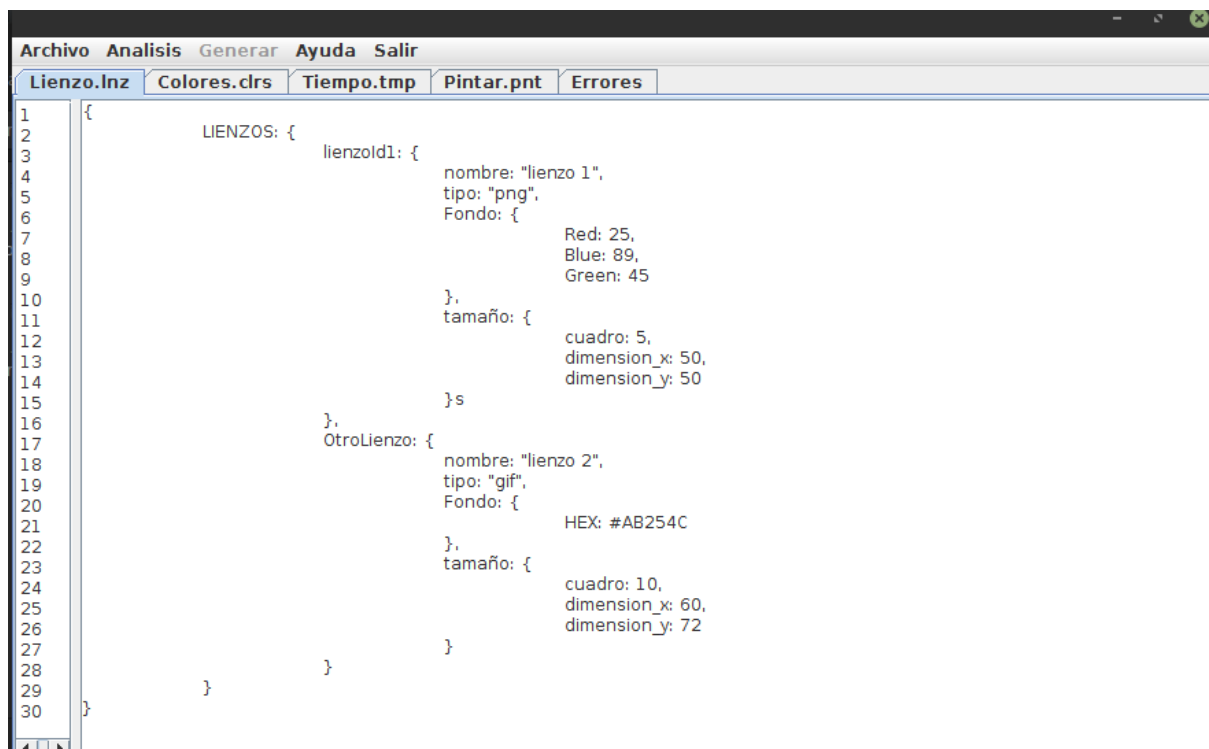
LENGUAJE:

Archivo Lienzo (.lnz):

En este archivo se especifica por medio de una estructura JSON los lienzos del proyecto. La estructura información inicia con la propiedad “LIENZOS” y dentro de ella se pueden agregar varias estructuras LIENZO separadas por coma.

La estructura LIENZO tiene las siguientes propiedades:

