

# **Manual de Usuario**

## *Compilador LED*

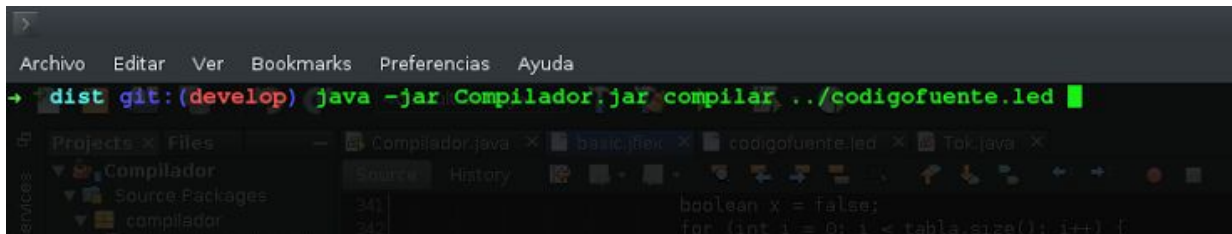
Autores:

García Robles, Carlos Adan

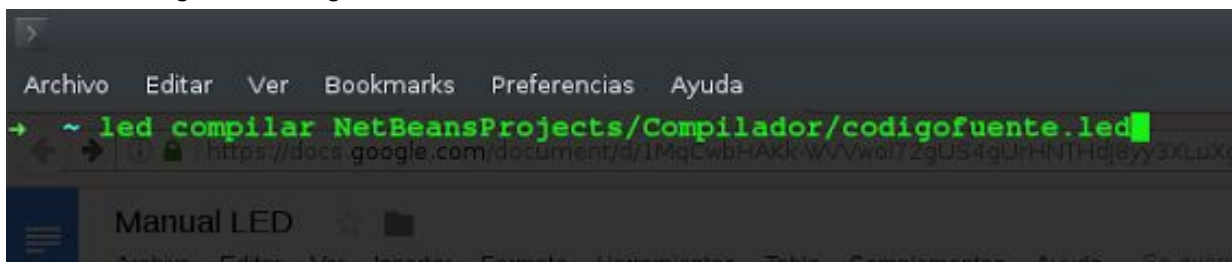
Juárez Amézquita, Melvin Abisaí

# 1. ¿Cómo ejecutarlo?

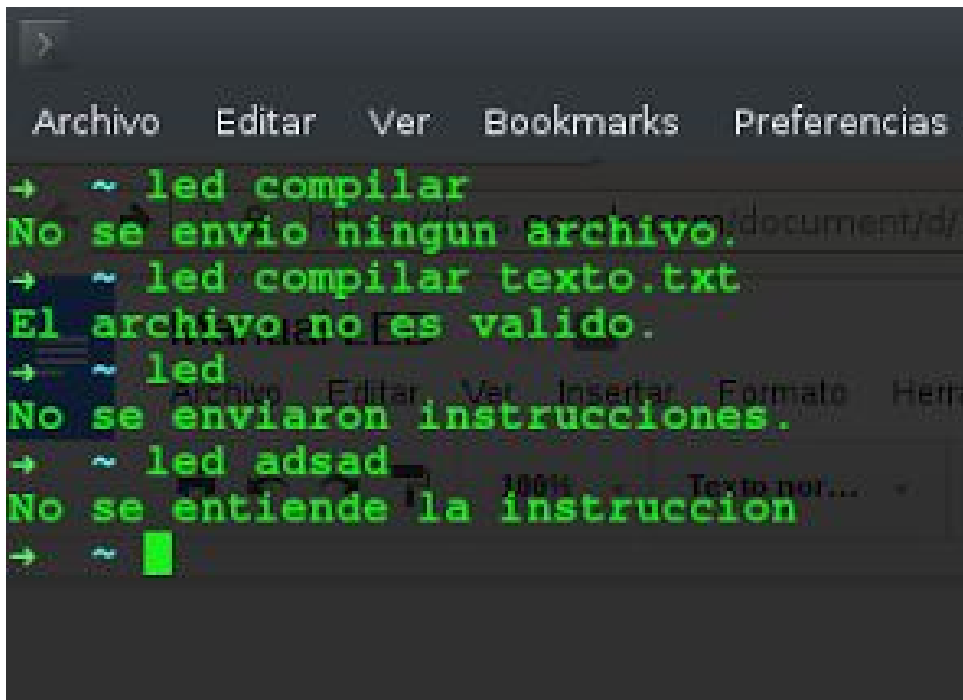
Desde la línea de comandos se escriben la directiva “compilar” y la ruta del archivo.



