

# **MANUAL DE USUARIO**

## **(IDE Animaciones)**

## **Requerimientos**

Sistema Operativo (Windows o Linux)

Java 1.8.0\_201 o compatibles.

Jar ejecutable o proyecto completo.

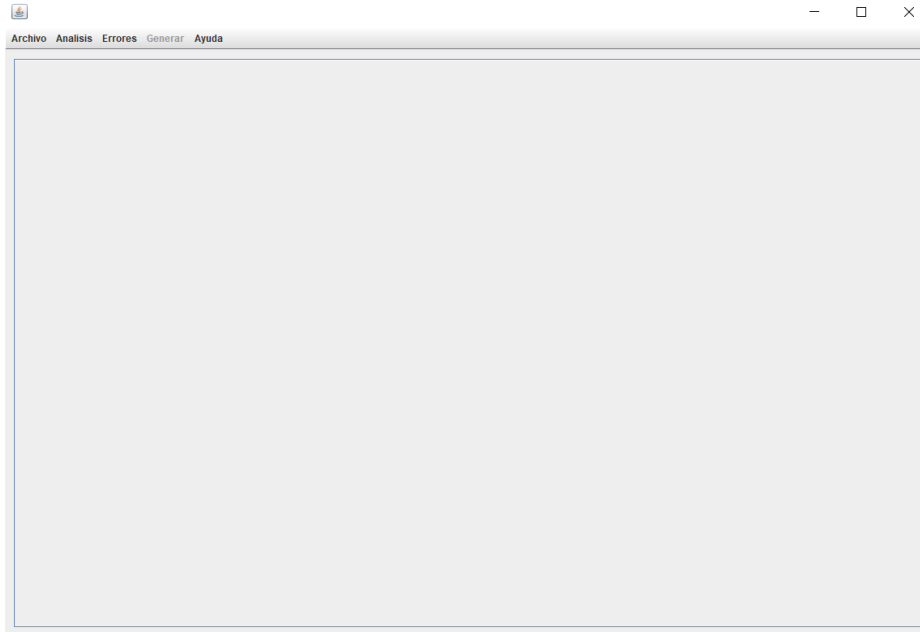
Entender lenguaje Json (para editar archivos en el IDE).

Los archivos deben de tener la extencion: .lnz, .clrs, .tmp y .pnt

## Descripción

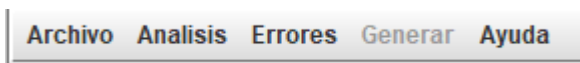
El programa tiene dos partes, la primera es un editor de código en el cual el usuario podrá escribir la configuración de cada tipo de archivo, la segunda parte es un editor de imágenes en el cual el usuario podrá seleccionar colores y pintar en su respectiva imagen.

## IDE editor de Código



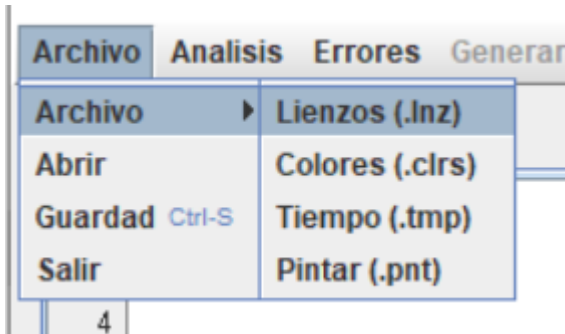
En esta parte es donde el usuario generara y analizara los archivos de entrada para la creación de la segunda parte del IDE.

## Opciones



## Pestaña Archivo

En esta pestaña el usuario podrá crear, abrir y guardar los archivos de entrada, además de que también contiene la opción de salir del IDE.

