

Universidad de San Carlos de Guatemala
Ingeniería en Ciencias y Sistemas
Organización de lenguajes de compilador 1

Manual de Usuario DATAFORGE

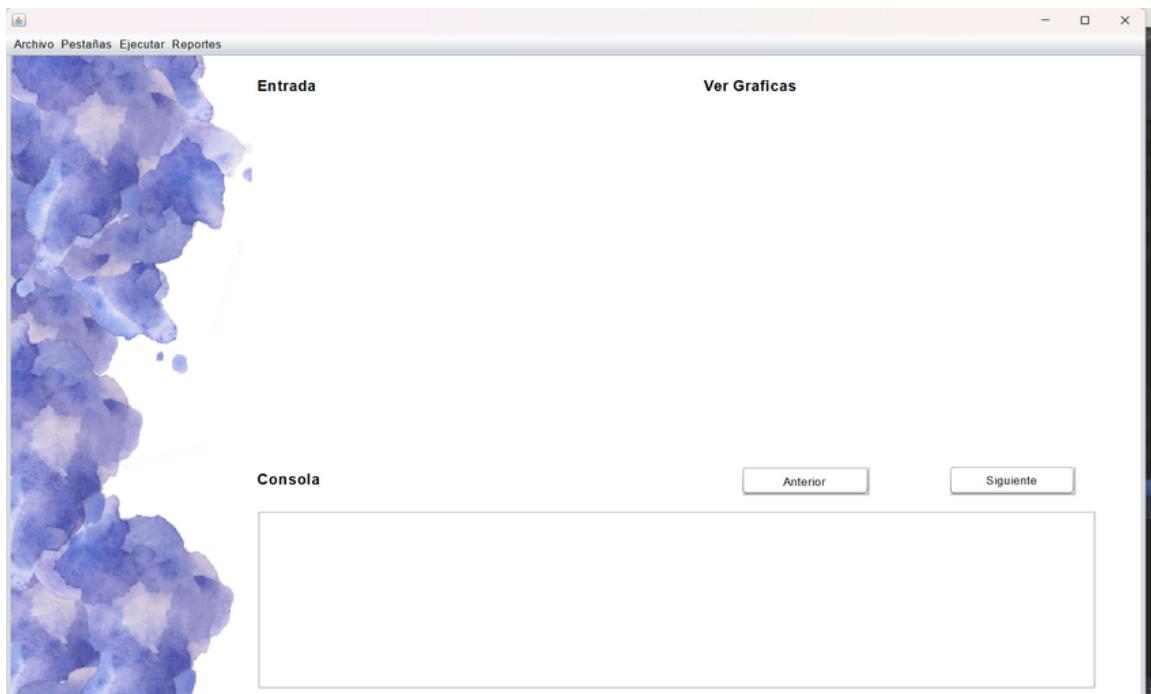
María Patricia Serrano Ramírez
202201689
09 de Marzo del 2024

Manual de Usuario

A continuación se detallara el funcionamiento así como los requisitos de la aplicación y así obtener su desarrollo exitoso al ejecutarla