El ejemplo utiliza la ruta relativa, pero se puede utilizar también la ruta absoluta del archivo, como en la siguiente imagen.

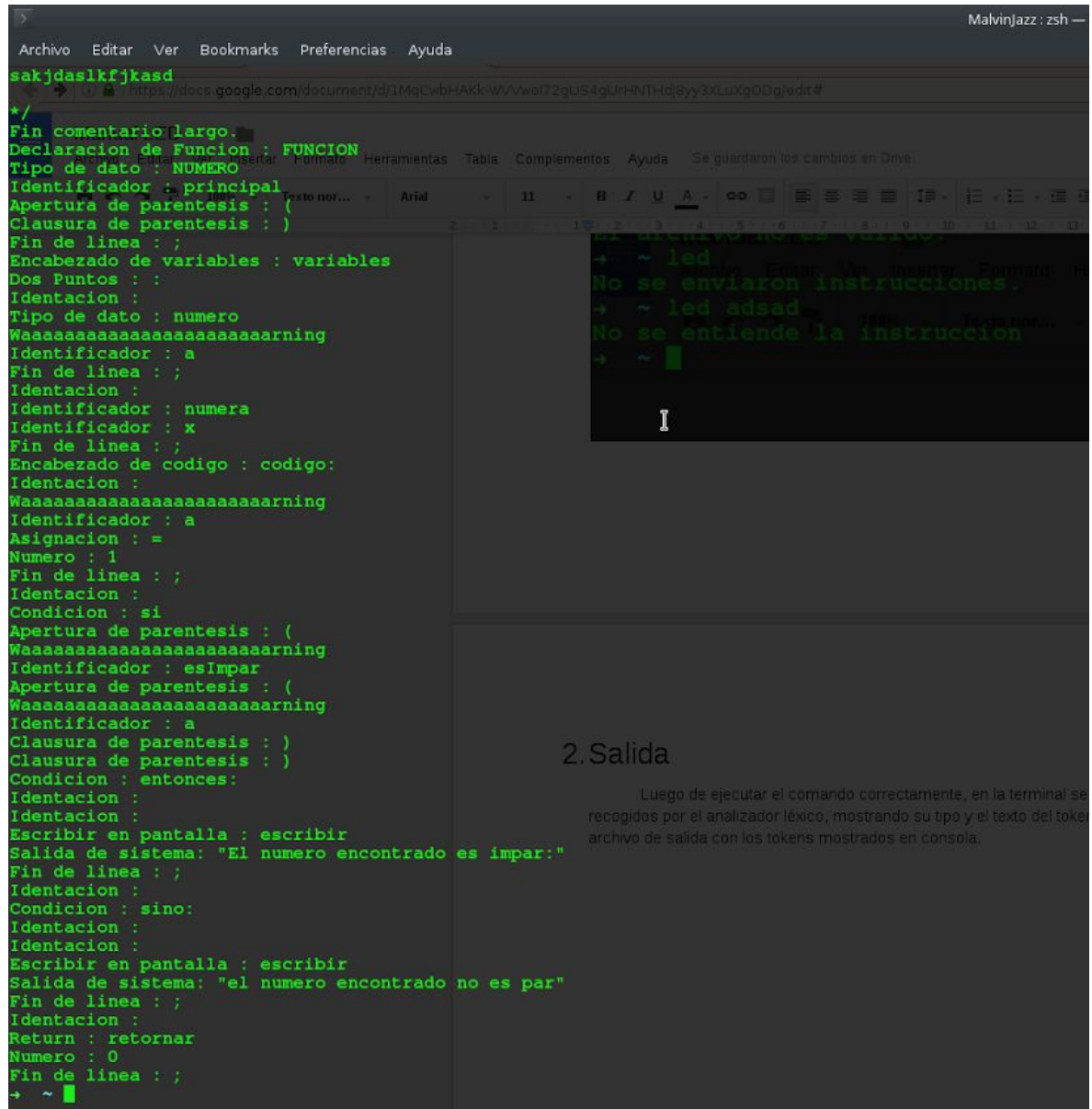


De no ser así, nos va a mostrar algunos errores en la salida de la terminal.



