



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en ciencias de sistemas.

Manual de usuario

Nombre del estudiante

José Luis Espinoza Jolón - 202202182

Ing.(a). Kevin Lajpop

Guatemala, marzo 10 de 2024

Contenido

• Descripción	4
• Entorno de trabajo	5
1. Abrir Archivo	5
2. Nueva pestaña (+):	6
Otras funcionalidades:	6
▪ Guardar	6
▪ Eliminar	6
3. Ejecutar	7
4. Gráfica:	7
5. Reportes	7
▪ Reporte de Tokens:	7
▪ Reporte de errores	7
▪ Reporte de Tabla de Símbolos:	7
6. Area de entrada	7
7. Area de consola:	7
• Uso del lenguaje	8
Parámetros iniciales:	8
Declaración de variables:	9
Declaración de arreglo:	9
Operaciones Aritméticas	10
Funciones estadísticas:	10
Impresión en consola	11
▪ Grafica de barra	11

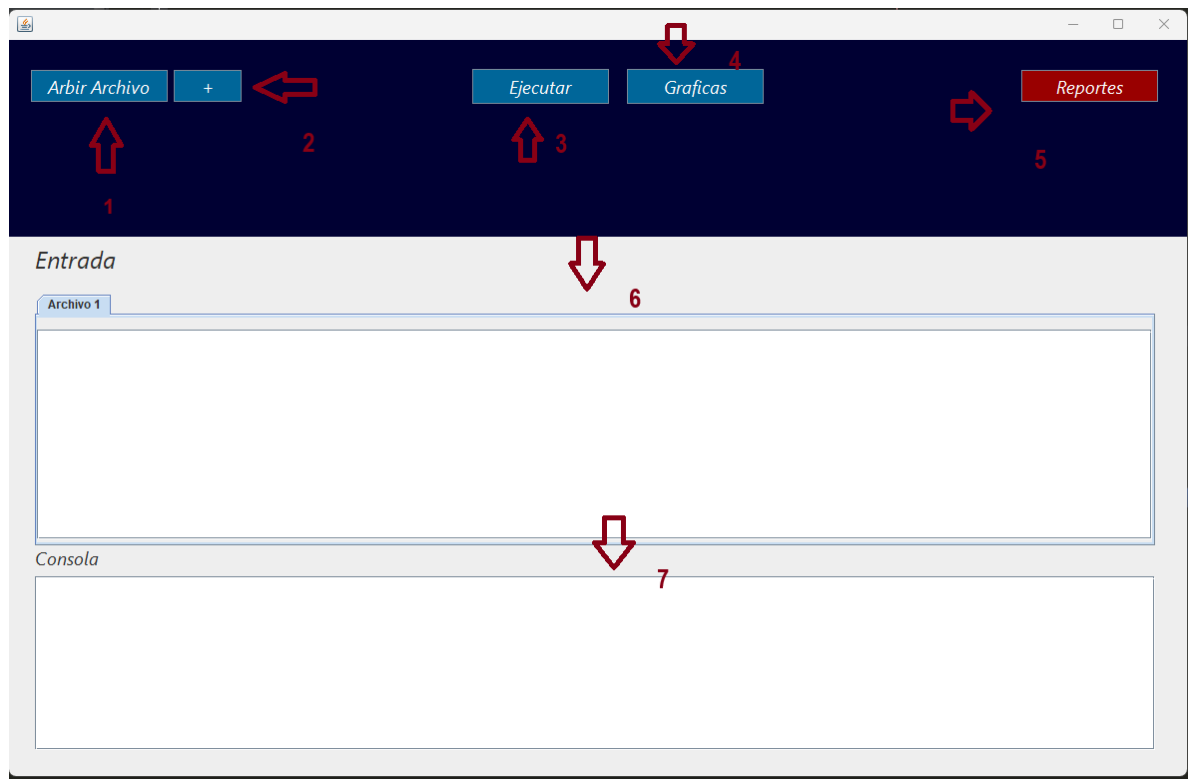
▪ Grafica de piel:	13
▪ Grafica de linea:	14
▪ Gráfica histograma:	15
Ejemplos de los reportes:	16

Descripción

El curso de Organización de Lenguajes y Compiladores 1, perteneciente a la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, requiere de usted, como conocedor de la construcción de analizadores Léxico y Sintáctico, crear un sistema que sea capaz de realizar operaciones aritméticas y estadísticas, además de poder generar diversos gráficos a partir de una colección de datos.