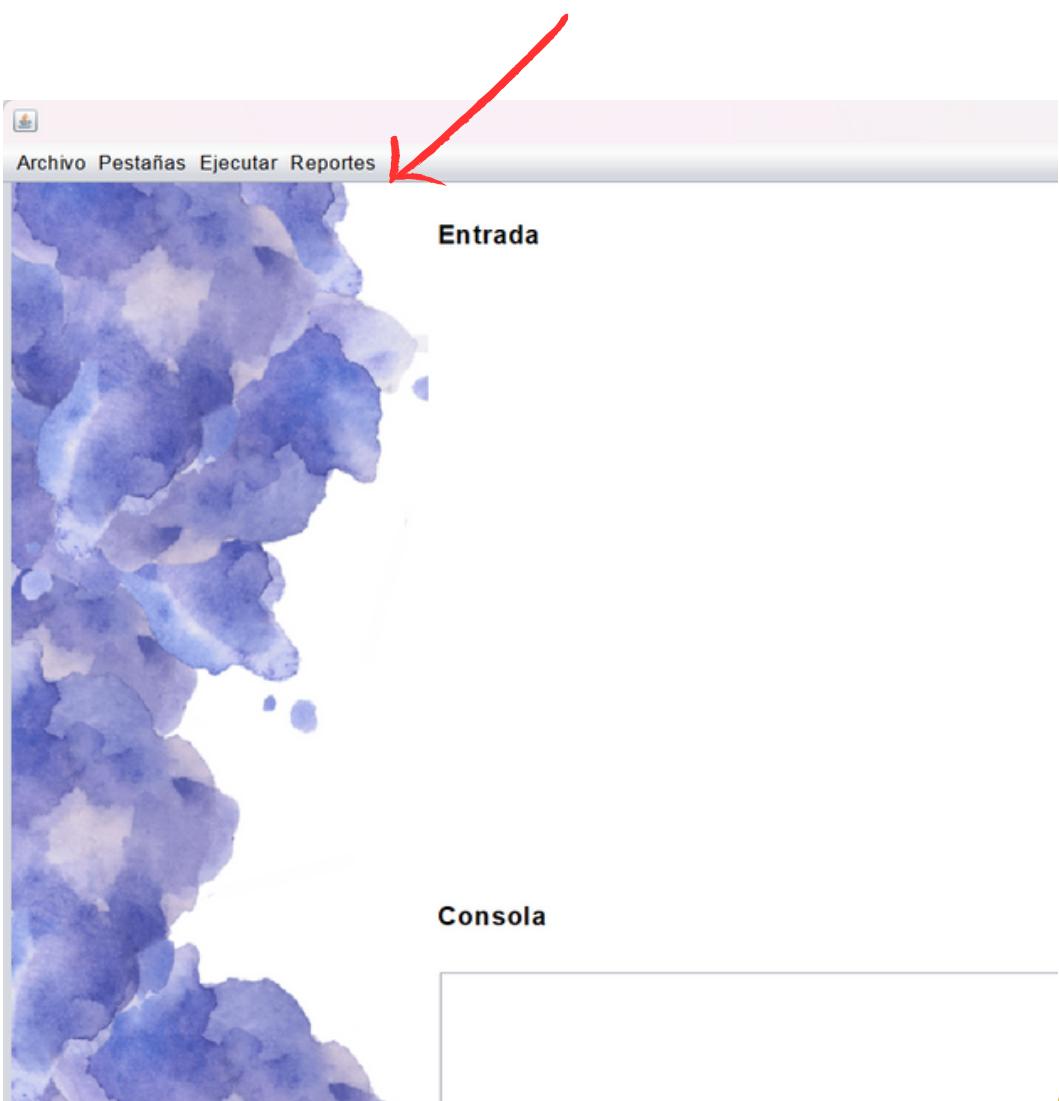
Ventana Principal:

Al ejecutar la aplicación primero se nos mostrara la ventana principal



BARRA DE MENU

En la parte superior podemos observar la Barra de Menú con distintas opciones como Archivo, Pestañas, Ejecutar, Reportes.



Archivo

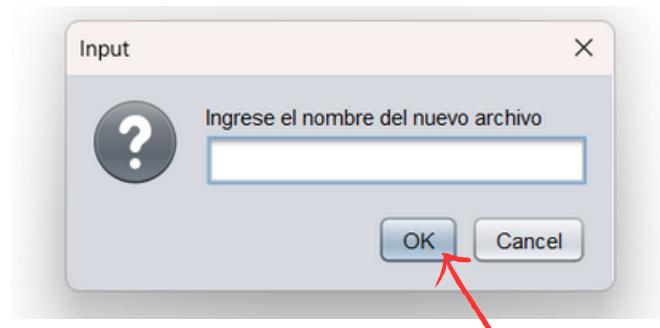
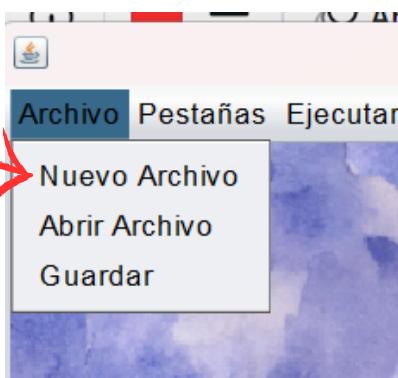
El menu archivo consta de tres opciones:

1. Nuevo Archivo: Permite crear nuevos archivos y abrirlos.
2. Abrir Archivo: Permite abrir archivos con la extensión .df
3. Guardar: Permite Guardar archivos nuevos y guardar los cambios realizados en un archivo.

