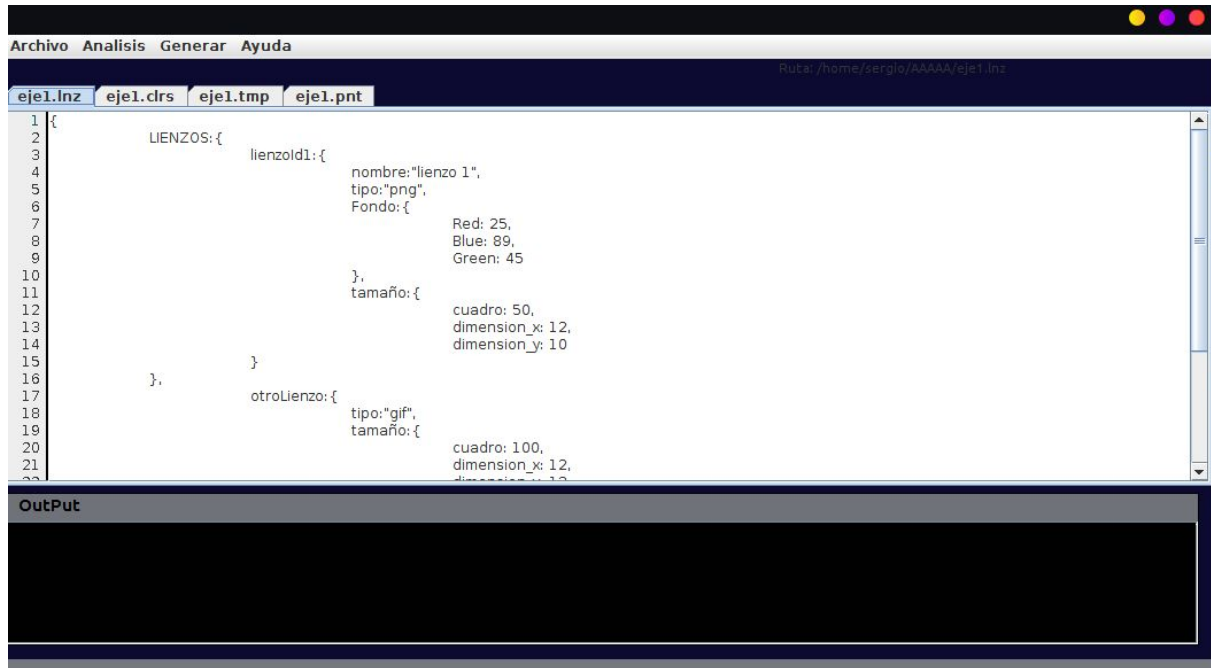


Manual De Usuario

El objetivo principal de la aplicación es la creación de imágenes png y gif mediante archivos de entrada que describen al imagen. Esta aplicación se encargará de analizar dichos archivos de entrada para formar un imagen y luego construirlo para el uso del usuario.

Antes de empezar el análisis se permite la posibilidad al usuario de editar los archivos de entrada mediante un editor de texto integrado, en el cual se mostrarán cuatro pestañas siendo estas las cuatro necesarias para construir los imnganes.



Aqui se vera el numero de fila y el contenido del archivo que se desea analizar. Como anteriormente mencionado existen cuatro pestañas representando los archivos de entrada que son de extensión:

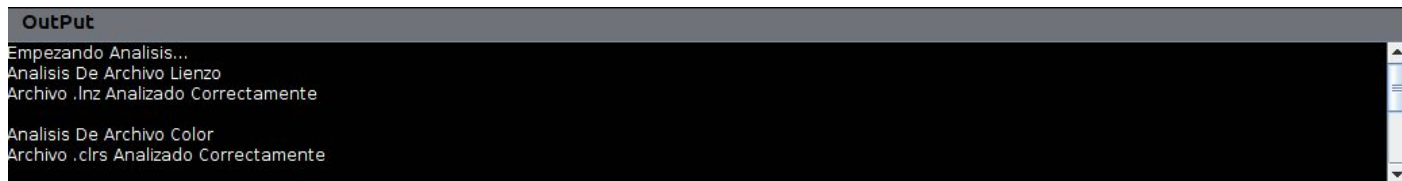
- Lienzo(.lnz)
- Color(.clrs)
- Tiempo(.tmp)
- Pintar(.pnt)



Para cargar un archivo existente de tu computadora al editor de texto, debes de ir a la pestaña "Archivo" y hacer click en Abrir, en donde se mostrará un buscador de archivos en donde debes de seleccionar el que desees abrir, recuerda que la extensión debe de ser una de las anteriores. Si quieres crear un archivo desde cero debes dirigirte a la misma pestaña pero ahora seleccionamos la opción nuevo, en donde seleccionas la ubicación donde quieres guardarlo y su nombre.