Entorno de trabajo

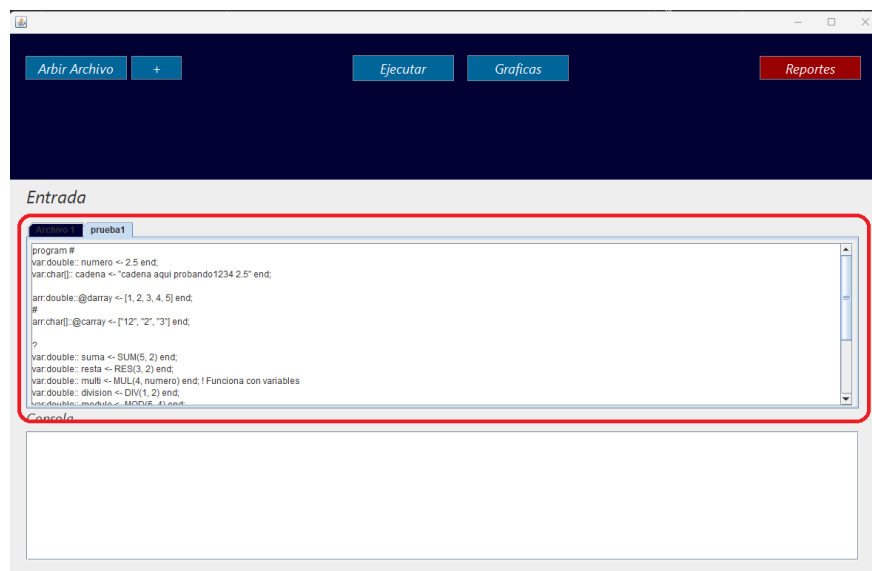
El usuario al ejecutar el programa visualizara la siguiente vista.



Las siguientes funciones que puede usar el usuario es el siguiente.

1. **Abrir Archivo:** El editor debe tener la capacidad de abrir archivos con las extensiones .df cuyo contenido se deberá mostrar en el área de entrada en una nueva pestaña con el nombre del archivo.

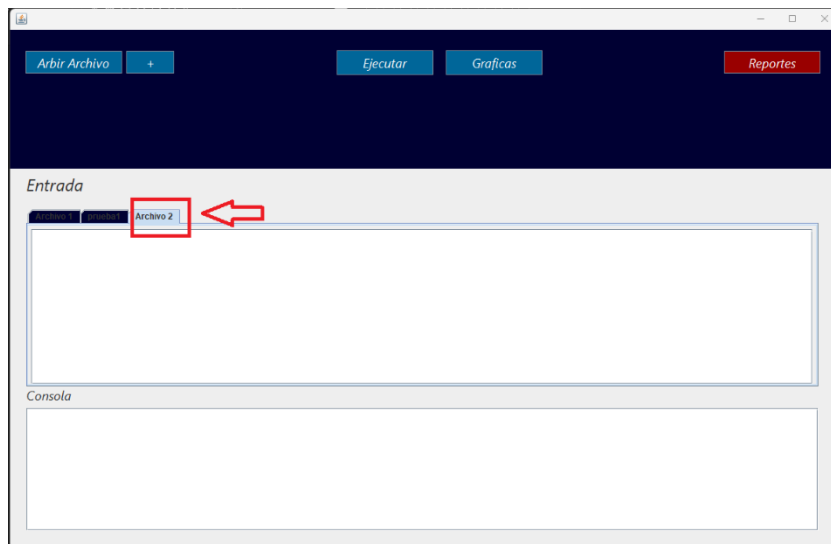
De la siguiente manera se visualizará un archivo cargado.



