MANUAL DE USUARIO

(Fase 1)

Requerimientos

Sistema Operativo (Windows o Linux)

Java 1.8.0_201 o compatibles.

Jar ejecutable o proyecto completo.

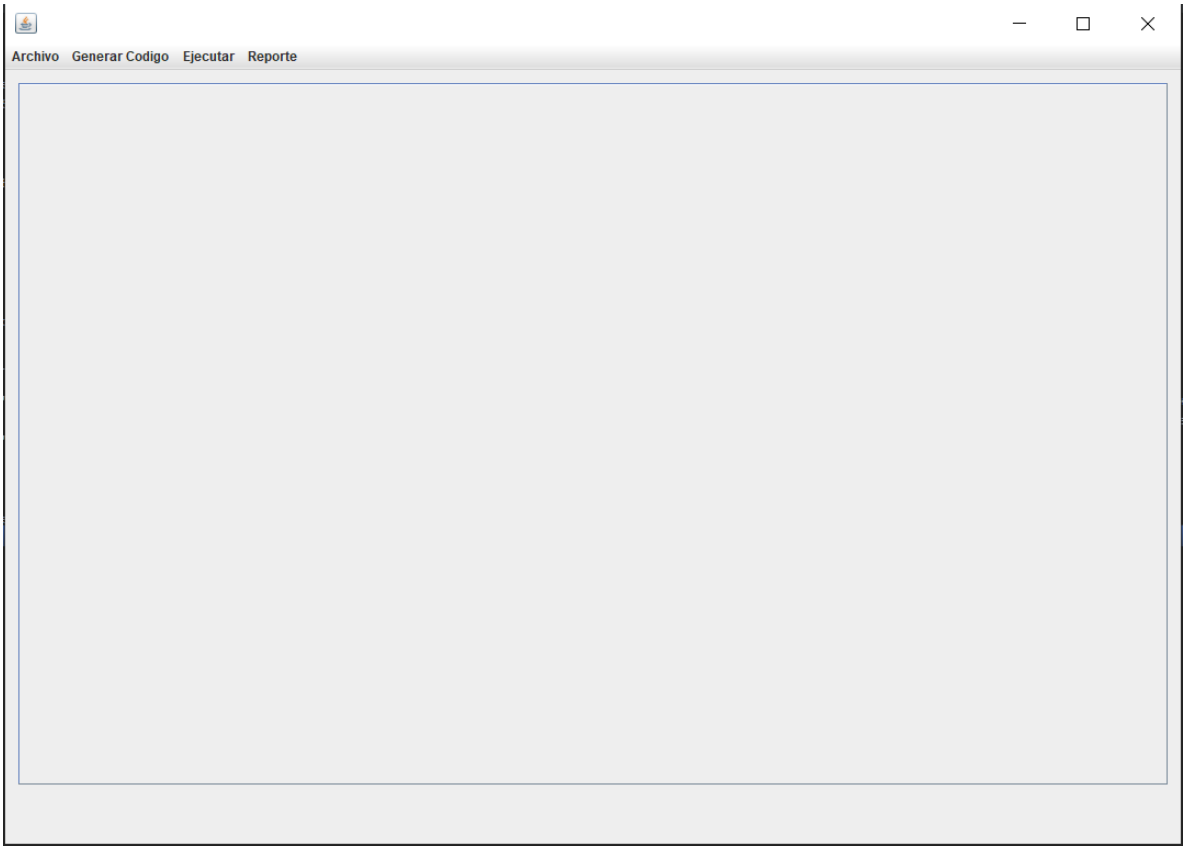
Entender los lenguajes VISUAL BASIC, JAVA, PYTHON Y CPP (para editar archivos en el IDE).

Los archivos deben de tener la extensión: .mlg

Descripción

El programa es un editro de código, el cual genera código de tres direcciones a partir del archivo de entrada que se este leyendo.