- **Identificador:** Será el nombre único que se le dará a cada lienzo. Solo puede contener caracteres alfanuméricos, no puede tener espacios en blanco o caracteres especiales. Los únicos caracteres especiales permitidos son guiones bajos (_) y puede iniciar solo con letras o _
- **Fondo:** Donde se indicará el color de fondo que tendrá el lienzo, pudiendo ser descrito por una estructura RGB o en código hexadecimal con una estructura HEX iniciado por un numeral (#) seguido de seis números hexadecimales, los valores en RGB pueden de aparecer en cualquier orden y tan solo una vez.
- **Tamaño:** Es una estructura JSON la cual Indica el tamaño del lienzo teniendo como propiedades “cuadro” que especificará el tamaño de cada cuadro en pixeles y “dimensión_x” que dará el número de cuadros horizontales y “dimensión_y” que dará el número de cuadros verticales.
- **Nombre:** Se indicará con una cadena de caracteres el nombre de salida de la imagen que se generará, se puede indicar en cualquier lugar pero tan solo una vez.
- **Tipo:** Indicará el tipo de salida, el cual puede ser un conjunto de imágenes PNG o un GIF, se puede indicar en cualquier lugar pero tan solo una vez.



```
1 {
2     LIENZOS: {
3         lienzo1: {
4             nombre: "lienzo 1",
5             tipo: "png",
6             Fondo: {
7                 Red: 25,
8                 Blue: 89,
9                 Green: 45
10            },
11            tamaño: {
12                cuadro: 5,
13                dimension_x: 50,
14                dimension_y: 50
15            }
16        },
17        OtroLienzo: {
18            nombre: "lienzo 2",
19            tipo: "gif",
20            Fondo: {
21                HEX: #AB254C
22            },
23            tamaño: {
24                cuadro: 10,
25                dimension_x: 60,
26                dimension_y: 72
27            }
28        }
29    }
30 }
```

Archivo Colores (.clrs):

Los archivos de colores (“clrs”) almacenan los colores que se pueden usar en determinado lienzo. La estructura del archivo inicia con la propiedad COLORES y dentro de esta se indican los lienzos y sus respectivos colores basado en las siguientes propiedades:

- **Lienzo:** Es el identificador del lienzo al que pertenecen los colores, este lienzo debe de existir para que pueden ser asignados los colores.
- **IdentificadorColor:** Será el nombre único que se le asignará a un color sobre un lienzo. Un lienzo no puede tener colores con identificador repetido. dentro de esa propiedad se especifican los colores pudiendo ser descrito con una estructura RGB o una estructura HEX. Solo puede contener caracteres alfanuméricos, no puede tener espacios en blanco o caracteres especiales. Los únicos caracteres especiales permitidos son guiones bajos (_) y puede iniciar solo con letras o _

Pueden ser declarados uno o muchos colores sobre un lienzo separados por coma (,) y muchos lienzos separados de igual forma.

```
1 {  
2     COLORES: {  
3         lienzoId1: {  
4             MiAmarillo : {  
5                 Red: 255,  
6                 Blue: 12,  
7                 Green: 75  
8             },  
9             VerdePalido : {  
10                HEX: #Acd125  
11            }  
12        },  
13        OtroLienzo: {  
14            MiAmarillo : {  
15                HEX: #ADD125  
16            },  
17            OtroColor : {  
18                Red: 28,  
19                Blue: 0,  
20                Green: 14  
21            }  
22        }  
23    }  
24 }
```

Archivo Tiempo (.tmp):

Este archivo contiene las instrucciones correspondientes a la gestión del tiempo en la animación. La estructura del archivo inicia con la propiedad TIEMPOS y dentro de esta se indican los lienzos y sus respectivos tiempos basado en las siguientes propiedades:

- **lienzo:** Es el identificador del lienzo al que pertenecerán los datos que se estarán indicando, este lienzo debe de existir para que pueden ser asignados los colores.

