

# MANUAL DE USUARIO

ANALIZADOR LEXICO

# Requisitos de sistema

## Requisitos de Hardware

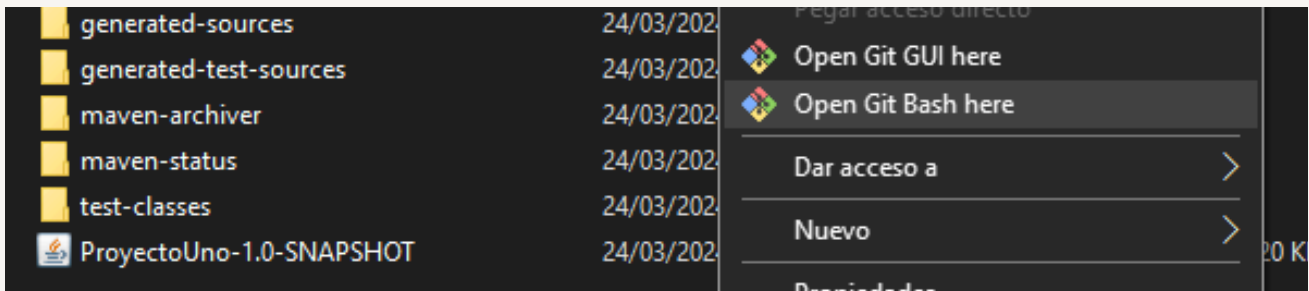
- Procesador: Cualquier procesador compatible con los sistemas operativos actuales.
- Memoria RAM: 2 GB o más recomendado.
- Tener instalado GraphViz Version dot
- Tener instalado Java 17 o posterior

## Requisitos de Software, cualquier sistema operativo:

- Windows (7, 8, 10)
- macOS
- Linux

# Iniciar Analizador Lexico

En el paquete proporcionado por el desarrollador encontrara un archivo con nombre como el de la imagen. tiene que tener instalado la consolo gitbash de preferencia,



PRESIONA DOS VECES SOBRE EL ARCHVIO DE PROGRAMA Y SE EJECUTARA EL PROGRAMA

# Tamaño de Lienzo

En la interfaz del programa, hay dos campos de entrada en la parte superior, etiquetados como "Número Filas" y "Número Columnas". El usuario debe ingresar la cantidad deseada de filas y columnas en estos campos para definir la estructura de la cuadrícula.

Una vez introducidos los números en ambos campos, el usuario debe presionar el botón "Establecer Tamaño". Al hacerlo, se generan cuadros de la cantidad indicada que se ubicarán automáticamente en el panel rojo a la derecha de la pantalla. Esta acción ajusta el tamaño del lienzo de acuerdo con los valores proporcionados.

