

Manual de Usuario



Al iniciar el programa el usuario se encontrará con la siguiente interfaz que cuenta con 4 áreas principales, las cuales son la barra de navegación, un área para manejar el texto que se ingrese, otra área para mostrar las gráficas que sean generadas por el lenguaje y la última área es la consola donde se mostraran las impresiones hechas durante el lenguaje.

1. Menú de navegación: En este menú se maneja el control sobre pestañas archivos y reportes, para crear una nueva pestaña se selecciona la parte de crear nueva pestaña para empezar a introducir el texto a mano o se puede abrir un archivo ya existente, cuando ya se tenga el código deseado se le da clic en la parte de menú ejecutar y el código ya procede a ejecutarse y crear todo lo necesario
2. Área de pestañas: en esta área se mostrará el texto a analizar, acá se pueden realizar modificaciones según se consideren necesarias, al ejecutar el código se ejecutará el código de la pestaña que este seleccionada actualmente
3. Área de graficas: acá se mostrarán las gráficas según se hayan creado en el código, esta parte cuenta con dos botones, los cuales se encargarán de ir cambiando las imágenes según la cantidad creada en el código
4. Área de terminal, esta área no se puede modificar y solo se encargará de mostrar las impresiones hechas en el código o también cuando se cree una grafica tipo histograma en esta área se mostrará la tabla de frecuencias de los datos.

El lenguaje

Variables

En este lenguaje a trabajar el usuario necesita hacer dos declaraciones importantes para la ejecución, y estas son:

PROGRAM

<! Resto de código aquí !>

END PROGRAM

Estas palabras serán necesarias para poder reconocer que en esta parte existe código que puede ser ejecutado

Para la declaración de variables se utiliza la siguiente sintaxis

var:double :: numero <- 2.5 end;

O si lo que se va a declarar es un string se hace de la siguiente manera

var: char[] :: texto <- "VARIABLE STRING" end;

En este lenguaje también se manejan impresiones donde se puede poner en consola el texto o numero según se quiera, para hacer una impresión se declara así:

console::print = "El numero es: " ,numero end;

Se pueden imprimir múltiples variables o strings según estas sean separadas por coma, en este lenguaje se permite la declaración de una variable y la posible utilización de esta dentro de futuras funciones, así como para imprimir podemos observar que se esta imprimiendo una variable declarada previamente

arr:double::@Arreglo<- [1,-1,2,numero,3] end;

también se permite el manejo de arreglos, los cuales son declarados con la palabra arr al principio y el nombre con un "@" para indicar que este será un arreglo, seguido de una flecha y una lista de valores. NO SE PUEDEN MANEJAR ARREGLOS MULTIDIMENSIONALES ACTUALMENTE.

Un arreglo puede ser impreso de la siguiente manera

console::column = "Arreglo a imprimir" -> @Arreglo end;

De esta manera se puede visualizar el arreglo como una tabla y ver fácilmente los datos que este contiene

Este lenguaje permite el uso de operaciones aritméticas de la siguiente manera

```
var:double :: suma <- SUM(5, 2) end;  
console::print = "El valor de la suma de 5 y 2: ", suma end;  
  
var:double :: resta <- RES(3, 2) end;  
console::print = "El valor de la resta de 3 y 2: ", resta end;  
  
var:double :: multi <- MUL(4, numero) end;  
console::print = "La multiplicacion es ",multi end;  
  
var:double :: division <- DIV(1, variable) end;  
console::print = "La division es: ", division end;  
  
var:double :: modulo <- MOD(5, 4) end;  
console::print = "The modulo of 5 and 4 = ", modulo end;  
  
var:double :: sumaf <- MUL( SUM(7,3) , RES(7, DIV(25,5))) end;  
console::print = "The result is ", sumaf end;
```

Se pueden anidar operaciones y en las operaciones se pueden llamar variables de tipo double para sumársela a un entero u otra variable

También se cuentan con operaciones estadísticas las cuales se utilizan de la siguiente manera, estas operaciones se pueden realizar con arreglos o con listas de datos

```
var:double :: mediaCalc <- media(@Arreglo) end;  
var:double :: mediaC <- media([1,2,3]) end;  
  
var:double:: medianaPrueb <- mediana([1,2,3]) end;  
  
var:double :: modaLV <- moda([1,2,3,4,5,6,1,2,2]) end;  
  
var:double :: maxN <- max([1,2,3,4,5,6,1,2,2]) end;  
var:double :: minN <- min([1,2,3,4,5,6,-0.1,2,2]) end;
```