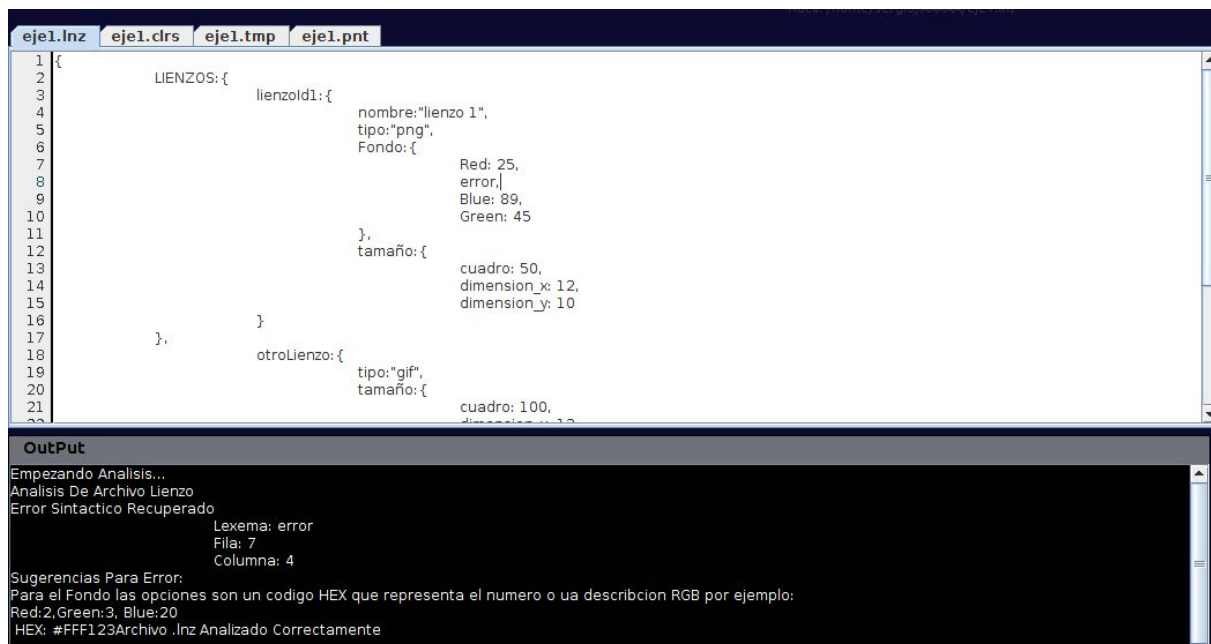
Para guardar los cambios que se realizan puedes hacer click en la misma pestaña o con Ctrl+S.

Cuando ya tengas los archivos listos puedes analizarlos en la pestaña Analizar, para ver los resultados de dicho análisis se tiene un panel que se llama Output en la parte inferior, aquí se verán los errores si existieran o se verá que el archivo fue analizado con éxito.



Como anteriormente mencionado aqui se veran los errores existen en cada archivo, indicando así el número de fila y el carácter o conjunto de caracteres provocando el error, ya sea lexico, sintactico o semantico, para el facilitamiento de correcciones de parte de usuario.