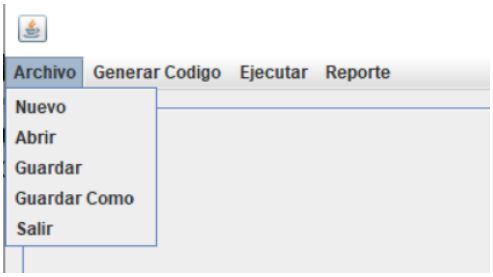
Ventana Principal



En esta ventana es donde se abre, o crean los archivos respetivos del programa.

Opciones Disponibles

Pestaña Archivo



Nuevo: Despliega la pestaña denominada Programa, que contendrá exclusivamente el código fuente del programa a compilar.

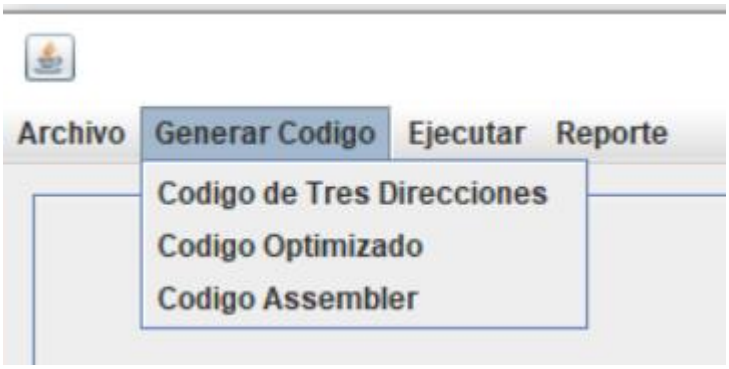
Abrir: Esta opción permite abrir un archivo .mlg, éste contiene el código fuente de un programa almacenado con anterioridad. El contenido de este archivo será mostrado en la pestaña denominada Programa. Esta opción es únicamente para los archivos con extensión .mlg.

Guardar: Guarda el contenido de la pestaña activa en ese momento, en la dirección que el usuario desee.

Guardar Como: Permite guardar el contenido de la pestaña activa en ese momento, con diferente nombre, pero con la extensión que le corresponda.

Salir: Finaliza la ejecución de la aplicación.

Pestaña Generar Codigo



Actualmente solo se cuenta con la opción de Código de Tres Direcciones, este genera el código del mismo en el editor de texto.

Pestaña de Reportes

Actualmente solo se cuenta con el reporte de Errores, este se muestra en caso de que existan errores en el texto o código.

Descripción Lenguajes

Los lenguajes solo admiten las variables primitivas Integer, Float y Char.

VISUAL BASIC

Declaración Variables

Declaración	Ejemplo asignación contenidos
Dim A As Integer	A = 123

Bucles

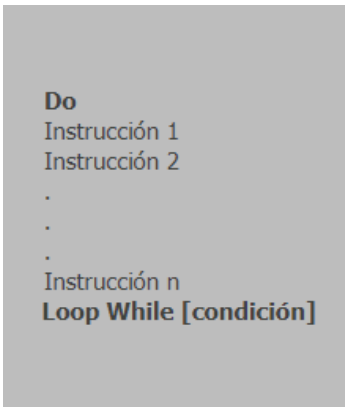
Código (versionesVB menos recientes)

```
Rem Curso Visual Basic
aprenderaprogramar.com
Option Explicit

Dim VAR As Integer
Dim Vi As Integer
Dim Vf As Integer

Private Sub Form_Load()
Vi = 1
Vf = 3
For VAR = Vf To Vi Step –1
   '[También supondría tres repeticiones
For VAR = Vi to Vf]
   MsgBox ("hola")
Next VAR
End Sub
```

Código (versiones VB más recientes)

```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On

Public Class Form1
   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
      Dim VAR As Integer
      Dim Vi As Integer
      Dim Vf As Integer
      Vi = 1
      Vf = 3
      For VAR = Vf To Vi Step -1
         '[También supondría tres repeticiones For VAR = Vi to Vf]
         MsgBox("hola")
      Next VAR
   End Sub
End Class
```

```
Do While [condición]
Instrucción 1
Instrucción 2
.
.
.
Instrucción n
Loop
```