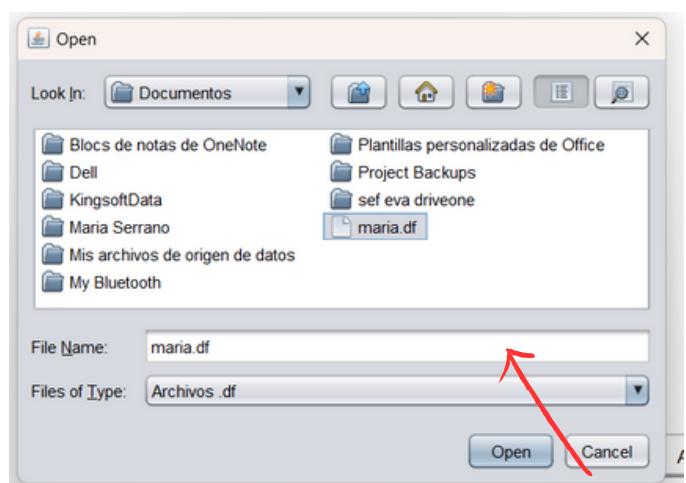
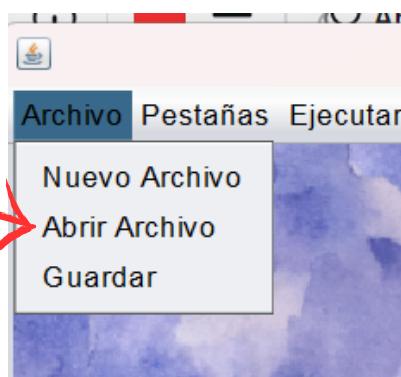