31 de ago 19:54

es1

28 %

Numero Filas3Numero Columnas3

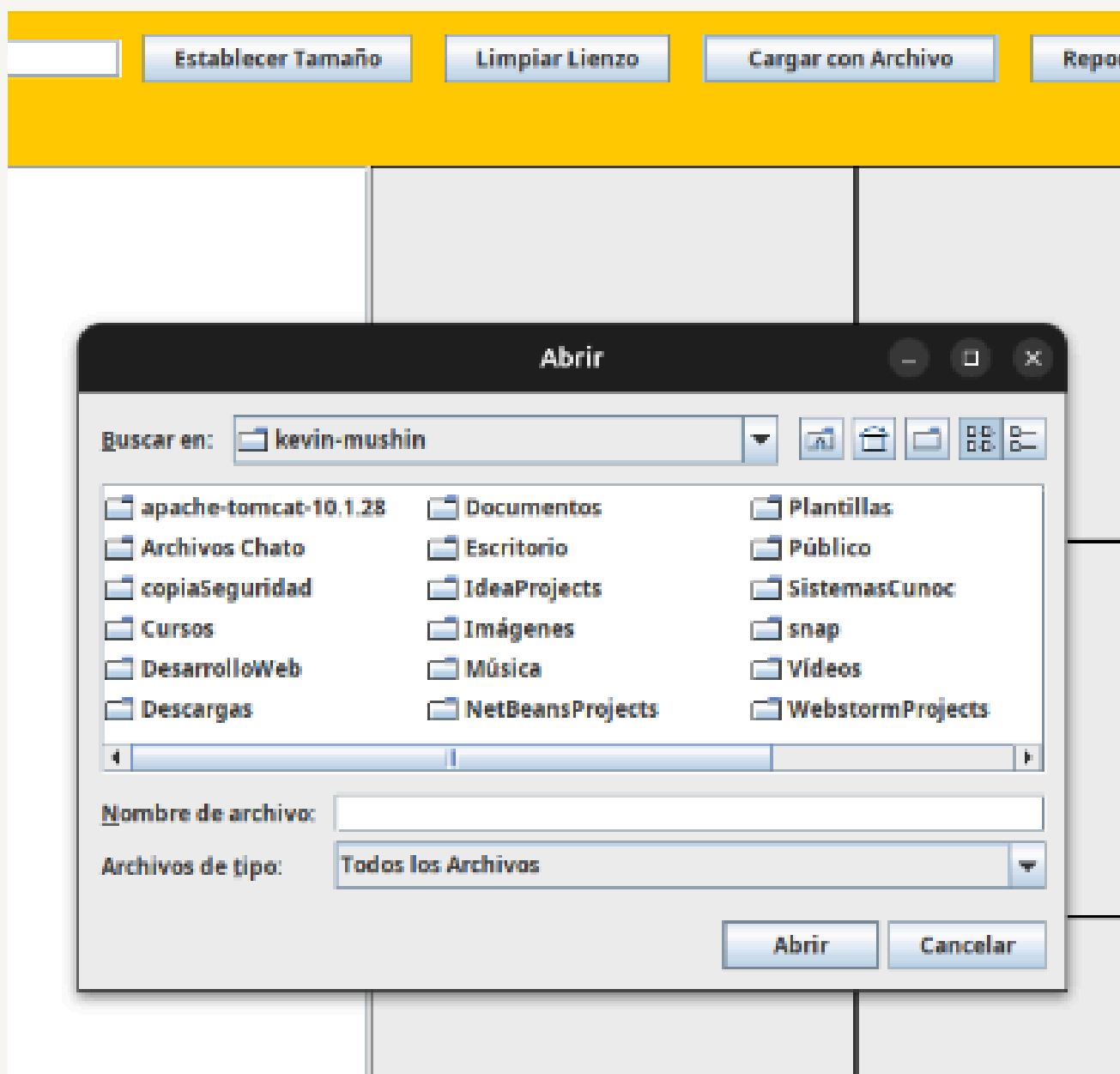
Establecer TamañoLimpiar LienzoCargar con ArchivoReporte AnalisisGuardar Lienzo

VacioANALIZAR


# Cargar Archivo Texto

Cuando el usuario hace clic en el botón "Cargar con Archivo", se activa una función que despliega un cuadro de diálogo del sistema operativo. Este cuadro de diálogo permite navegar por las carpetas y directorios del equipo para localizar un archivo de texto (.txt). El usuario puede explorar diferentes ubicaciones, como "Documentos", "Descargas", u otros directorios personales, para encontrar el archivo que desea cargar.

Una vez seleccionado el archivo deseado, el usuario debe hacer clic en el botón "Abrir" del cuadro de diálogo. Al hacerlo, el contenido del archivo de texto se importa automáticamente y se muestra en el JTextArea de la aplicación, una sección que permite visualizar y editar texto. Esta área de texto es un componente que puede expandirse para mostrar grandes cantidades de datos de manera organizada.