## 2. Salida

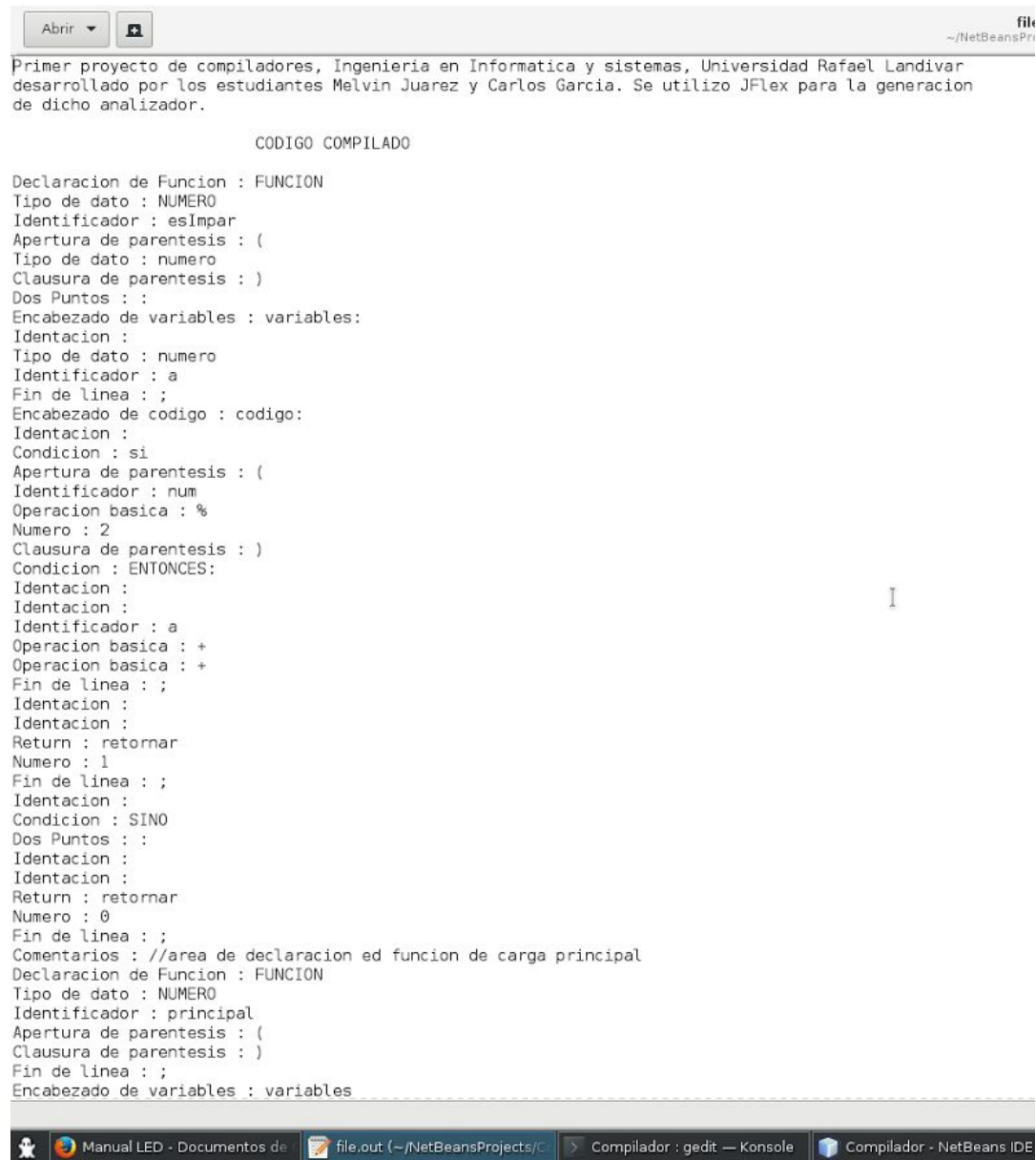
Luego de ejecutar el comando correctamente, en la terminal se muestran los tokens recogidos por el analizador léxico, mostrando su tipo y el texto del token. Además, escribe un archivo de salida con los tokens mostrados en consola.