- **inicio:** Es el id de la imagen a partir de la cual se va empezar la animación excluyendo las anteriores al momento de realizar la exportación.
- **fin:** Es el id de la imagen que será considerada la última que conformará la animación excluyendo las posteriores a ella al momento de realizar la exportación.
- **imagenes:** Es un arreglo compuesto por las propiedades de cada imagen que forman al lienzo para producir la animación. la primer imagen en este atributo tiene el indice cero.
- **id:** Cada imagen que compone un lienzo tiene un id, el cual es un identificador que podrá ser usado al momento de seleccionar la imagen a pintar en un lienzo. Un lienzo no puede tener imágenes con identificador repetido. Solo puede contener caracteres alfanuméricos, no puede tener espacios en blanco o caracteres especiales. Los únicos caracteres especiales permitidos son guiones bajos (_) y puede iniciar solo con letras o _
- **duracion:** Esta propiedad define la cantidad de tiempo que se mostrará determinada imagen antes de pasar a la siguiente, este valor está dado en mili-segundos.

```

2  {
3      TIEMPOS: {
4          lienzoId1: {
5              inicio: "_id1",
6              fin: "_id2",
7              imagenes: [
8                  {
9                      id: "_id1",
10                     duracion: 1000
11                 },
12                 {
13                     id: "_id2",
14                     duracion: 2000
15                 }
16             ]
17         },
18         OtroLienzo: {
19             inicio: "_id1",
20             fin: "_id2",
21             imagenes: [
22                 {
23                     id: "_id1",
24                     duracion: 2000
25                 },
26                 {
27                     id: "_id2",
28                     duracion: 1000
29                 }
30             ]
31         }
32     }
}

```

Archivo Pintar (.pnt):

Este archivo contiene las instrucciones de cómo está pintado cada imagen de cada lienzo y con cuales colores. En este archivo se pueden declarar variables que se pueden usar en el resto del archivo por lo que la estructura está dividida en 2 bloques.

Bloque de declaración de variables (VARS)

El bloque de declaración permite declarar variables de tipo diferentes tipos, las variables pueden o no ser inicializados en esta sección, también se pueden definir variables en lista separadas por comas. Los nombres de variables solo puede contener caracteres alfanuméricos, no puede tener espacios en blanco o caracteres especiales. Los únicos caracteres especiales permitidos son guiones bajos (_) y puede iniciar solo con letras o _

Tipos de variables:

- **int:** almacena un valor entero.
- **String:** almacena una cadena de caracteres.
- **boolean:** almacena un valor booleano, hay dos literales booleanas true y false.

Ejemplos de declaración simple:

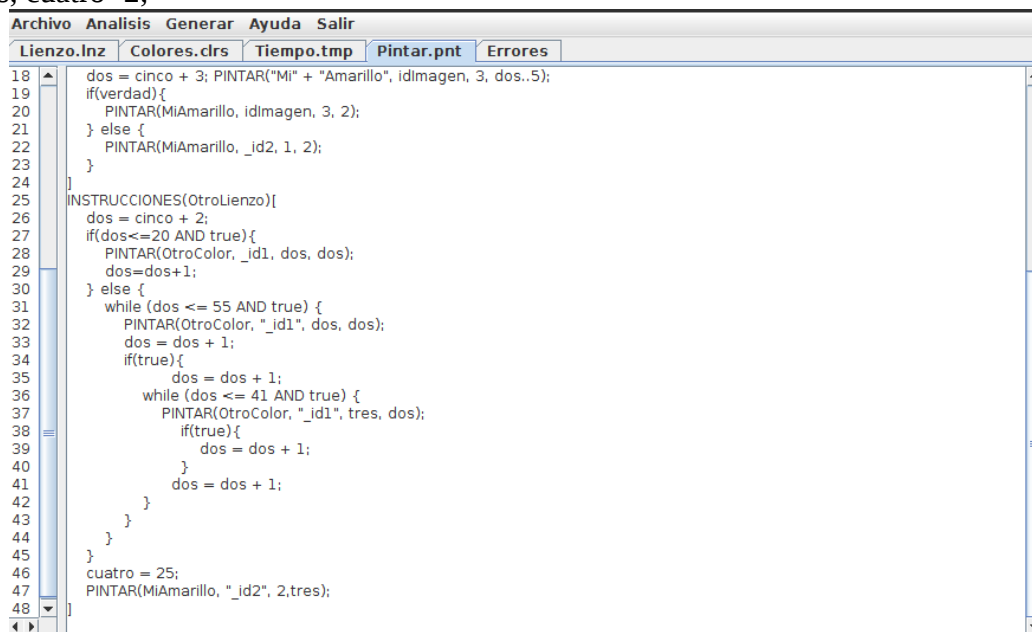
```
int numero;  
int otraVar;  
String nombre;  
boolean varBool;
```