Al presionar el botón de cargar archivo se reflejara el texto en el área de texto en el cual ademas de ingresar el archivo se puede agregar texto o eliminar texto , ademas se muestra la fila , columna de la ubicacion del cursor y el numero de linea

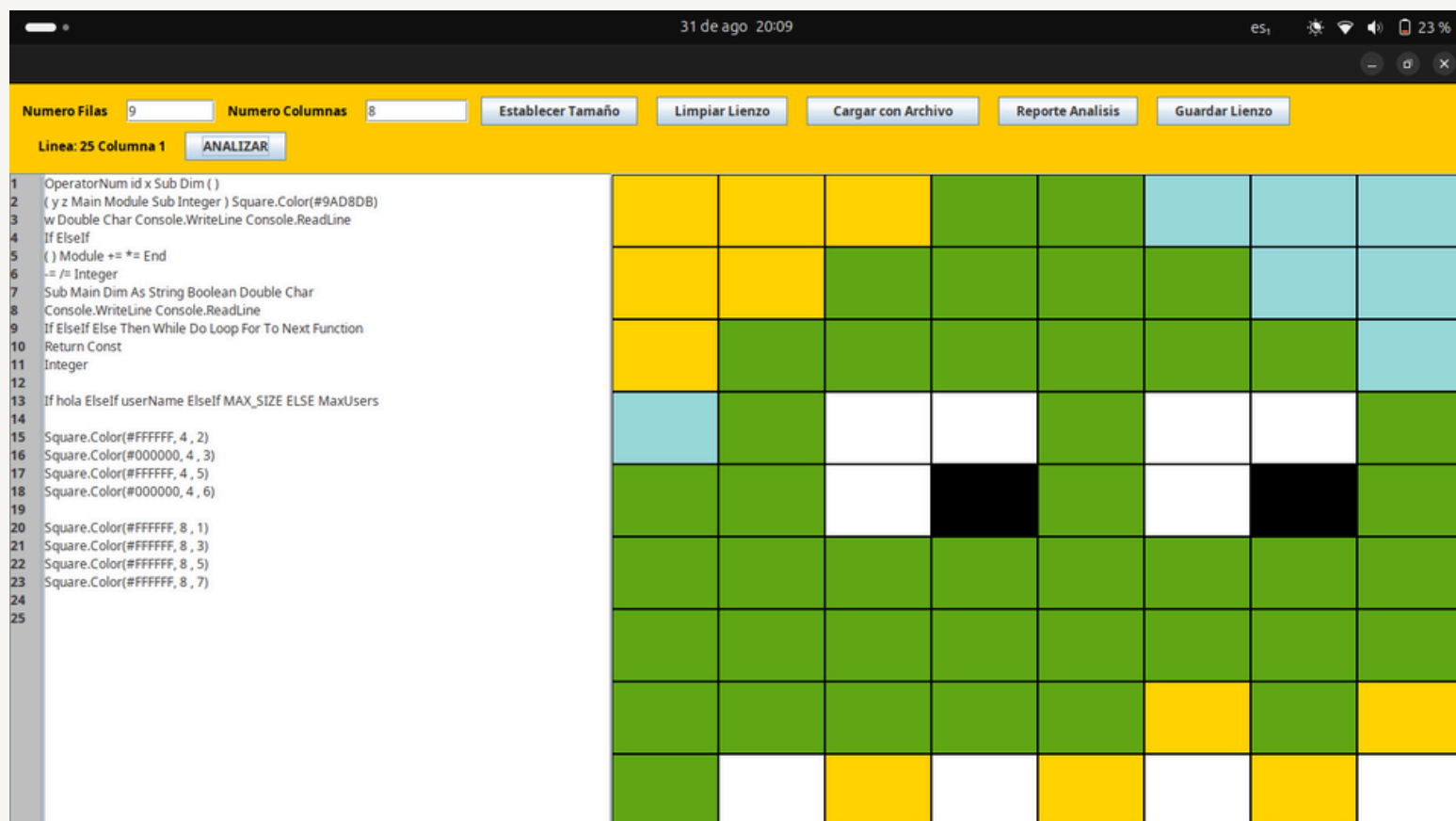
Numero Filas	<input type="text" value="3"/>	Numero Columnas	<input type="text" value="3"/>	Establecer T
Linea: 25 Columna 1		ANALIZAR		