The screenshot shows a code editor with a C program and its terminal output. The program is a lexical analyzer that processes the input string "sakjdaslkfjkasd". It identifies tokens such as "s", "a", "k", "j", "d", "a", "s", "l", "k", "f", "j", "k", "a", "s", "d". The terminal output shows the tokens and their types, along with the program's output: "El numero encontrado es impar:" and "el numero encontrado no es par".

```
sakjdaslkfjkasd
*/
Fin comentario: largo.
Declaracion de Funcion : FUNCION
Tipo de dato : NUMERO
Identificador : principal
Apertura de parentesis : (
Clausura de parentesis : )
Fin de linea : ;
Encabezado de variables : variables
Dos Puntos : :
Identacion :
Tipo de dato : numero
Waaaaaaaaaaaaaaaaaaaaaarning
Identificador : a
Fin de linea : ;
Identacion :
Identificador : numera
Identificador : x
Fin de linea : ;
Encabezado de codigo : codigo:
Identacion :
Waaaaaaaaaaaaaaaaaaaaaarning
Identificador : a
Asignacion : =
Numero : 1
Fin de linea : ;
Identacion :
Condicion : si
Apertura de parentesis : (
Waaaaaaaaaaaaaaaaaaaaaarning
Identificador : esImpar
Apertura de parentesis : (
Waaaaaaaaaaaaaaaaaaaaaarning
Identificador : a
Clausura de parentesis : )
Clausura de parentesis : )
Condicion : entonces:
Identacion :
Identacion :
Escribir en pantalla : escribir
Salida de sistema: "El numero encontrado es impar:"
Fin de linea : ;
Identacion :
Condicion : sino:
Identacion :
Identacion :
Escribir en pantalla : escribir
Salida de sistema: "el numero encontrado no es par"
Fin de linea : ;
Identacion :
Return : retornar
Numero : 0
Fin de linea : ;
+ ~ led
No se enviaron instrucciones.
+ ~ led adsad
No se entiende la instruccion
+ ~
```

El archivo de salida contiene la lista de tokens encontrados, de manera limpia y ordenada. Dicho archivo se guarda junto al archivo compilado.