Archivo

1. Nuevo Archivo: solicita nombre del archivo.

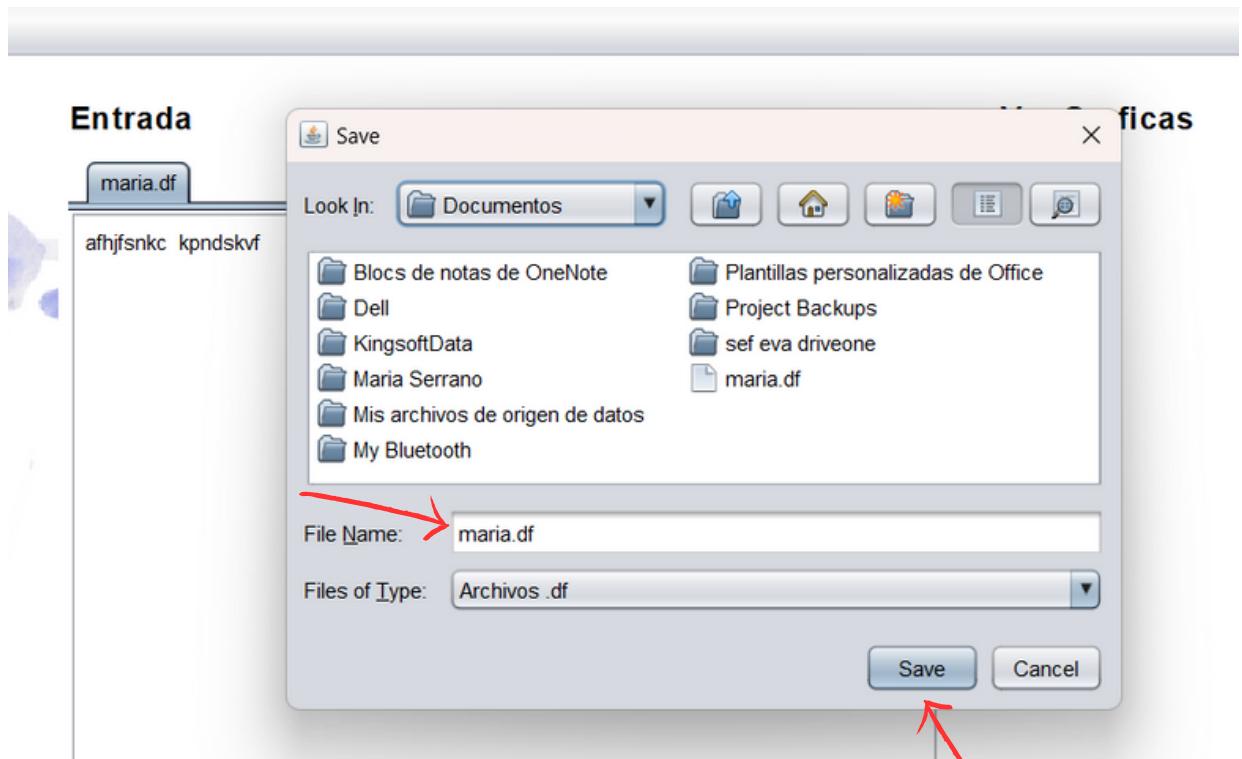
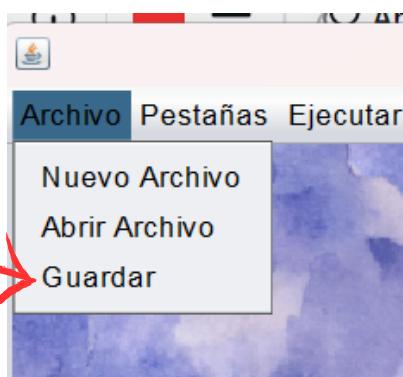


2. Abrir Archivo: seleccionamos el archivo



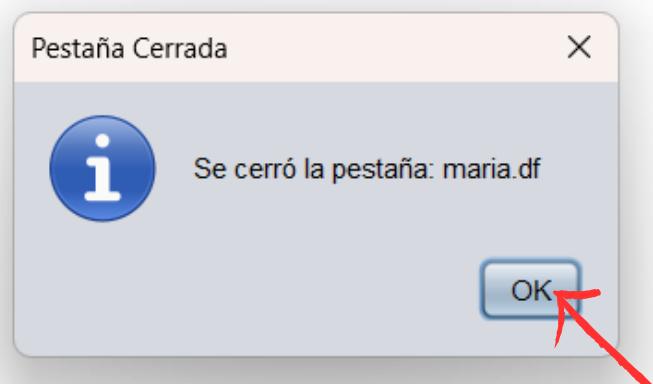
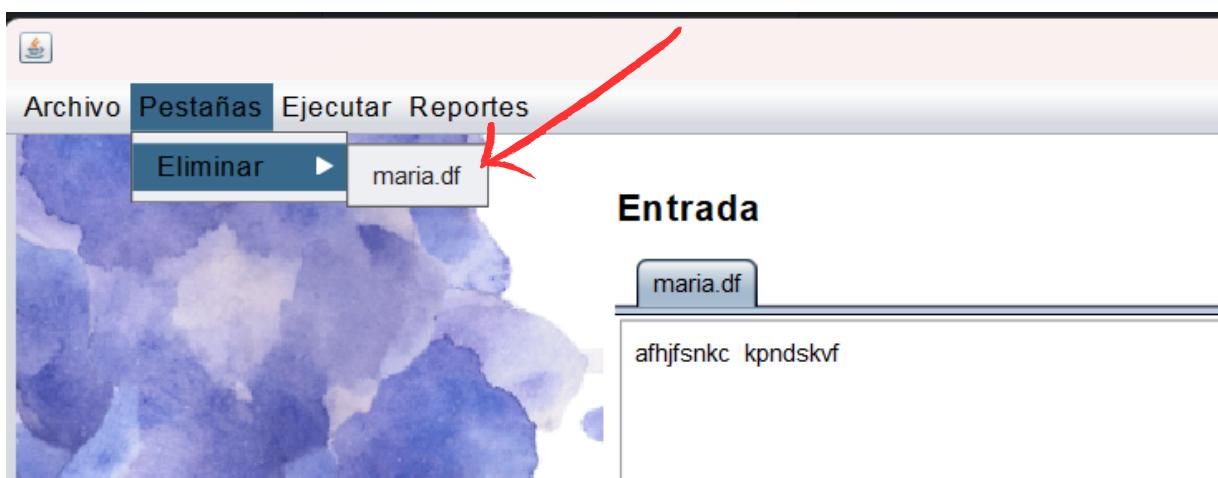
Archivo