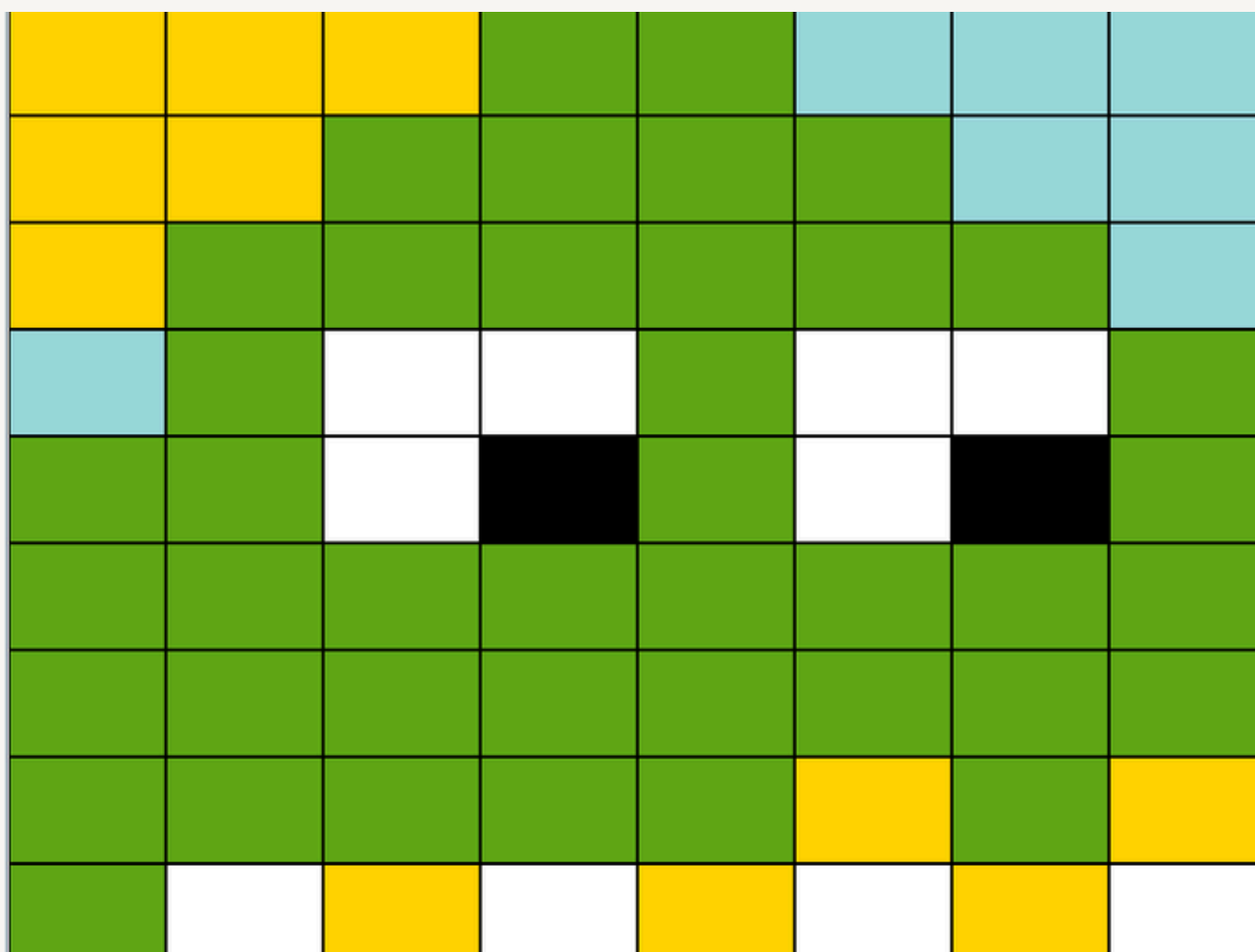
```
1 OperatorNum id x Sub Dim ( )
2 ( y z Main Module Sub Integer ) Square.Color(#9AD8DB)
3 w Double Char Console.WriteLine Console.ReadLine
4 If ElseIf
5 ( ) Module += *= End
6 -= /= Integer
7 Sub Main Dim As String Boolean Double Char
8 Console.WriteLine Console.ReadLine
9 If ElseIf Else Then While Do Loop For To Next Function
10 Return Const
11 Integer
12
13 If hola ElseIf userName ElseIf MAX_SIZE ELSE MaxUsers
14
15 Square.Color(#FFFFFF, 4 , 2)
16 Square.Color(#000000, 4 , 3)
17 Square.Color(#FFFFFF, 4 , 5)
18 Square.Color(#000000, 4 , 6)
19
```