2. **Nueva pestaña (+):** El editor debe tener la capacidad de crear archivos en

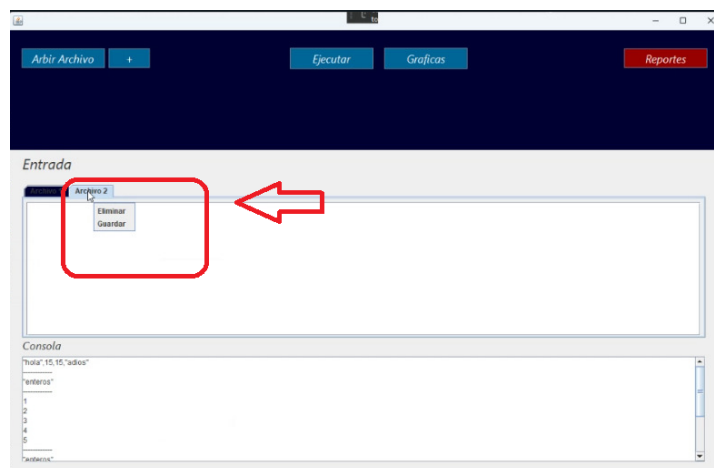
blanco el cual podrá ser editado en una pestaña que tiene el nombre del archivo, se puede crear múltiples.

De la siguiente manera se visualizará una nueva pestaña en la vista.



Otras funcionalidades:

- **Guardar:** El editor debe tener la capacidad de guardar el estado del archivo en el que se estará trabajando.
- **Eliminar:** Cada pestaña puede ser cerrada en cualquier momento. Si los cambios no se han guardado, se descartan.



3. **Ejecutar:** se envía la entrada de la pestaña actualmente seleccionada al intérprete con la finalidad de realizar el análisis, léxico, sintáctico y la ejecución de instrucciones.

4. **Gráfica:** Al darle clic se visualizará la vista donde se va a mostrar las gráficas generadas por el usuario.



Nota: los botones sirven para mostrar las demás gráficas, si no fuera solo una.

5. **Reportes:** Existe 3 tipos de reportes

- **Reporte de Tokens:** se mostrarán todos los tokens reconocidos por el analizador léxico.
- **Reporte de errores:** se mostrarán todos los errores léxicos y sintácticos encontrados.
- **Reporte de Tabla de Símbolos:** se mostrarán todas las variables y arreglos declarados.

6. **Area de entrada:** Se va a visualizar el código generado por el usuario.

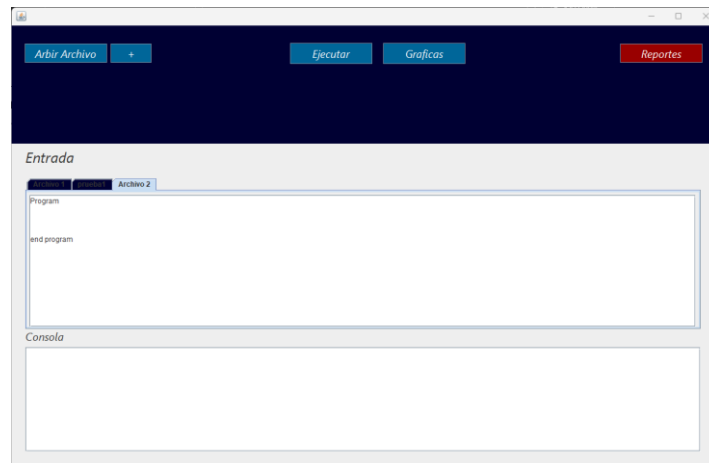
7. **Area de consola:**

En la consola de salida se mostrarán los resultados, mensajes y todo lo que sea indiciado en el lenguaje. Tiene como restricción el no ser editable por el usuario y únicamente puede mostrar información.

Uso del lenguaje

Parámetros iniciales:

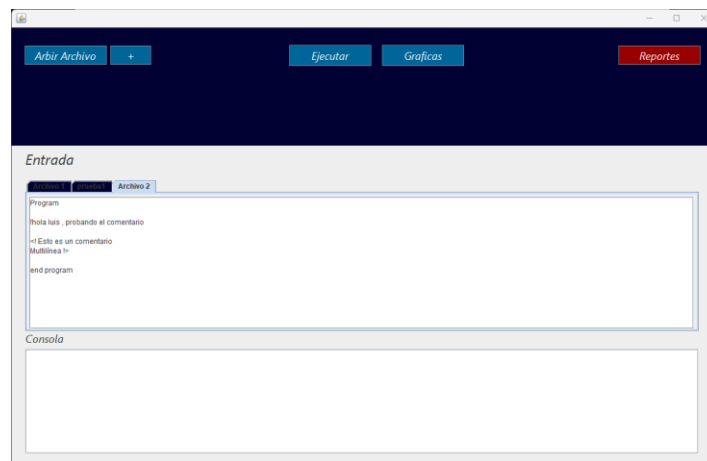
Todas las sentencias del lenguaje deben venir entre PROGRAM y END



visualización de los parámetros iniciales

Comentarios:

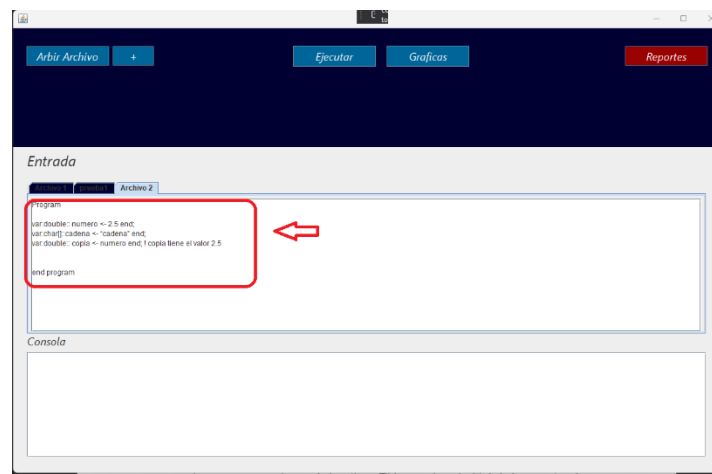
Existe 2 tipos de Comentarios, los comentarios les pueden ayudar al usuario a la hora de describir una parte de código o para que sirva y recordarse



Comentarios en el programa

Declaración de variables:

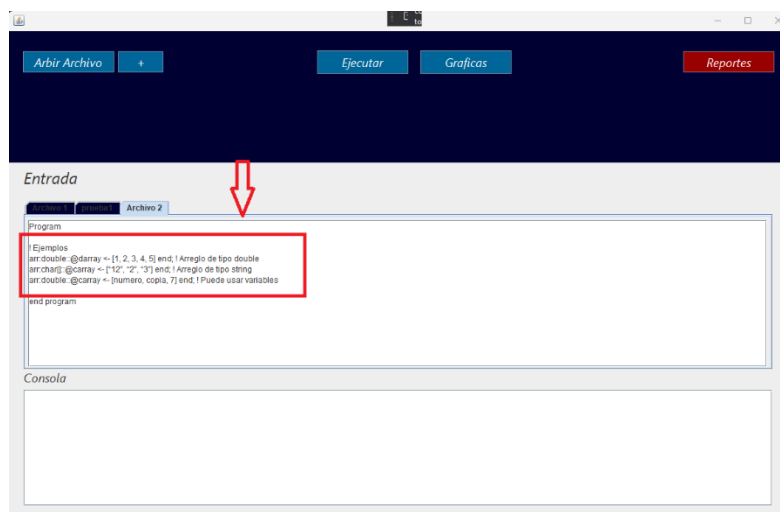
Se puede declarar 2 tipos de datos los tipos char[] acepta cadenas de texto dentro de doble comillas y los tipos double que acepta tipos numérico, para poder declarar una variable la estructura es la siguiente, debe de llevar la palabra reservada “var”, luego dos puntos “:”, luego el tipo de dato, dobles dos puntos “::”, el identificador que desea de su variable, el símbolo reservado “<-“ , la expresión y de ultimo finaliza con las palabras reservada de “end” y “;”



Sintaxis: var:<TIPO>::<ID> <- <EXPRESION> end;

Declaración de arreglo:

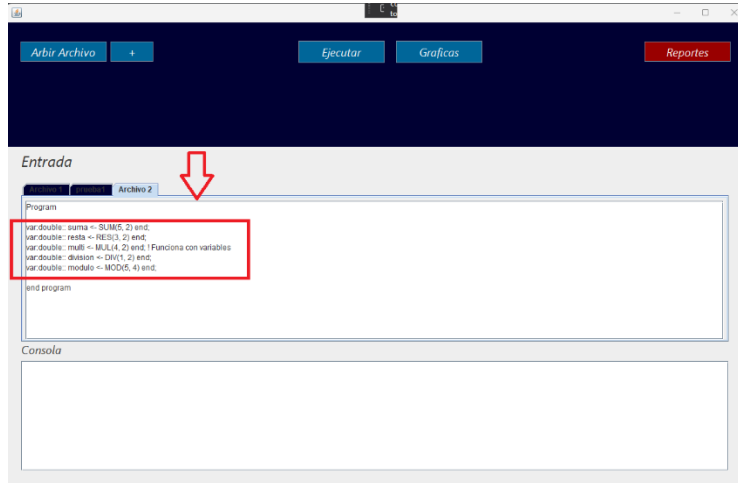
Un arreglo debe ser declarado antes de ser utilizado. Todos los arreglos tendrán un tipo de dato y un nombre identificador que siempre inicia con el símbolo “@”.