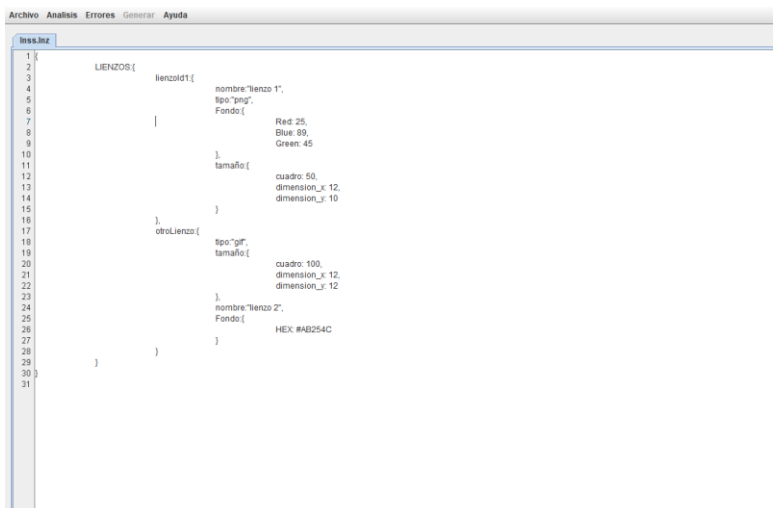


**Archivo:** En esta sección el usuario podrá crear nuevos archivos según sea el tipo de archivo o extensión que necesite crear.

**Abrir:** abre un archivo de cualquier extensión (de las 4 que se tienen), y este será leído para luego poder editarlo.

**Guarda:** guarda los cambios realizados en el código de cada archivo que este abierto dentro del IDE.

**Salir:** Saldrá y cerrará el programa.



Una vez se abra o edite un archivo de entrada se podrá visualizar de esta manera.

Ejemplos de una estructura básica de cada archivo:

### Archivo .lnz

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.

```
{
  LIENZOS:{
    lienzoId1:{
      nombre:"lienzo 1",
      tipo:"png",
      tamaño:{
        cuadro: 50,
        dimension_x: 12,
        dimension_y: 10
      },
      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 .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 \_

```
{
  COLORES:{
    lienzoId1:{
      MiAmarillo:{
        Red: 255,
        Blue: 12,
        Green: 75
      },
      VerdePalido:{
        HEX: #Acd125
      }
    },
    otroLienzo:{
      MiAmarillo:{
        HEX: #ADD125
      },
      otroColor:{
        Green: 14,
        Red: 28,
        Blue: 0
      }
    }
  }
}
```

### Archivo .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 primera imagen en este atributo tiene el índice 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:{
    lienzoId1:{
      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 .pnt

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 pueden 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;
```

### **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 Asignación de valores**

```
color1="idColor";
```

```
lienzo1="idLienzo";
```

```
otraVar = numero -3;
```

```
numero = 200 - numero - 35;
```

```
varBool = otraVar > 5;
```

#### **Instrucción pintar**

Esta instrucción tiene la siguiente estructura: `pintar(idColor,idImagen, posX, posY)` y se emplea 4 parametros para esta instruccion

**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

```

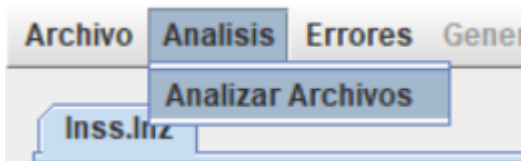
(/* */)

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(lienzoId1) [
    /*
    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);
]

```

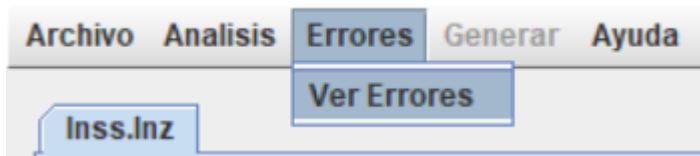
## Pestaña Análisis

En esta pestaña activara un recorrido de cada archivo asi de esta manera, se podrá determinar el estado de cada archivo si es correcto o incorrecto, si es correcto activara la pestaña de Generar, de lo contrario activara la pestaña Errores.