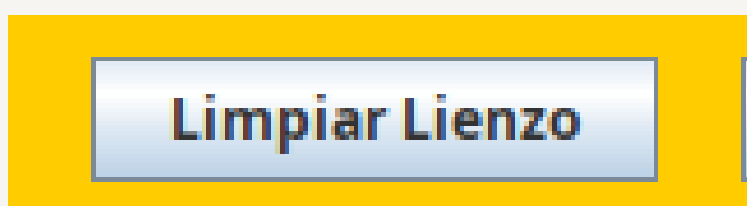
# Resultado de Analisis

Al presionar el boton de analisis se obtiene el resultado de analisis, mostrando con cada tipo de palabra representando un color especifico en un panel de color





Existe tambien un boton en el cual se puede limpiar el lienzo para realizar un nuevo analisis



# Generacion De Reportes

La interfaz presenta una sección de generación de reportes que muestra un análisis detallado de los paneles de colores generados en la aplicación. Esta sección se divide en tres partes principales:

Tabla de "Tokens Encontrados": Muestra un listado con las siguientes columnas:

Token: Identifica palabras reservadas, variables, o colores que han sido detectados durante el análisis.

Lexema: Representa el contenido o el valor específico del token identificado.

Fila y Columna: Indican la posición en términos de fila y columna del token en el documento o área analizada.

Fila Cuadro y Columna Cuadro: Definen la ubicación específica del cuadro correspondiente al token en la cuadrícula generada.

Color: Muestra el código hexadecimal del color asociado con el token, como #60A917 para verde o #FFD300 para amarillo.