1. Guardar: Permite el mismo nombre o se lo cambias.



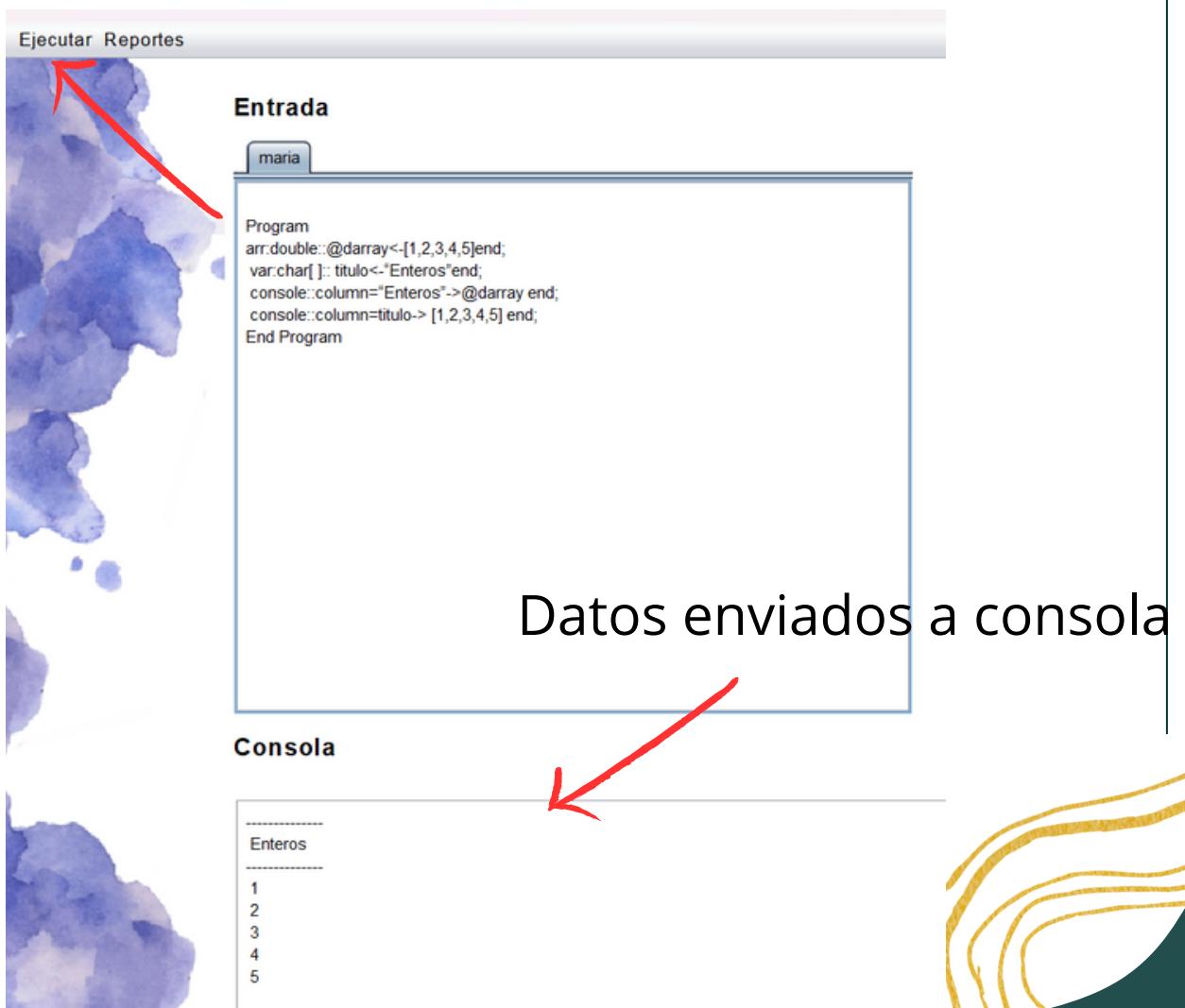
Pestañas

El menú pestañas tiene un sub menú eliminar que muestra el listado de las pestañas abiertas al darle clic puedes cerrar las pestañas.



Ejecutar

El menú Ejecutar este envía al analizador toda la cadena de texto que se encuentra en la pestaña abierta para analizarla y dependiendo de lo que envié el archivo se pueden mostrar impresiones en consola o se pueden generar graficas las cuales se mostraran es sus apartados



Ejecutar

Graficas solicitadas

Entrada

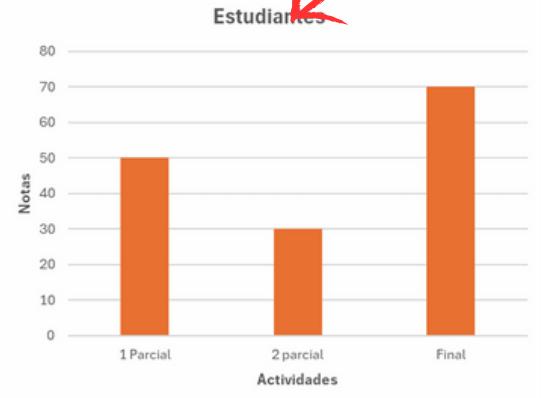
```
maria

Program
arr:double::@darray<-[1,2,3,4,5]end;
var:char[ ]:: titulo<-"Enteros"end;
console::column="Enteros">@darray end;