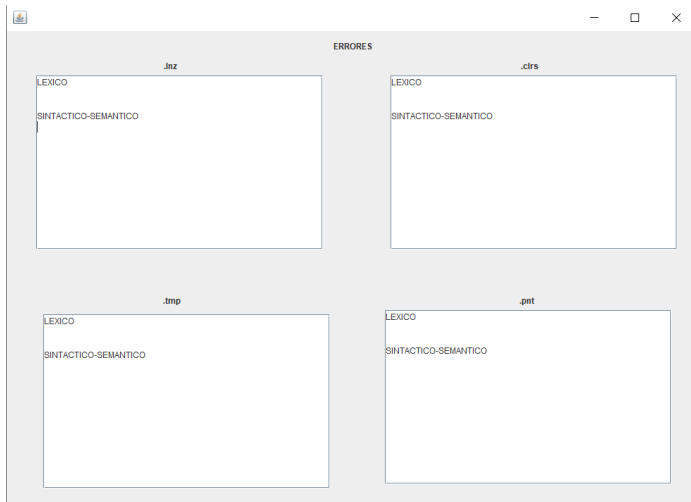


Seleccionando Analizar Archivos, analiza los cuatro archivos de distinto tipo, para esto deben de abrirse los archivos de lo contrario el análisis estará incorrecto.

## Pestaña Errores



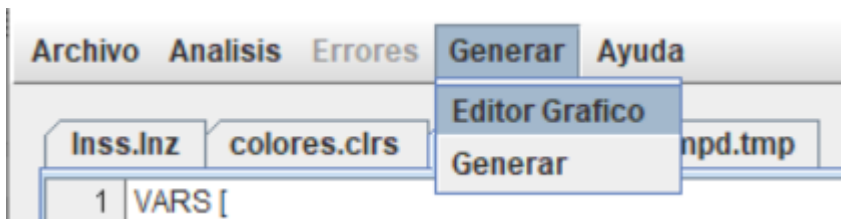
Una vez activada esta pestaña el usuario podrá visualizar el listado de errores que se tuvieron en cada archivo, basta con darle clic en Ver Errores para visualizar la siguiente ventana:



Se divide en cuatro áreas cada área es para cada uno de los 4 tipo de archivo, en cada área se tiene una sección de errores léxicos, y una de errores sintácticos y semánticos. En caso de que los archivos contengan errores estos se mostrarán según sea su sección.

### Pestaña Generar

Se activará cuando el análisis de los archivos haya sido exitoso.



Esta pestaña contiene dos opciones:

**Editor Grafico:** mostrará la ventana inicial del IDE editor gráfico, la cual si descripción estará en sus sección.

**Generar:** generar las respectivas imágenes de cada lienzo, únicamente mostrara una ventana donde el usuario seleccionara la ruta o ubicación donde se guardaran las imágenes.

## Pestaña Ayuda



En esta seccion se brinda información al usuario.

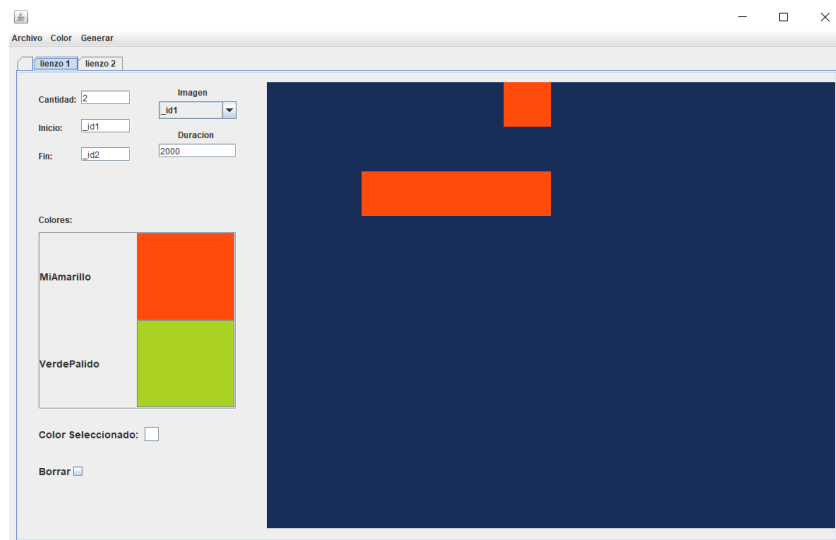
**Manual de Usuario:** mostrara dicho documento para ser visualizado.