The screenshot shows a NetBeans IDE window with a single file open. The file's title bar is "file.out (~/.NetBeansProjects/C...". The editor area contains the following text:

```
Primer proyecto de compiladores, Ingenieria en Informatica y sistemas, Universidad Rafael Landivar
desarrollado por los estudiantes Melvin Juarez y Carlos Garcia. Se utilizo JFlex para la generacion
de dicho analizador.

CODIGO COMPILADO

Declaracion de Funcion : FUNCION
Tipo de dato : NUMERO
Identificador : esImpar
Apertura de parentesis : (
Tipo de dato : numero
Clausura de parentesis : )
Dos Puntos : :
Encabezado de variables : variables:
Identacion :
Tipo de dato : numero
Identificador : a
Fin de linea : ;
Encabezado de codigo : codigo:
Identacion :
Condicion : si
Apertura de parentesis : (
Identificador : num
Operacion basica : %
Numero : 2
Clausura de parentesis : )
Condicion : ENTONCES:
Identacion :
Identacion :
Identificador : a
Operacion basica : +
Operacion basica : +
Fin de linea : ;
Identacion :
Identacion :
Return : retornar
Numero : 1
Fin de linea : ;
Identacion :
Condicion : SINO
Dos Puntos : :
Identacion :
Identacion :
Return : retornar
Numero : 0
Fin de linea : ;
Comentarios : //area de declaracion ed funcion de carga principal
Declaracion de Funcion : FUNCION
Tipo de dato : NUMERO
Identificador : principal
Apertura de parentesis : (
Clausura de parentesis : )
Fin de linea : ;
Encabezado de variables : variables
```

The IDE's status bar at the bottom shows the following tabs: "Manual LED - Documentos de...", "file.out (~/.NetBeansProjects/C...", "Compilador : gedit — Konsole", and "Compilador - NetBeans IDE".

### 3. Segmento de datos

Al inicio de una función debe estar el segmento de datos, que se divide en dos:

- Sección de archivos
- Sección de variables

La sección de archivos es importante, pues el fin principal del intérprete es facilitar datos a partir de consultas a estos archivos.

Sintaxis	archivos: <id_archivo> <ruta_archivo[, ruta_archivo]>;
Parámetros	<ul style="list-style-type: none"><li>• id_archivo: Nombre que se le dará al archivo o a la lista de archivos.</li><li>• ruta_archivo: Ruta al archivo, puede ser específica o relativa, siempre debe ser una ruta a un archivo .ods, .xls o .xlsx.</li></ul>

Sintaxis	variables: <tipo> <id_variable[, id_variable [= valor]]>;
Parámetros	<ul style="list-style-type: none"><li>• tipo: El tipo puede ser numero, cadena o booleano.</li><li>• id_variable: Identificador de la variable.</li><li>• valor: Se puede inicializar la variable.</li></ul>