Ejemplos de declaración con inicialización:

```
int numero = 15;  
int otraVar=2 + 3 + numero;  
String nombre="valor";  
String nombre2=nombre;
```

Ejemplos de declaración simple en lista:

```
int numero=2, otraVar;  
int dos, tres, cuatro=2;
```



```
Archivo Analisis Generar Ayuda Salir  
Lienzo.lnz Colores.clrs Tiempo.tmp Pintar.pnt Errores  
18 dos = cinco + 3; PINTAR("Mi" + "Amarillo", idimagen, 3, dos,5);  
19 if(verdad){  
20     PINTAR(MiAmarillo, idimagen, 3, 2);  
21 } else {  
22     PINTAR(MiAmarillo, _id2, 1, 2);  
23 }  
24 ]  
25 INSTRUCCIONES(OtroLienzo)[  
26     dos = cinco + 2;  
27     if(dos<=20 AND true){  
28         PINTAR(OtroColor, _id1, dos, dos);  
29         dos=dos+1;  
30     } else {  
31         while (dos <= 55 AND true) {  
32             PINTAR(OtroColor, "_id1", dos, dos);  
33             dos = dos + 1;  
34             if(true){  
35                 dos = dos + 1;  
36                 while (dos <= 41 AND true) {  
37                     PINTAR(OtroColor, "_id1", tres, dos);  
38                     if(true){  
39                         dos = dos + 1;  
40                     }  
41                     dos = dos + 1;  
42                 }  
43             }  
44         }  
45     }  
46     cuatro = 25;  
47     PINTAR(MiAmarillo, "_id2", 2,tres);  
48 ]
```

Bloque de declaración de instrucciones (INSTRUCCIONES)

Por cada lienzo que se trabaja se debe agregar un bloque INSTRUCCIONES y entre paréntesis el identificador del lienzo. En este bloque se podrán usar las siguiente instrucciones:

- **asignación de valores**
- **pintar**
- **if - else**
- **while**

Instrucción Asignación de Valores

```
color1="idColor";  
lienzo1="idLienzo";  
otraVar = numero -3;  
numero = 200 - numero - 35;  
varBool = otraVar > 5;
```

Operaciones Aritméticas

Las operaciones aritméticas permitidas son:

Suma (+)
Resta (-)
Multiplicación (*)
División (/)

Y solo se aplican a valores y variables enteras.

Instrucción pintar

Esta instrucción tiene la siguiente estructura: **PINTAR(idColor,idImagen, posX, posY)** y se emplea 4 parámetros para esta instrucción.

- **idColor:** Este valor corresponde a un id de algún color definido en el archivo “clr”.
- **idImagen:** Este parámetro indica el id de la imagen a pintar, el id está definido en el archivo “tmp”.
- **posX:** Este valor corresponde a la posición en el eje x del cuadro a pintar.
- **posY:** Este valor corresponde a la posición en el eje y del cuadro a pintar.

Dentro de los parámetros idColor, idImagen, posX y posY se pueden usar variables, operaciones aritméticas y una sub-estructura especial de rango usando puntos seguidos: (..) para indicar un rango que se va a pintar.

Instrucción while

Esta instrucción tiene la siguiente estructura: **while(condiciones lógicas){ }** y representa un ciclo que se detiene hasta que la condición lógica no se cumpla. La condición lógica está compuesta por booleanos, operaciones relacionales y operaciones lógicas.

Instrucción if-else

Esta instrucción tiene la siguiente estructura: **if(operaciones lógicas){ } else { }** la sentencia else es opcional. La condición lógica está compuesta por booleanos, operaciones relacionales y operaciones lógicas.

Operaciones Relacionales:

1. ==
2. <
3. >
4. <=
5. >=
6. <>

Las operaciones relacionales permiten evaluar y comparar valores o variables enteras.

Operaciones Lógicas:

1. AND
2. OR

Los operadores lógicos permiten encadenar operaciones relacionales.