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

Operaciones Aritméticas

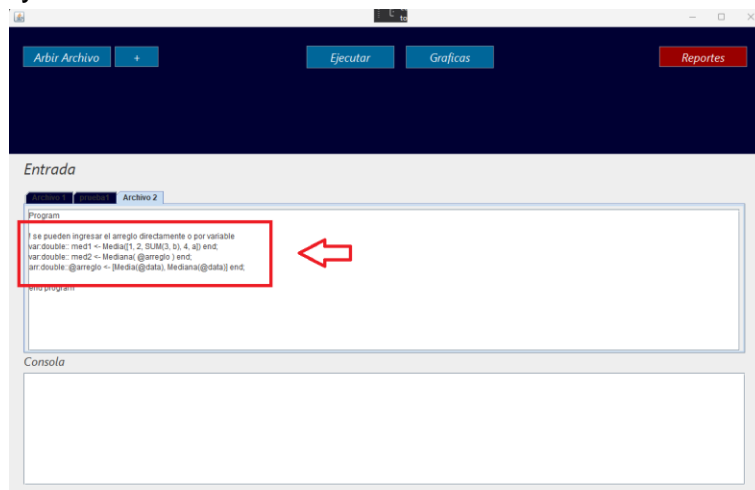
Las operaciones aritméticas solamente pueden ser realizadas entre expresiones de tipo double. Pueden ser utilizadas como expresiones en declaraciones de variables y como elemento de un arreglo. Todas las operaciones aritméticas pueden anidarse entre ellas, puede usar suma, resta, multiplicación, división y modulo



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

Funciones estadísticas:

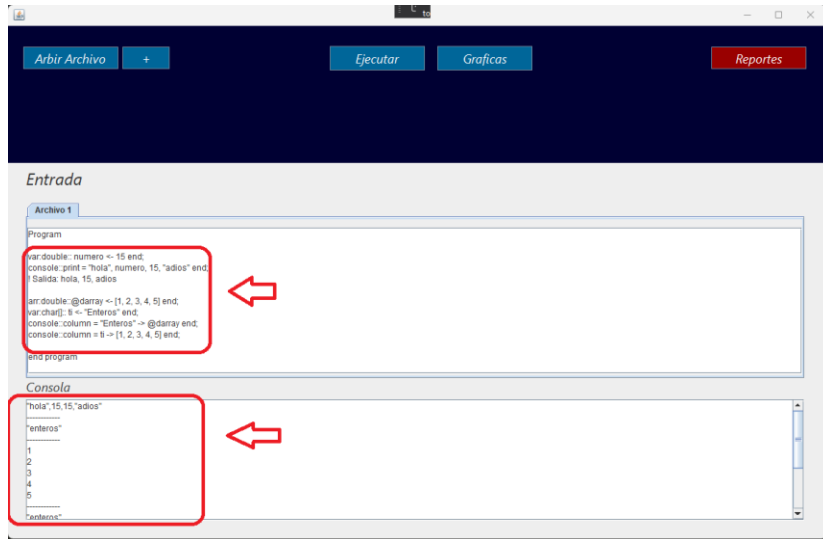
Las funciones estadísticas reciben un arreglo de tipo de double y realiza el cálculo solicitado para devolver el resultado como un número tipo double. El arreglo que ingresa a estas funciones puede ingresarse como una variable o declararse directamente, pueden realizar media, mediana, moda, varianza, max y min.



Sintaxis: `arr:<TIPO>:: @<ID> <- <Media(<ARREGLO_DOUBLE>)> end;`

Impresión en consola

Existen dos formas de imprimir en consola, puedes las expresiones o puedes imprimir arreglos, esta función permite mostrar arreglos en consola en un formato de tabla solicitando un título y un arreglo de cualquier tipo de dato. El arreglo puede ser de una variable o declarado directamente. El título puede ser ingresado directamente como una expresión o de una variable.