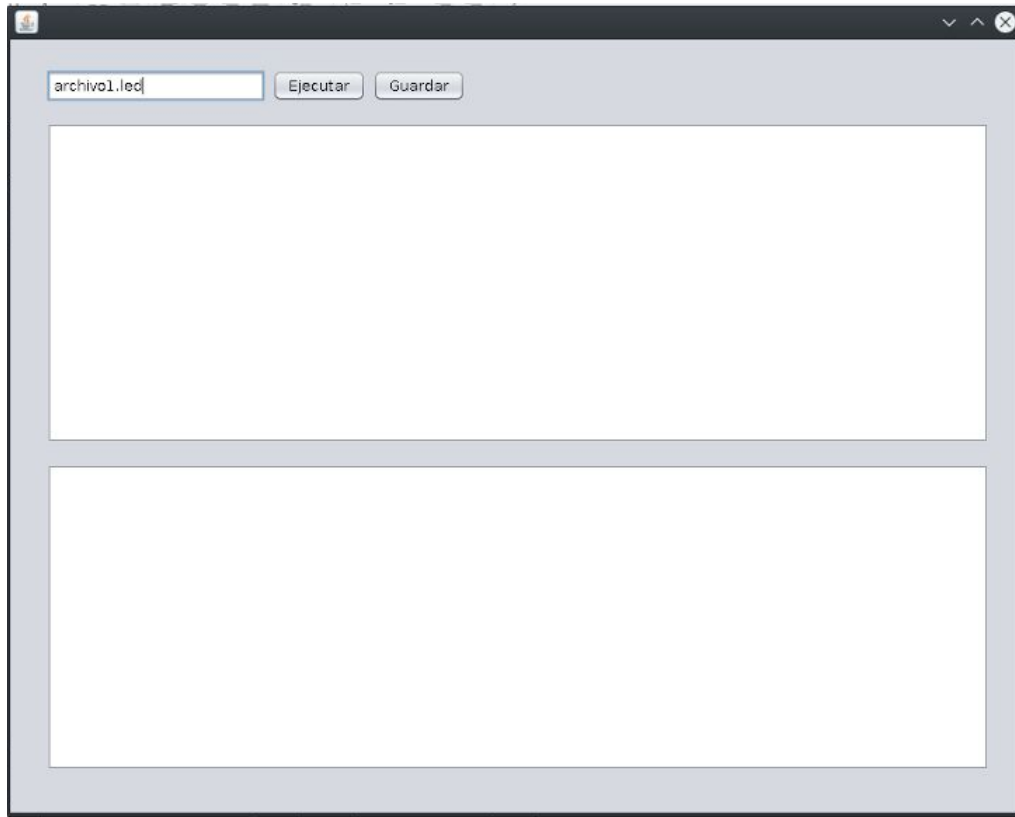
#### A. Métodos de archivos

Abrir archivo	
Sintaxis	abrir <id_archivo>;
Retorno	<ul style="list-style-type: none"><li>• Método, inicializa el archivo para poder leerlo sin dificultad.</li></ul>
Acceso a las celdas	
Sintaxis	<ol style="list-style-type: none"><li>1. &lt;id_archivo&gt;[fila, columna]</li><li>2. &lt;id_archivo&gt;[hoja, fila, columna]</li></ol>
Parámetros	<ul style="list-style-type: none"><li>• id_archivo: Identificador del archivo al que queremos acceder.</li><li>• fila: Número de fila a la que queremos acceder.</li><li>• columna: Número de columna a la que queremos acceder.</li><li>• hoja: Número de hoja a la que queremos acceder.<ul style="list-style-type: none"><li>○ Los tres anteriores se pueden sustituir por un '?', esto define que</li></ul></li></ul>

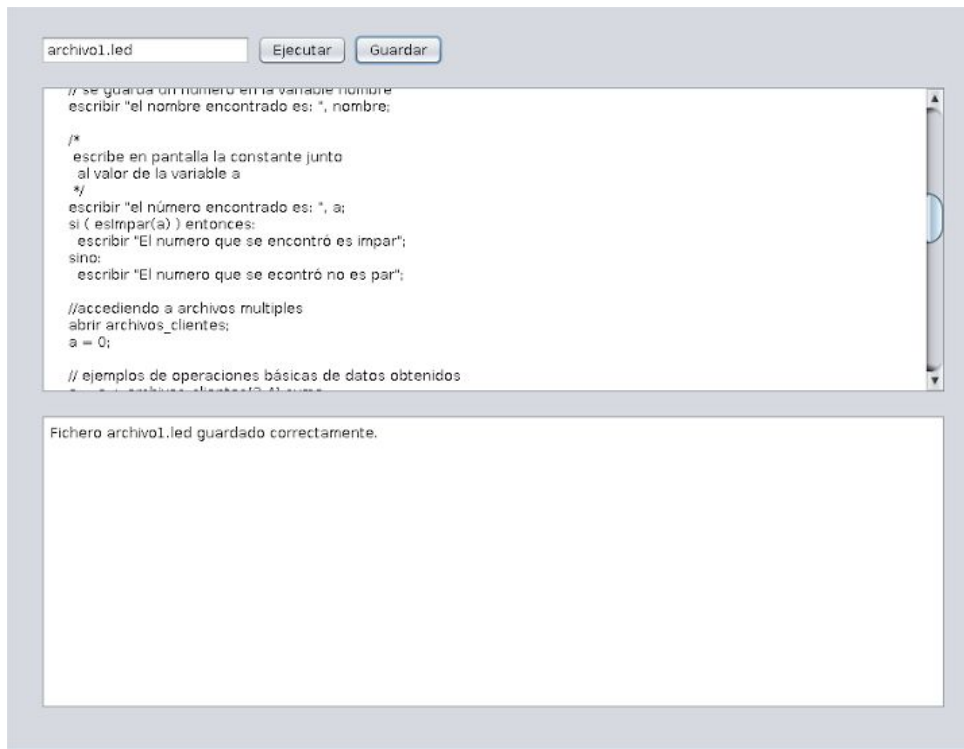
	queremos acceder a todas las celdas especificadas por los otros parámetros.
Retorno	<ol style="list-style-type: none"> <li>1. Valor de la celda, ya sea cadena, numero o booleano. <ol style="list-style-type: none"> <li>a. Si existe un '?', se retorna un arreglo de los valores obtenidos de las celdas</li> </ol> </li> </ol>

