

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de Lenguajes y Compiladores 1

Primer Semestre 2024

Ing. Kevin Lajpop

Carlos Acabal



DataForge

Proyecto1

Manual de Usuario

César Fernando Sazo Quisquinay

202202906

Objetivo Principal:

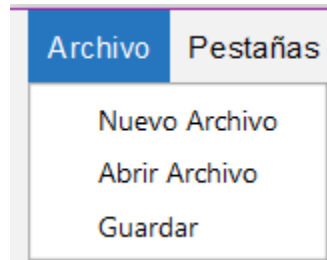
Crear un sistema capaz de realizar operaciones aritméticas y estadísticas, además de poder generar diversos gráficos a partir de una colección de datos que se hacen por medio de un archivo de texto, para esto se utilizaran los conocimientos sobre analizadores léxicos, sintácticos y semánticos para poder ejecutar las tareas y que por medio del archivo del texto se puedan mandar las instrucciones.

Menú Principal:



Esta es la ventana principal del programa, donde se pueden destacar cuatro áreas, la primera, la barra de menú superior donde se encuentran los apartados Archivo, Pestañas, Acción y Reportes, que permiten la manipulación de todas las funcionalidades del programa, un área de trabajo como seria el área de "Entrada", una sección para la consola para las impresiones que se deseen hacer sobre la ejecución y un apartado para ver las gráficas generadas.

El primer paso para la ejecución del programa sería cargar un archivo o crear uno desde 0, este texto se podrá editar en el área de “Entrada”, pudiendo agregar la cantidad de caracteres que se desee, al mismo tiempo se tendrá la capacidad de guardar el archivo de texto que se haya escrito en esta área de trabajo.



Para que se realicen las instrucciones, se deben seguir unas reglas gramaticales las cuales son las siguientes:

La principal regla del lenguaje que maneja el programa es que no reconoce entre mayúsculas y minúsculas, por lo que se tendrá libertad en el uso de estas, dando el mismo resultado sin importar si es una u otra.

Todas las instrucciones o sentencias que se declaren deben venir entre PROGRAM y END PROGRAM

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

Comentarios

El lenguaje del programa permite el manejo de comentarios que son ignorados por los analizadores con el objetivo de dejar mensajes a otros desarrolladores en el código fuente.

- Comentarios de una línea

Este comentario comenzará con “!” y termina al detectar un salto de línea.

- Comentarios multilínea

Este comentario comenzará con “<!” y terminará con “!>” sin importar la cantidad de saltos de línea que se haya dado.

```
! Esto es un comentario de una sola línea
<! Esto es un comentario
Multilínea !>
```

Declaración de variables

Una variable debe ser declarada antes de poder ser utilizada. Todas las variables tendrán un tipo de dato y un nombre de identificador. Las variables pueden ser declaradas utilizando otra variable como tal.

```
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
```

Declaración de Arreglos

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 @ y una lista de valores que se le asignara, que dentro de esta lista de valores pueden ir valores enteros, variables, operaciones aritméticas o estadísticas.

```
! 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 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.

- Suma
Es la operación aritmética que consiste en realizar la suma entre dos valores. Esta se realiza mediante la función `SUM(A,B)`
- Resta
Es la operación aritmética que consiste en realizar la resta entre dos valores. Esta se realiza mediante la función `RES(A,B)`
- Multiplicación
Es la operación aritmética que consiste en realizar la multiplicación entre dos valores. Esta se realiza mediante la función `MUL(A,B)`
- División
Es la operación aritmética que consiste en realizar la división entre dos valores. Esta se realiza mediante la función `DIV(A,B)`
- Módulo:
Es la operación aritmética que permite obtener el residuo de la división entre dos valores. Esta se realiza mediante `MOD(A, B)`

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.

Los resultados de estas funciones pueden ser utilizados pueden utilizarse en operaciones aritméticas, declaraciones de variables y declaraciones de arreglos.

- **Media**
Calcula la media del conjunto de números dentro de un arreglo de tipo double.
- **Mediana**
Calcula la mediana del conjunto de números dentro de un arreglo de tipo double.
- **Moda**
Calcula la moda del conjunto de números dentro de un arreglo de tipo double.
- **Varianza**
Calcula la varianza del conjunto de números dentro de un arreglo de tipo double.
- **Max**
Encuentra el valor más grande dentro de un conjunto de números dentro de un arreglo de tipo double.
- **Min**
Encuentra el valor más pequeño de un conjunto de números dentro de un arreglo de tipo double.

Impresión en Consola

El lenguaje del programa tiene dos funciones que permite mostrar en consola de salida las expresiones dependiendo de su tipo.

Imprimir Expresiones:

Esta función nos permite imprimir un conjunto de expresiones separadas por comas para ser mostradas en la consola. No pueden ingresar arreglos en esta sentencia.

```
! Imprime cada expresión separado por coma  
console::print = <EXP>, <EXP>, ... end;
```

Ejemplo:

```
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

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.

```
! Muestra una tabla de una columna en consola con cada línea un valor  
console::column = <CADENA> -> <ARREGLO> end;
```

Ejemplo:

```
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;
```

Funciones de Graficación

Para una mejor visualización de la información el lenguaje cuenta con funciones que permite graficar de manera personalizada un conjunto de datos utilizando la siguiente distintas sentencias pertenecientes a los gráficos exclusivamente, estos gráficos son visibles en la sección de la derecha llamado Gráficos donde se puede ver a manera de galería todas las gráficas por medio de los botones “<” y “>” que irán mostrando todas las gráficas generadas.

Atributos de los Graficos

título:

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.

títuloX:

Valor esperado: expresión o variable tipo char[].

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

títuloY:

Valor esperado: expresión o variable tipo char[].

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

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.

Ahora veremos los atributos necesarios para graficar cada tipo de grafico.

Gráfica de Barras

- Titulo
- EjeX
- EjY
- tituloX
- tituloY
- Exec

Gráfica de Pie

- Titulo
- Values
- Label
- Exec

Gráfica de Línea

- Titulo
- ejeX
- ejeY
- tituloX
- tituloY
- Exec

Histograma

- Título
- Values

Sección Pestañas

Permite la eliminación de la pestaña que se tenga seleccionada.

Sección Acción

Esta sección permite la ejecución del programa, que se basa en extraer el texto de la pestaña seleccionada, para poder enviarlo a los analizadores léxicos y sintácticos generados por herramientas conocidas como Jflex y cup. Para así poder reconocer e interpretar las instrucciones.

Sección Reportes

Muestran de forma visual las herramientas utilizadas para realizar la ejecución del código. Con estas herramientas nos referimos a los Tokens realizados por el analizador léxico, tabla de símbolos generados en la declaración de variables o arreglos en la sección del analizador sintáctico y errores que se hayan presentado durante la ejecución, guardándolos según su tipo, léxico o sintáctico. Estos son generados y mostrados en formato HTML.

Cada reporte puede ser generado posterior a la ejecución del programa, de lo contrario mostrara un mensaje de error de que no se ha cargado ningún Token, Error, Símbolo