**Manual Técnico:** mostrara dicho documento para ser visualizado.

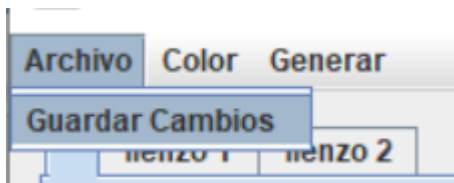
**Acerca de:** brinda información mínima sobre el programador y el IDE.

## IDE Editor Grafico

En esta segunda parte, se edita manualmente cada imagen que contenga un lienzo, además de que se pueden realizar algunos cambios que también son afectados en los archivos de entrada.

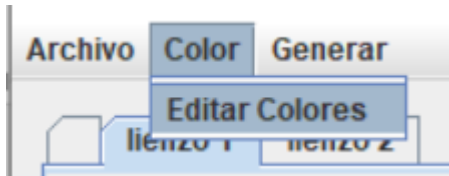


## Pestaña Archivo

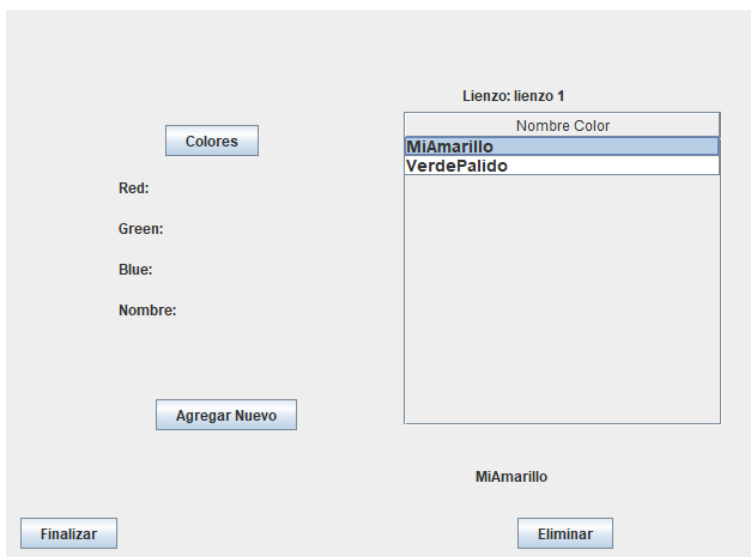


En esta seccion Guardar Cambios generara y guardara los cambios realizador en el editor grafico para el editor de código según el archivo seleccionado.

### Pestaña Color



Si el usuario desea editar los colores de su lienzo seleccionado, esta seccion ayudara agregar o eliminar colores del listado actual. Al hacer clic en la opción Editar Colores se mostrará la siguiente ventana.



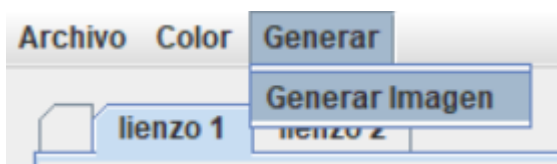
En esta ventana podrá visualizar una gama de colores amplia.

Para agregar un color nuevo hacer clic en el botón colores, este le mostrara la paleta de colores disponibles, una vez seleccionado, deberá escribir el nombre que le dará al nuevo color, luego se mostrara la gama RGB del color seleccionado. Por último, haz clic en agregar Nuevo para guardar el nuevo color.

Para eliminar haga clic en el color deseado dentro de la tabla de colores, una ves hecho esto haga clic en el botón eliminar para realizar el cambio.