console::column=titulo-> [1,2,3,4,5] end;

graphBar(
titulo::char[] = "Estudiantes" end;
ejeX::char[] = ["1 Parcial", "2 parcial", "Final"] end;
ejeY::double = [50, 30, 70] end;
tituloX::char[] = "Actividades" end;
tituloY::char[] = "Notas" end;
EXEC graphBar end;
) end;

End Program
```

Ver Graficas



Consola

```
-----
Enteros
-----
1
2
3
4
5
```

Anterior

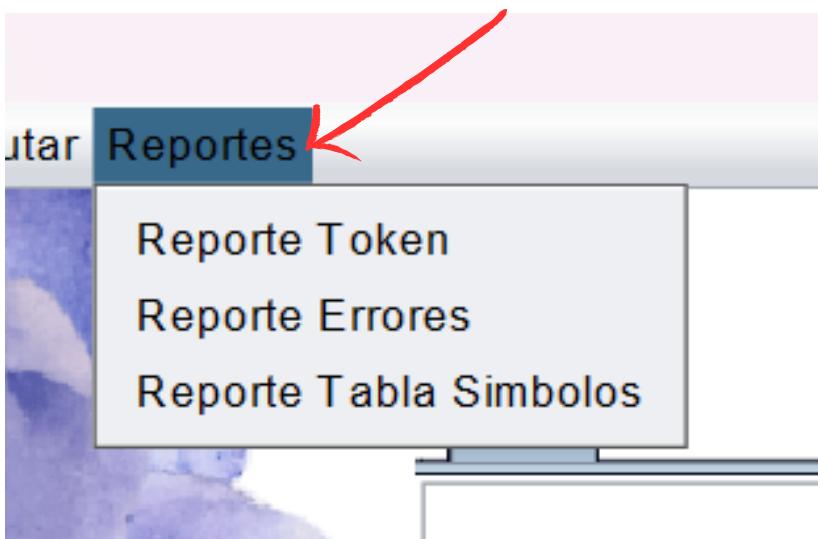
Siguiente

Datos enviados a consola

Reportes

El menú Reportes consta de tres opciones:

- Reporte Token: Muestra un reporte de los tokens analizados.
- Reporte Errores: Muestra los errores que se encontraron en la cadena enviada.
- Tabla Símbolos: Muestra los datos relevantes como ID, CADENAS Y NUEMROS.