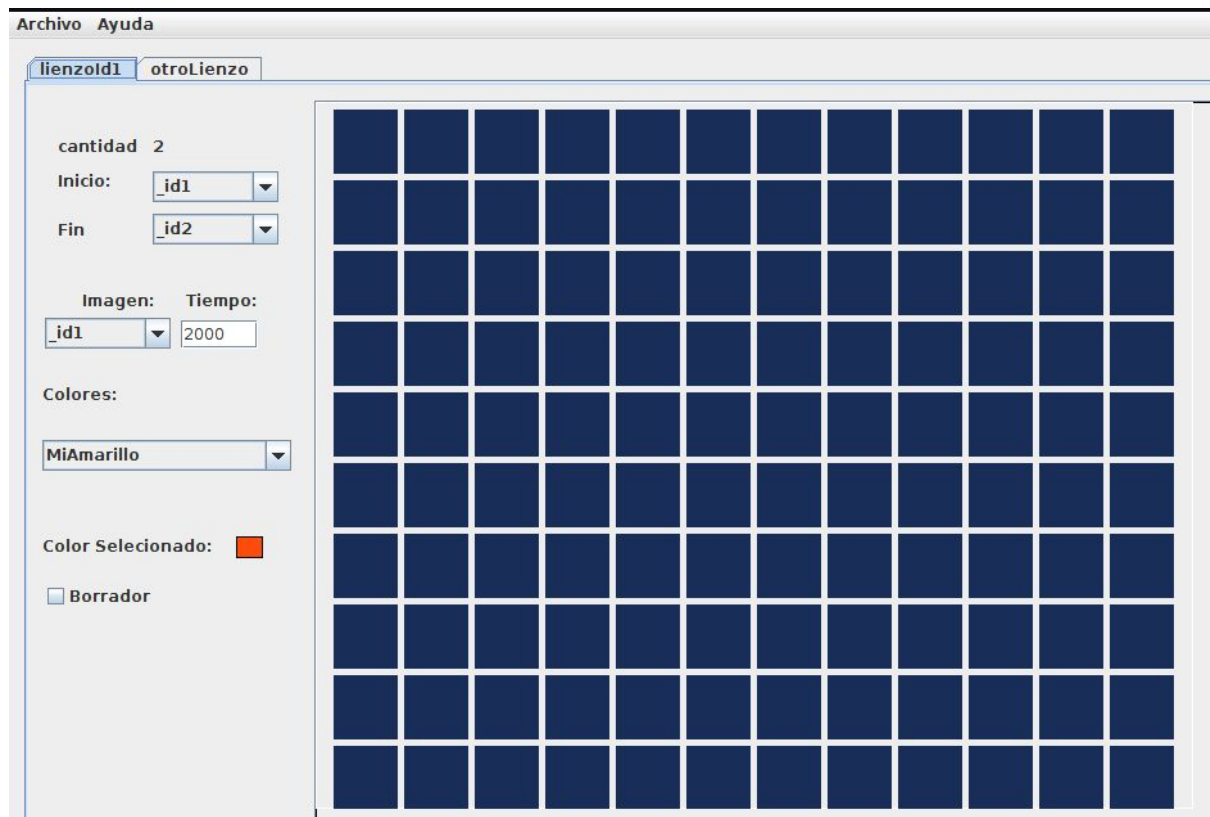
Como se puede ver en la figura siguiente, existe un error en la fila 7 lo cual es provocado por el lexema "error" y posterior a eso nos indica el Output una sugerencia para corregir el error.



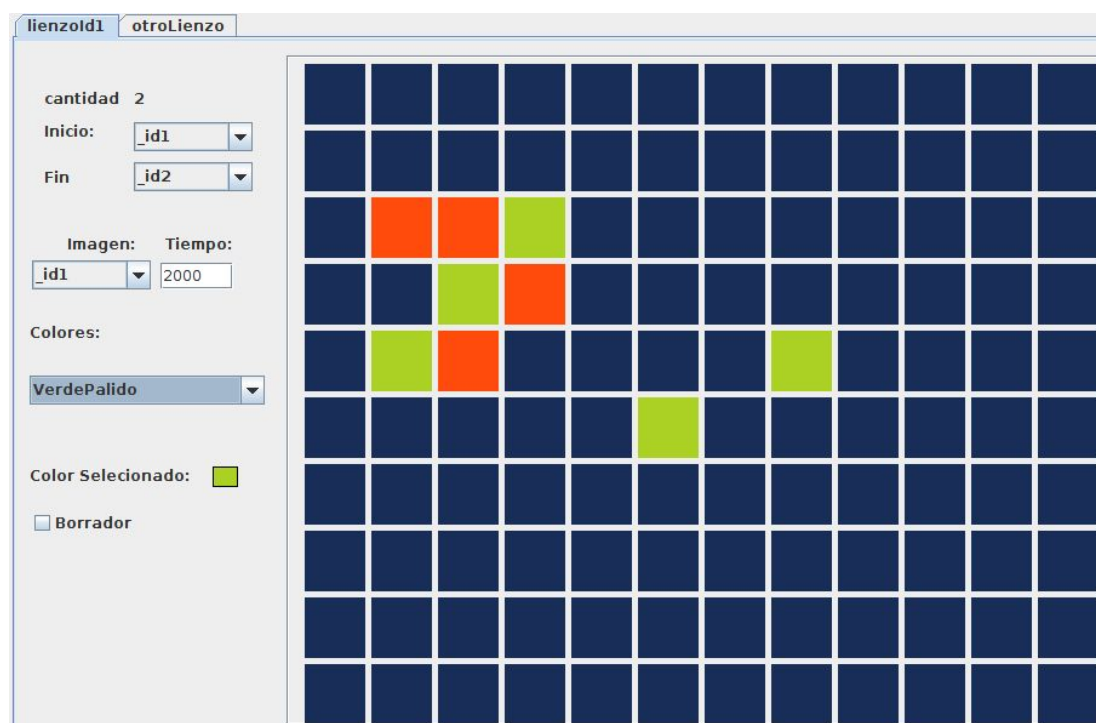
Al momento de analizar cada archivo tendrá su bloque en donde se indique si el análisis fue exitoso o no.