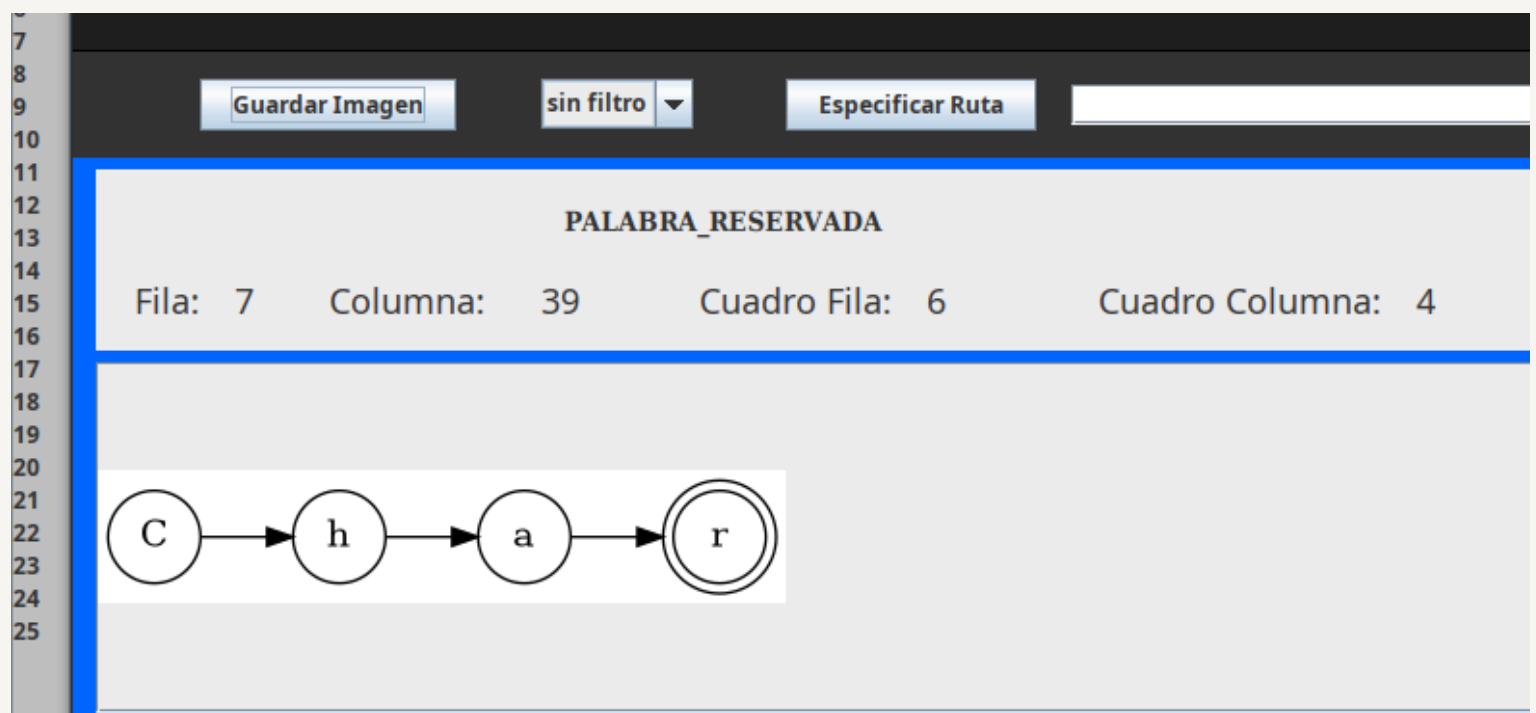
## TOKENS ENCONTRADOS

## ORDER

Columna	Fila Cuadro	Columna Cuadro	Color
5	5	2	#60A917
1	5	3	#FFFFFF
1	5	4	#000000
10	5	5	#60A917
1	5	6	#FFFFFF
1	5	7	#000000
14	5	8	#60A917
17	6	1	#60A917
24	6	2	#60A917
32	6	3	#60A917
39	6	4	#60A917
1	6	5	#60A917
19	6	6	#60A917
1	6	7	#60A917
4	6	8	#60A917
11	7	1	#60A917
16	7	2	#60A917