Operaciones de archivos	
Sintaxis	<id_archivo>[[hoja, ]fila, columna].<operacion>
Parámetros	<ul style="list-style-type: none"> <li>operacion: Función del archivo a ejecutar.</li> </ul>
Suma	
Retorno	<ul style="list-style-type: none"> <li>numero: suma de los valores ubicados en esas celdas.</li> </ul>
Resta	
Retorno	<ul style="list-style-type: none"> <li>numero: diferencia de los valores ubicados en esas celdas</li> </ul>
Promedio, Media	
Retorno	<ul style="list-style-type: none"> <li>numero: promedio de los valores de las celdas</li> </ul>
Moda	
Retorno	<ul style="list-style-type: none"> <li>numero: valor que más se repite en las celdas.</li> </ul>
Mayor	
Retorno	<ul style="list-style-type: none"> <li>numero: valor más alto encontrado.</li> </ul>
Menor	
Retorno	<ul style="list-style-type: none"> <li>numero: valor más bajo encontrado.</li> </ul>
<b>Errores</b>	<ul style="list-style-type: none"> <li>Dato Inválido: En algún archivo existe una celda con un valor no numérico.</li> </ul>
Concatenar	
Retorno	<ul style="list-style-type: none"> <li>cadena: concatenación de los valores de las celdas.</li> <li>Forma: "[fila, columna] -&gt; valor_celda".</li> </ul>

# IDE



La forma de utilizar el IDE es muy sencilla, en la caja de texto se escribe el nombre del archivo a compilar, luego se ejecuta.



En el primer área de texto se puede escribir el código y eventualmente ejecutarlo. Abajo se muestra un log del código.

Gramatica utilizada.

PROGRAMA -> COMENTARIOS PROGRAMA | DECLARACION\_FUNCIONES PROGRAMA  
| PREPROCESADOR PROGRAMA

COMENTARIOS -> Comentario | Comentario COMENTARIOS

PREPROCESADOR -> incluir Directiva | incluir Directiva PREPROCESADOR

DECLARACION\_FUNCIONES ->

funcion TokenTipos Identificador ParenL PARAMETROS ParenR DosPuntos CUERPO  
| funcion TokenTipos Identificador ParenL ParenR DosPuntos CUERPO

PARAMETROS -> TokenTipos Identificador | TokenTipos Identificador Coma PARAMETROS



CUERPO -> SEGARCHIVOS SEGVARIABLES SEGCODIGO  
| SEGVARIABLES SEGCODIGO  
| SEGCODIGO

SEGARCHIVOS -> TokenIdentacion TokenEFile DosPuntos ARCHIVOS

ARCHIVOS -> TokenIdentacion TokenIdentacion Identificador ArchivoNombre TokenFinLinea  
TokenSaltoLinea  
| TokenIdentacion TokenIdentacion ArchivoNombre TokenComas  
ARCHIVOS