Si el análisis es exitoso se podrá ejecutar la función de generar Lienzo o de Editar Gráfico de la pestaña de Generar.

La función editar gráfico abrirá una pestaña que se dedique a facilitar la edicion de los lienzos provenientes de los archivos de entrada para el usuario. En donde se mostrarán los lienzos existentes y sus imágenes posibles a editar.



En el caso de la figura anterior se puede apreciar la existencia de dos lienzo en el cual el primero cuenta con 2 imágenes y actualmente referenciando el imagen _id1, aqui se podra editar el imagen seleccionando el color en Colores y luego dandole click a uno de los píxeles existentes del imagen esto llevará a que cambie de color. También se puede usar el borrador que consiste en regresar el color a el color de fondo del lienzo.



Aquí se puede cambiar el inicio y fin de los imágenes a exportar y además el tiempo que van a permanecer mostrados si se trata de un gif.

Si se termina de Editar los lienzos para generarlos en png o gif te debes de dirigir a la pestaña de archivo y hacer click en el menú que dice generar. Si todo marcha bien los imágenes se generan en la carpeta imágenes del proyecto y se cerrará el editor de Imágenes. Dependiendo del tipo de Lienzo se generará un gif o png.



La última pestaña de la aplicación es la de ayuda, en el cual se mostrará información sobre el programa incluyendo el manual Técnico, manual de Usuario e Información de desarrollador.

Explicación De los Archivos De Entrada:

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.

Ejemplo:

```
{
  LIENZOS:{
    lienzo1:{
      nombre:"lienzo 1",
      tipo:"png",
      Fondo:{
        Red: 25,
        Blue: 89,
        Green: 45
      },
      tamaño:{
        cuadro: 50,
        dimension_x: 12,
        dimension_y: 10
      }
    },
    otroLienzo:{
      tipo:"gif",
      tamaño:{
        cuadro: 100,
        dimension_x: 12,
        dimension_y: 12
      },
    },
  },
}
```

```

        nombre:"lienzo 2",
        Fondo:{
            HEX: #AB254C
        }
    }
}
}

```

Archivo Color(.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.

Ejemplo:

```

{
    COLORES:{
        lienzo1:{
            MiAmarillo:{
                Red: 255,
                Blue: 12,
                Green: 75
            },
            VerdePalido:{
                HEX: #Acd125
            }
        },
        otroLienzo:{
            MiAmarillo:{
                HEX: #ADD125
            },
            otroColor:{

```

```

    Green: 14,
    Red: 28,
    Blue: 0
  }
}
}

```

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 milisegundos.

```

{
  TIEMPOS:{
    lienzold1:{
      inicio:"_id1",
      fin:"_id2",
      imagenes:[
        {
          id:"_id1",
          duracion:2000
        }
      ]
    }
  }
}

```

```

        },
        {
            id:"_id2",
            duracion: 1000
        }
    ]
},
otroLienzo:{
    inicio:"_id1",
    fin:"_id2",
    imagenes:[
        {
            id:"_id1",
            duracion:2000
        },
        {
            id:"_id2",
            duracion: 1000
        }
    ]
}
}
}

```

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.

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 siguientes instrucciones:

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

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 subestructura 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:
==
<
>
<=
>=
<>

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

- Operaciones Lógicas:

AND

OR

Los operadores lógicos permiten encadenar operaciones relacionales.

Comentarios

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

Ejemplo completo del archivo pnt:

```
VARs [
/*
  Sección de variables
*/
  int uno = 25; //primera variables
  int dos, tres = 25;
  int cuatro;
  String idImagen="_id1",idImagen2="_id2";
  int cinco = uno - 25 + 35;
  boolean verdad=true, falso=false;
]
INSTRUCCIONES(lienzold1) [
/*
  Sección de sentencias
*/
  dos = cinco + 3; PINTAR("MiAmarillo",idImagen, 2..5, 2);
  if(verdad){
    PINTAR("MiAmarillo",idImagen,2 + 3, cuatro);
  }else{
    PINTAR("MiAmarillo","_id2",1, 2);
  }
]
INSTRUCCIONES(otroLienzo) [
  if(tres==dos){
    PINTAR("otroColor",idImagen2,uno,uno);
  }else{
    while(dos<=20 AND true){
      PINTAR("otroColor","_id1",dos,dos)
      dos=dos+1;
    }
  }
  cuatro = 25;
  PINTAR("MiAmarillo", "_id2",2,tres..cinco);
```