Reportes

1. Reporte Token: al dar clic se abrirá automáticamente en el navegador la tabla de tokens.

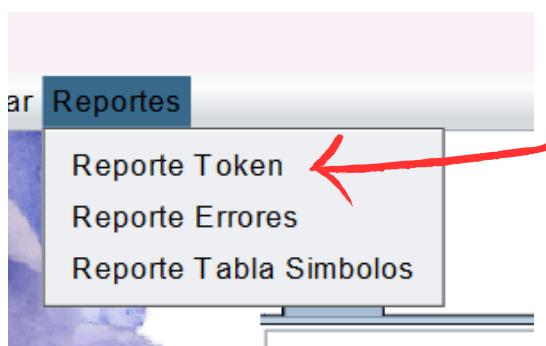


Tabla de Tokens

#	Lexema	Tipo	Línea	Columna
1	PROGRAM	RESERVADA	0	0
2	var	RESERVADA	1	0
3	:	SIMBOLO	1	4
4	double	RESERVADA	1	6
5	::	SIMBOLO	1	13
6	valio4	ID	1	16
7	<=	SIMBOLO	1	23
8	25.5	DOUBLE	1	26
9	end	RESERVADA	1	31
10	:	SIMBOLO	1	35

Reportes

1. Reporte Errores: al dar clic se abrirá automáticamente en el navegador la tabla de errores.

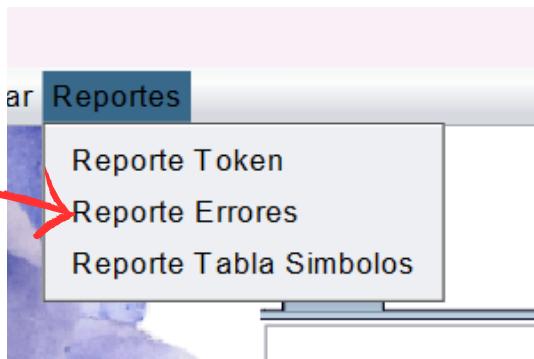


Tabla de Errores

ID	Tipo	Descripcion	Línea	Columna
1	Lexico	El caracter +no pertenece al lenguaje	8	0
2	Lexico	El caracter *no pertenece al lenguaje	11	41
3	Lexico	El caracter /no pertenece al lenguaje	14	24

Reportes

1. Tabla de Símbolos: al dar clic se abrirá automáticamente en el navegador la tabla de símbolos.

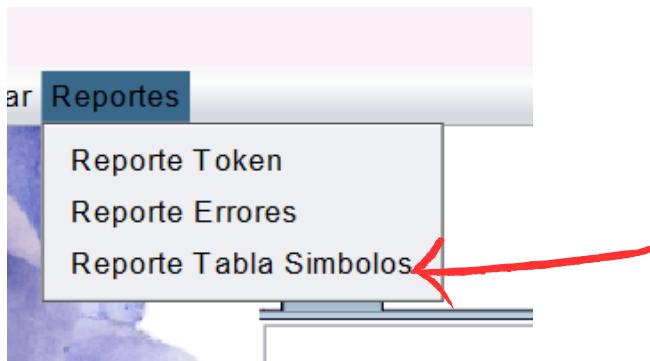


Tabla de Símbolos

#	Nombre	Tipo	Valor	Línea	Columna
1	número	variable double	3.1416	1	1
2	titulo	variable string	"Hola Mundo"	2	1
3	@arreglo1	arreglo string	["titulo1", "cadena2"]	3	1
4	@arreglo2	arreglo double	[1,2,3,4,5]	4	1

Estructura archivos

Los archivos que se cargan tienen una estructura especial para cada cosa a continuación un listado de ejemplos permitido.