SEGVARIABLES -> TokenIdentacion TokenEVariables DosPuntos DVAR

DVAR -> TokenIdentacion TokenIdentacion TokenTipos VARIABLES TokenFinLinea  
TokenSaltoLinea

VARIABLES -> Identificador | Identificador TokenComas VARIABLES

SEGCODIGO -> TokenIdentacion TokenECodigo DosPuntos CODIGO

CODIGO -> COMENTARIOS | CODELINE | SENTENCIAS

CODELINE -> ASIGNACION TokenFinLinea TokenSaltoLinea CODELINE  
| OPERACION TokenFinLinea TokenSaltoLinea CODELINE  
| ACCESO TokenFinLinea TokenSaltoLinea CODELINE  
| ABRIR TokenFinLinea TokenSaltoLinea CODELINE  
| ESCRIBIR TokenFinLinea TokenSaltoLinea CODELINE  
| LLAMADA TokenFinLinea TokenSaltoLinea CODELINE  
| RETORNO TokenFinLinea TokenSaltoLinea CODELINE

ASIGNACION -> Identificador TokenIgual OPERACION  
| Identificador Token Igual VARIABLE  
| Identificador TokenIgual TokenResta VARIABLE

ACCESO -> Identificador KeyL CELDAS KeyR  
| Identificador KeyL CELDAS KeyR TokenPunto Opeestadistica

CELDAS -> KeyL Decimal TokenComa Decimal KeyR

| KeyL Decimal TokenComa Decimal TokenComa Decimal KeyR T

| KeyL TokenAns TokenComa Decimal KeyR

| KeyL Decimal TokenComa TokenAns KeyR

| KeyL TokenAns TokenComa Decimal TokenComa Decimal KeyR

| KeyL Decimal TokenComa TokenAns TokenComa Decimal KeyR

| KeyL Decimal TokenComa Decimal TokenComa TokenAns KeyR

ABRIR -> TokenOpen Identificador TokenFinLinea TokenSaltoLinea

OPERACIONBASICA -> VARIABLE TokenOperacion VARIABLE

| VARIABLE TokenOperacion OPERACIONBASICA

OPERACION -> OPERACIONBASICA

| VARIABLE TokenOperacion OPERACIONBASICA

VARIABLE -> Identificador | Decimal | Identificador KeyL CELDAS KeyR

ABRIR -> TokenOpen Identificador

ESCRIBIR -> TokenPrint TokenSout SYSTEMOUT

| TokenPrint TokenSout

SYSTEMOUT -> VARIABLE | VARIABLE TokenComa SYSTEMOUT

LLAMADA -> REFERENCIAFUNC

REFERENCIAFUNC -> Identificador ParenL PARAMETROS ParenR

RETORNO -> TokenReturn VARIABLE

SELECTIVAS ->

TokenIf ParenL CONDICION ParenR TokenThen DosPuntos TokenSaltoLinea

CODIGO

| TokenIf ParenL CONDICION ParenR TokenThen DosPuntos TokenSaltoLinea

CODIGO TokenElse CODIGO

CONDICION -> VARIABLE Condicionales VARIABLE

El código del programa puede encontrarse y descargarse en el repositorio de GitHub:

<https://github.com/MalvinJazz/Compilador>

- Lenguaje Utilizado: Java
- Ide Utilizado para el desarrollo del analizador: NetBeans 8.1
- JDK 1.8
- Librería JFlex para la construcción del analizador.
- Librería Apache POI, para lectura de excel.

Copyright (C) 2016 cgarciarobles <[carlosfipq@gmail.com](mailto:carlosfipq@gmail.com)>, MalvinJazz <[malvinjazz@live.com](mailto:malvinjazz@live.com)>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.