Y por ultimo el manejo de graficas donde cada tipo de grafica requiere ciertos valores

Para manejar una grafica de barras necesitamos declararla así:

```
graphBar(  
  ejeX::char[] = @titulos end; !Donde @titulos es un arreglo  
  titulo::char[] = "Estudiantes" end;  
  ejeY::double = [50, 30, 70] end;  
  tituloX::char[] = "Actividades" end;  
  tituloY::char[] = "Notas" end;  
  EXEC graphBar end;  
) end;
```

Donde requiere un arreglo de tipo char, un titulo para la grafica, un titulo en vertical y otro en horizontal, el manejo de estos arreglos pueden ser con el id de un arreglo declarado previamente o de una lista de valores declarada dentro de []

Para una grafica de Pie se hace de la siguiente manera

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

Donde label son los nombres de los datos a manejar y valores la cantidad de veces de este dato

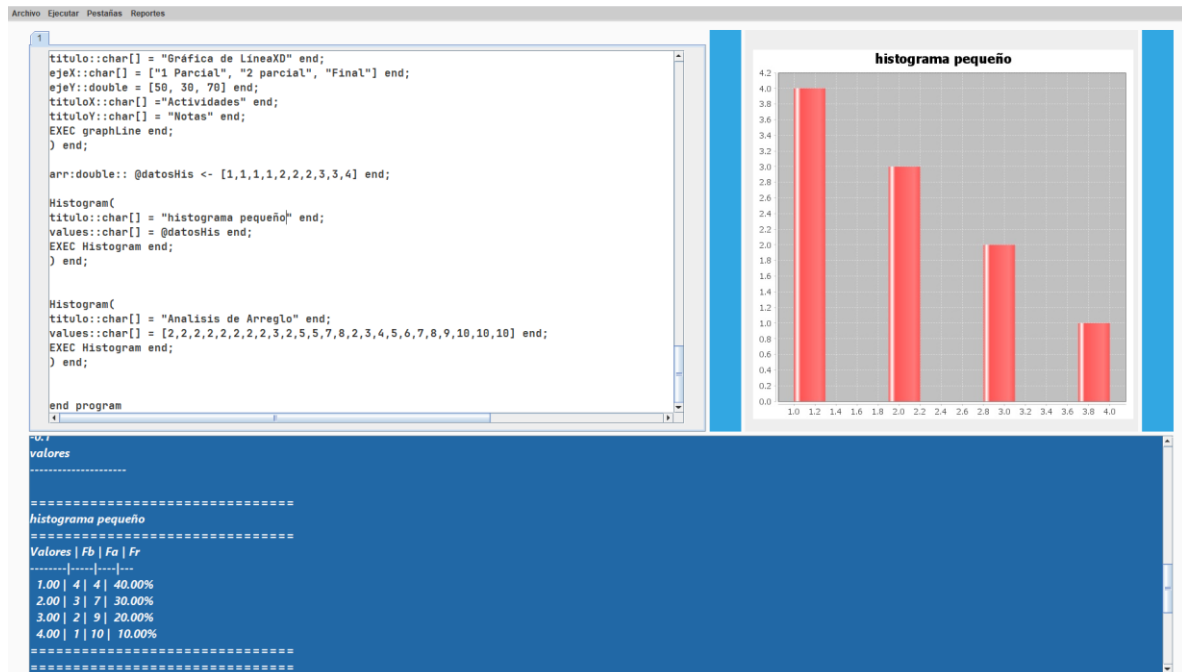
Para declarar una grafica de linea

```
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 graphLine end;  
) end;
```

O para un histograma

```
Histogram(  
  titulo::char[] = "Análisis de Arreglo" end;  
  values::char[] = [2,2,2,2,2,2,2,2,5,5,7,8,2,3,4,5,6,7,8,9,10,10,10] end;  
  EXEC Histogram end;  
) end;
```

Donde al crearse esta grafica en consola imprimirá una tabla de frecuencias de los datos dados por este arreglo en la declaración de valores



Acá podemos observar un ejemplo del programa con el código dentro del archivo de prueba para comprobar el funcionamiento correcto del programa, se puede observar la consola donde se muestran las impresiones y los print declarados por el usuario