Tipos de datos:

Tipo	Definición	Descripción	Ejemplo
Cadena	char[]	Es un grupo o conjunto de caracteres que pueden tener cualquier carácter, y este se encontrará delimitado por comillas dobles. “ ”	“cadena ejemplo”
Decimal	double	Este tipo de dato admite valores numéricos	0, 10, 0.5, 57.75, etc.

Encapsulado

```
PROGRAM  
<CÓDIGO>  
END PROGRAM
```

Declaración Variables

```
var:<TIPO>::<ID> <- <EXPRESION> end;  
! Ejemplos  
var:double:: numero <- 2.5 end;  
var:char[]::cadena <- "cadena" end;  
var:double:: copia <- numero end; ! copia tiene el valor 2.5
```

Estructura archivos

Declaración Arreglos

! Declaración de arreglos

```
arr:<TIPO>::@<ID> <- <LISTA_VALORES> end;
```

! Ejemplos

```
arr:double::@darray <- [1, 2, 3, 4, 5] end; ! Arreglo de tipo double
```

```
arr:char[]:@carray <- ["12", "2", "3"] end; ! Arreglo de tipo string
```

```
arr:double::@carray <- [numero, copia, 7] end; ! Puede usar variables
```

Operaciones

! Operaciones

```
var:double:: suma <- SUM(5, 2) end;
```

```
var:double:: resta <- RES(3, 2) end;
```

```
var:double:: multi <- MUL(4, numero) end; ! Funciona con variables
```

```
var:double:: division <- DIV(1, variable) end;
```

```
var:double:: modulo <- MOD(5, 4) end;
```

! Operaciones anidadas

```
var:double:: suma <- MUL( SUM(7,3) , RES(7, DIV(25,5) ) end;
```

```
arr:double::@darray <- [ SUM(7,3), DIV(25,5)] end; ! Arreglo con funciones
```

! se pueden ingresar el arreglo directamente o por variable

```
var:double:: med1 <- Media([1, 2, SUM(3, b), 4, a]) end;
```

```
var:double:: med2 <- Mediana( @arreglo ) end;
```

```
arr:double::@arreglo <- [Media(@data), Mediana(@data)] end;
```

! Tambien se pueden utilizar en operaciones aritmeticas

```
var:double:: mitad <- DIV( SUM(Max(@data), Min(@data) ), 2) end;
```

Estructura archivos

Imprimir expresiones

```
var:double:: numero <- 15 end;  
console::print = "hola", numero, 15, "adios" end;  
! Salida: hola, 15, adios
```

```
console::print = 1, 2, SUM(3,5), Media(@arreglo) end;  
! Salida: 1, 2, 8, 15.7
```

Imprimir arreglos

```
arr:double:@darray <- [1, 2, 3, 4, 5] end;  
var:char[ ]:: titulo <- "Enteros" end;  
console::column = "Enteros" -> @darray end;  
console::column = titulo -> [1, 2, 3, 4, 5] end;
```

Graficas

Barra

```
graphBar(  
    titulo::char[] = "Estudiantes" end;  
    ejeX::char[] = ["1 Parcial", "2 parcial", "Final"] end;  
    ejeY::double = [50, 30, 70] end;  
    tituloX::char[] = "Actividades" end;  
    tituloY::char[] = "Notas" end;  
    EXEC grapBar end;  
) end;
```

Estructura archivos

Pie

! Ejemplo

```
graphPie(  
    label::char[] = ["Uno", "Dos", "Tres"] end;  
    values::double = [50, 30, 20] end;  
    titulo::char[] = "Ejemplo Gráfica de Pie" end;  
    EXEC grapPie end;  
) end;
```

Line

```
graphLine(  
    titulo::char[] = "Gráfica de Línea" end;  
    ejeX::char[] = ["1 Parcial", "2 parcial", "Final"] end;  
    ejeY::double = [50, 30, 70] end;  
    tituloX::char[] = "Actividades" end;  
    tituloY::char[] = "Notas" end;  
    EXEC grapLine end;  
) end;
```

Histograma

```
Histogram(  
    titulo::char[] = "Analisis de Arreglo" end;  
    values::char[] = [2,2,2,5,5,7,8] end;  
    EXEC Histogram end;  
) end;
```