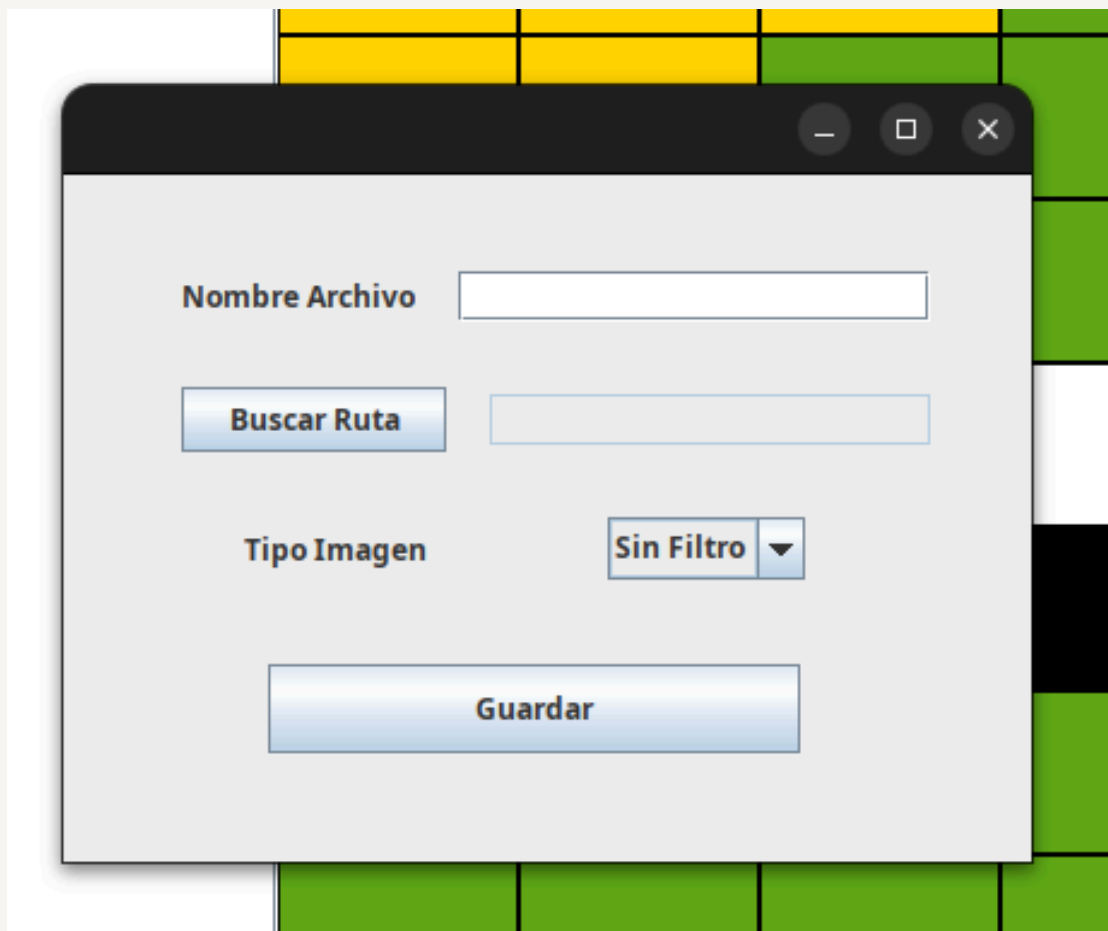
# Informacion Token

La ventana también ofrece opciones adicionales como "Guardar Imagen" para capturar la información visual generada, filtros para aplicar sobre los datos mostrados y un campo para especificar la ruta de almacenamiento. Esta funcionalidad es útil para analizar a fondo cada elemento del código, proporcionando una herramienta visual y detallada para entender la estructura del archivo analizado.



# Generacion de Lienzo

Esta funcionalidad es útil para almacenar y documentar el resultado del análisis o diseño realizado en la aplicación, permitiendo a los usuarios conservar una copia visual del trabajo para futuras referencias, informes, o presentaciones.

A screenshot of a software dialog box titled "Generacion de Lienzo". The dialog box has a light gray background and a black title bar with standard window controls (minimize, maximize, close). It contains the following elements: a text input field labeled "Nombre Archivo"; a blue button labeled "Buscar Ruta" next to an empty text input field; a label "Tipo Imagen" next to a dropdown menu currently showing "Sin Filtro"; and a large blue button labeled "Guardar" at the bottom. The dialog box is overlaid on a background of a grid with yellow and green cells.

# Tipos de Datos

**Identificadores:** Son nombres creados por el usuario para variables, funciones y otros elementos. Deben comenzar con una letra y pueden incluir letras, números y guiones bajos, pero no pueden comenzar con un número.

**Operadores:**

**Aritméticos:** Realizan cálculos matemáticos, como suma (+), resta (-), multiplicación (\*), división (/), exponente (^), y módulo (Mod).

**Relacionales:** Comparan valores, usando símbolos como igual (==), diferente (<>), mayor que (>), menor que (<), mayor o igual que (>=), y menor o igual que (<=).

**Lógicos:** Ejecutan operaciones de lógica como "y" (And), "o" (Or) y "no" (Not).

**Asignación:** Se usan para asignar valores a variables, como =, +=, -=, \*=, /=.

**Palabras reservadas:** Son palabras específicas del lenguaje de programación que no se pueden usar como identificadores, como Module, Sub, Main, If, Else, Loop, Function, entre otras.

**Tipos de datos:** Indican los diferentes tipos de valores que puede manejar el programa, tales como enteros (números como 0, 1, 2), decimales (2.5, 32.02), cadenas de texto ("texto"), booleanos (True, False) y caracteres individuales ('A', '2', '\$').

**Square.Color:** Es un token especial que sirve para pintar un cuadro en el lienzo con un color específico, determinado por un código hexadecimal. Por ejemplo, `Square.Color(#FFFFFF, 4, 2)` pintará de blanco el cuadro ubicado en la fila 4, columna 2.

**Comentarios:** Son textos dentro del código que el compilador no ejecuta, usados para documentar o explicar el código. Solo se aceptan comentarios de una línea, que comienzan con el símbolo `'` (por ejemplo, `'este es un comentario'`).

**Signos y símbolos:** Son caracteres que se usan para estructurar el código, como paréntesis `( )`, llaves `{ }`, corchetes `[ ]`, comas `,`, y puntos `.`.