Sintaxis: `console::print = <EXP>, <EXP>, ... end;` `console::column = <CADENA> -> <ARREGLO> end;`

De la siguiente forma se va a mostrar los datos en la consola.

Funciones de graficas: El lenguaje DataForge, puede crear 4 tipos de graficas que a continuación le mostraremos

- **Grafica de barra:**

La gráfica de barras cuenta con los siguientes atributos:

titulo: Valor esperado: expresión o variable tipo char[].

Descripción: Muestra el título de la gráfica en la parte superior.

ejeX: Valor esperado: arreglo tipo char[]

Descripción: Son los valores de los label que se muestran sobre el eje horizontal.

ejeY: Valor esperado: arreglo tipo double

Descripción: Son los valores numéricos que se muestran sobre el eje vertical.

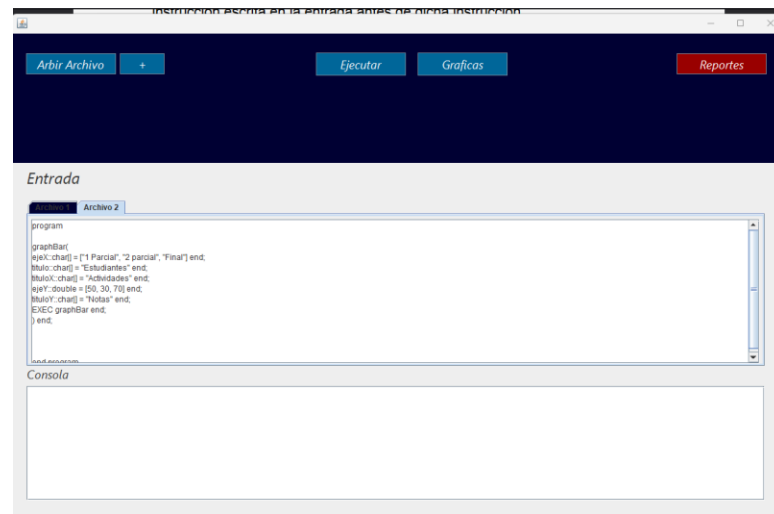
tituloX: Valor esperado: expresión o variable tipo char[].

Descripción: Valor que se muestra como título del eje horizontal.

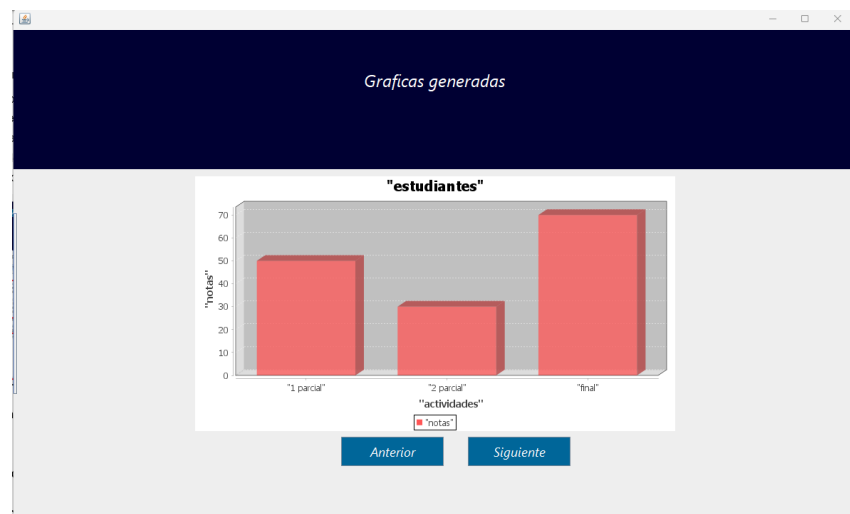
tituloY: Valor esperado: expresión o variable tipo char[].

Descripción: Valor que se muestra como título del eje vertical.

Exec: Muestra la gráfica en pantalla.



En la vista de graficas se visualizará.



- **Grafica de piel:**

La gráfica de pie cuenta con los siguientes atributos:

titulo: Valor esperado: expresión o variable tipo char[].

Descripción: Muestra el título de la gráfica en la parte superior.