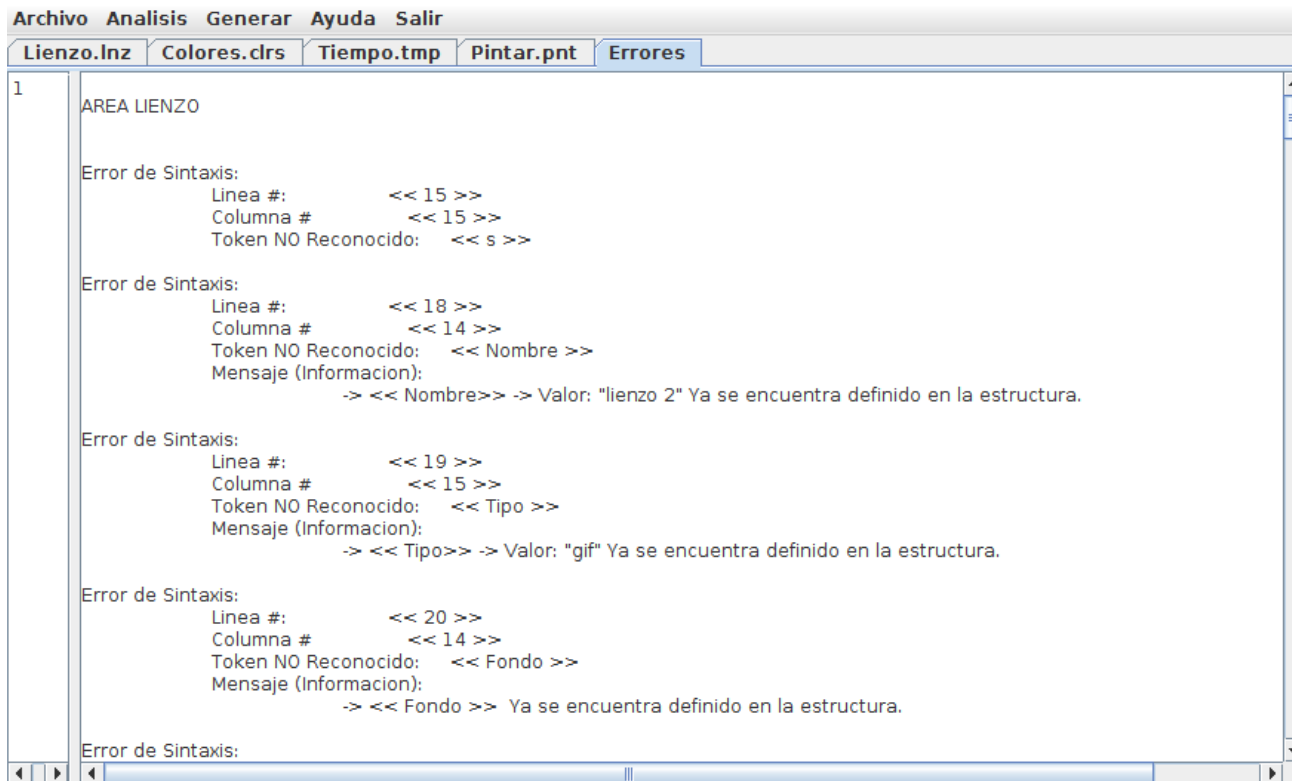
Comentarios

Dentro del archivo .pnt pueden insertarse comentarios tanto de una línea (//) como de múltiples líneas (/* */)

Reporte de errores léxicos, sintácticos y semánticos

Si el análisis detecta algún error de tipo léxico, sintáctico y semántico entonces muestran en la pestaña “Error” indicando:

- **Tipo**
- **Linea**
- **Columna**
- **Símbolo o lexema de error**
- **Descripción del error.**



REQUISITOS MÍNIMOS PARA ESTE PROGRAMA:

- Microsoft .NET Framework versión 4.0
- Windows XP, Vista, 7, 8, 8.1, 10
- Linux Ubuntu 15.04 y Superior:
- Espacio en disco: 100 MB de espacio libre en disco
- Pentium 1 GHz o superior con 1GB de RAM o más.