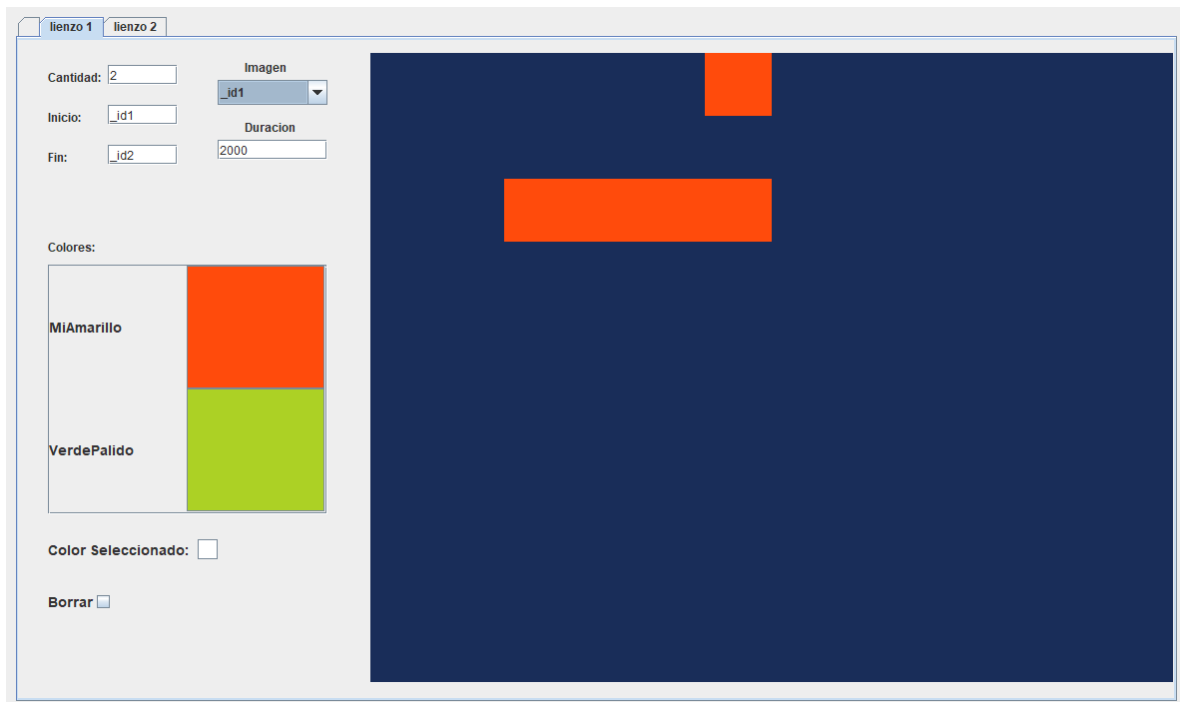
Una ves que termine haga clic en finalizar para regresar al edito gráfico.

### Pestaña Generar



Esta opción generar las imágenes correspondientes al cada lienzo, únicamente deberá seleccionar la ruta de la ubicación donde se guardarán dichas imágenes.

## Ventana Editor Grafico



En esta seccion de la ventana el usuario tendrá distintas opciones de edición.

El panel derecho contiene la imagen generada del archivo, para realizar cambio en esta parte basta con hacer clic en ella, cabe mencionar que un clic pintara un cuadro dentro de la imagen realizada.

**Cantidad:** denota la cantidad de imágenes que contiene el lienzo.

**Inicio:** El id de la imagen con la que empezara la animación gif en caso de que esta sea la configuración.

**Fin:** El id de la imagen con la que empezara la animación gif en caso de que esta sea la configuración.

**Imagen:** Muestra un listado de imágenes, una vez que sea seleccionada la imagen esta será que se mostrara en el panel de la derecha.

**Duración:** Es el tiempo en que durara cada animación de la imagen seleccionada, este tiempo es en milisegundos.

### Seccion colores

En esta seccion se muestra el listado de colores obtenidos de los archivos, para selección un color haga clic en el botón del color que de sea y este será con el que se pintara la imagen.

La descripción de color seleccionado será únicamente para mostrar el color elegido por el usuario.

La seccion de borrar, una ves que se activa borrara el color del cuadro donde se haga clic en la imagen.