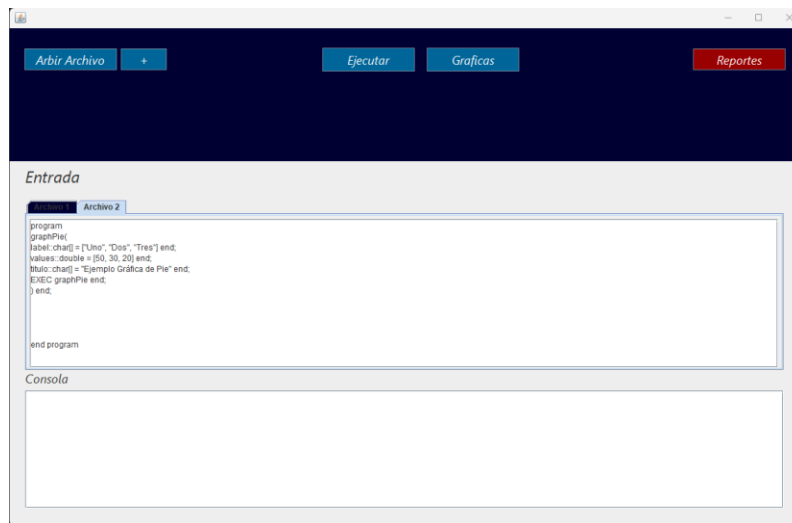
values: Valor esperado: arreglo tipo double

Descripción: Son los valores numéricos que se grafican.

label: Valor esperado: arreglo tipo char[]

Descripción: Son los valores de los label que se mostrarán en la gráfica.

Exec: Muestra la gráfica en pantalla.



En la vista de graficas se visualizará.



- **Grafica de linea:**

La gráfica de Línea cuenta con los siguientes atributos:

titulo: Valor esperado: expresión o variable tipo char[].

Descripción: Muestra el título de la gráfica en la parte superior.

ejeX: Valor esperado: arreglo tipo char[]

Descripción: Son los valores de los label que se muestran sobre el eje horizontal.

ejeY: Valor esperado:arreglo tipo double

Descripción: Son los valores numéricos que se muestran sobre el eje vertical.

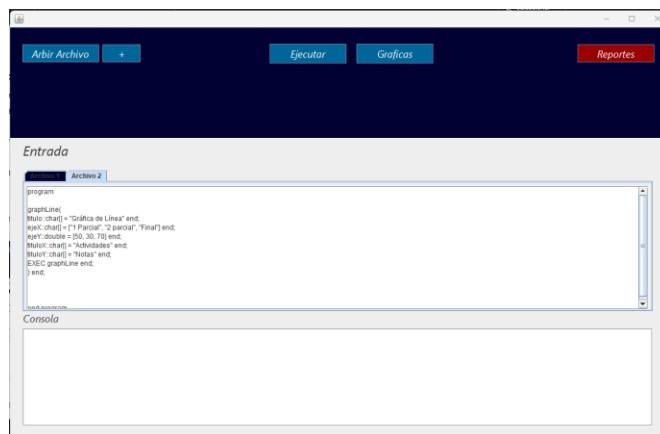
tituloX: Valor esperado: expresión o variable tipo char[].

Descripción: Valor que se muestra como título del eje horizontal.

tituloY: Valor esperado: expresión o variable tipo char[].

Descripción: Valor que se muestra como título del eje vertical.

Exec: Muestra la gráfica en pantalla.



En la vista de graficas se visualizará



▪ Gráfica histograma:

El histograma recibe un arreglo de tipo double y calcula la frecuencia de cada valor, la frecuencia acumulada y la frecuencia relativa. También genera una gráfica de cada valor y su frecuencia. Cuenta con los siguientes atributos:

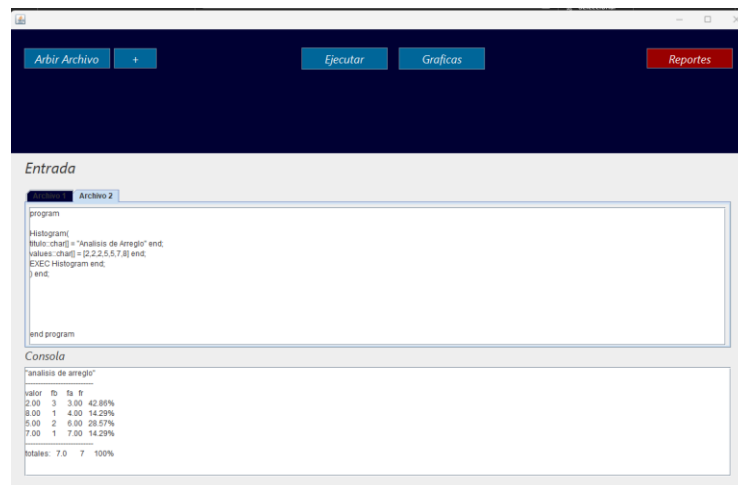
título: Valor esperado: expresión o variable tipo char[].

Descripción: Muestra el título de la gráfica en la parte superior.

valores: Valor esperado: arreglo tipo double

Descripción: Son los valores que se van a analizar para realizar el histograma.

Se visualizará en consola la información de la gráfica.



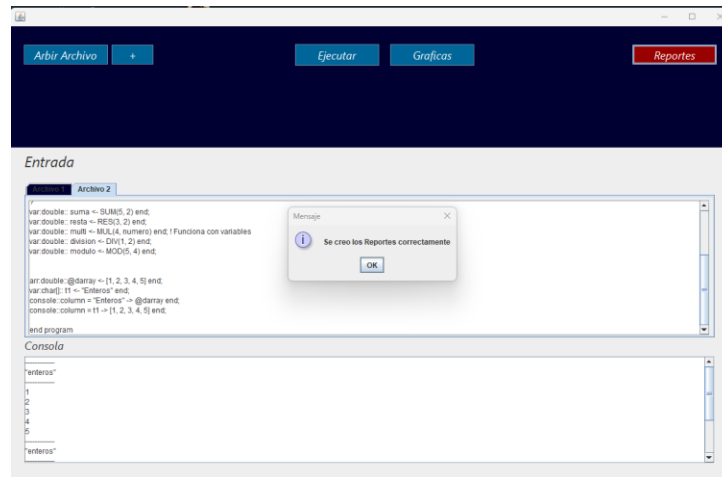
En la vista de graficas se visualizará



Nota: los atributos pueden venir desordenado y el programa debe de ejecutarlo.

Ejemplos de los reportes:

Al darle clic al botón de reporte se genera el siguiente mensaje



▪ Token

Reporte de tokens

#	Lexema	Tipo	Linea	Columna
1	program	Reservada	1	1
2	var	Reservada	2	0
3	:	dos_puntos	2	3
4	double	Tipos	2	4
5	:	Doble_puntos	2	10
6	numero	String	2	13
7	<-	Reserv	2	20
8	2.5	Numero	2	23
9	end	Reservada	2	27
10	:	Punto_com	2	30
11	var	Reservada	3	0
12	:	dos_puntos	3	3
13	char[]	Tipos	3	4
14	:	Doble_puntos	3	10
15	cadena	String	3	13
16	<-	Reserv	3	20
17	"cadena aqui probando1234 2.5"	cadena	3	23
18	end	Reservada	3	54
19	:	Punto_com	3	57
20	arr	Arreglo	5	0
21	:	dos_puntos	5	3
22	double	Tipos	5	4
23	:	Doble_puntos	5	10
24	@	Arroba	5	12
25	darray	String	5	13
26	<-	Reserv	5	20
27	[ParentA	5	23
28	1	Numero	5	24

▪ Errores

Reporte de Errores

#	Tipo	Descripción	Linea	Columna
1	Lexico	El caracter # no pertenece al lenguaje	1	9
2	Lexico	El caracter # no pertenece al lenguaje	6	0
3	Lexico	El caracter ? no pertenece al lenguaje	9	0

▪ Símbolos

Reporte de Simbolos

#	Nombre	Tipo	Valor
1	numero	variable double	2.5
2	cadena	variable char[]	"cadena aqui probando1234 2.5"
3	@darray	arreglo double	1,2,3,4,5
4	@carray	arreglo char[]	"12","2","3"
5	suma	variable double	7.0
6	resta	variable double	1.0
7	multi	variable double	10.0
8	division	variable double	0.5
9	modulo	variable double	1.0
10	@darray	arreglo double	1,2,3,4,5
11	t1	variable char[]	"enteros"