Condiciones

```
if condicion then

     'Instrucciones

end if
```

```
Select valor
            Case caso1
                'Instrucciones
            Case caso2
                'Instrucciones
            Case casoN
                'Instrucciones
            Case Else
                'Instrucciones
        End Select
```

Los case de un select se pueden escribir con rangos con **To**, por ejemplo, **1 TO 5**, también podemos usar comas para indicar varios valores. Por ejemplo:

```
Module Module1

    Sub Main()

        Dim valor As Integer = 3

        Select Case valor
            Case 1 To 5
                Console.WriteLine("El valor esta entre 1 y 5")
                Console.WriteLine("El valor esta es 6 o 8 o 9")
            Case Else
                Console.WriteLine("El valor es otro")
        End Select

        Console.ReadLine()

    End Sub

End Module
```

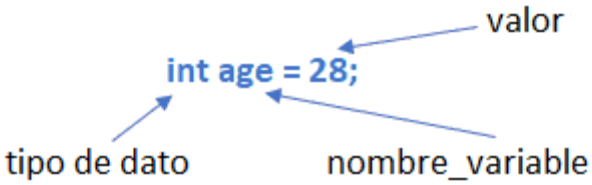
Espero que os sea de ayuda. Si tenéis dudas, preguntad, estamos para ayudar.

Condicionales Lógicos

And, or, not

JAVA

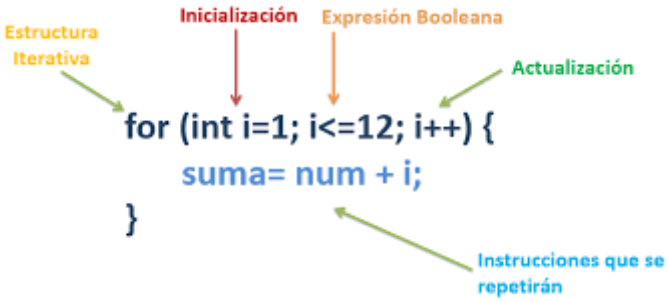
Declaración Variables



Bucles

```
int i = 0;
while (i < 10) {
    System.out.println("i: " + i);
    i++;
}
```

```
int i = 5;
do {
    System.out.println(i);
    i++;
} while (i <= 10);
```



Condicionales

```
if (condicion1){
    sentenciasA;
} else if (condicion2){
    sentenciasB;
} else if (condicion3){
    sentenciasC;
}
```

```
switch(expresion){

      case valor1:
           sentencias B1;
      break;

      case valor2:
           sentencias B2;
      break;

      case valor3:
           sentencias B3;
      break;

      ...

      [default:
           sentencias B4;]

}
```

Lógicos

And = &&

Or = ||

Not = ¡

Diferente = ¡=

VB

Variables

$$x + 3 = 5$$

Bucles

```
i = 1
while i <= 3:
    print(i)
    i += 1
print("Programa terminado")
```

```
for num in range(0, 11, 2):
    print(num)
```

Condicionales

```
if condición_1:
    bloque 1
elif condición_2:
    bloque 2
else:
    bloque 3
```

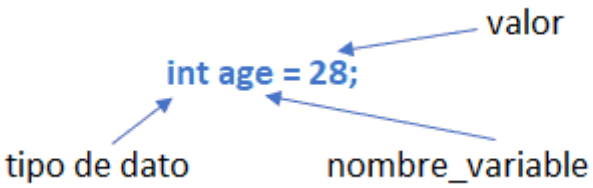
Lógicos

And, or, not

```
def evaluacion (nota):
    valoracion="aprobado"
    if nota<5:
        valoracion="suspenso"
        return valoracion
```

CPP

Variables



Bucles

Condicionales

```
if (condicion1){
    sentenciasA;
} else if (condicion2){
    sentenciasB;
} else if (condicion3){
    sentenciasC;
}
```

```
switch(expresion){

     case valor1:
          sentencias B1;
     break;

     case valor2:
          sentencias B2;
     break;

     case valor3:
          sentencias B3;
     break;

     ...

     [default:
          sentencias B4;]

}
```

Lógicos

And = &&

Or = ||

Not = ¡

Diferente = ¡=

Descripción del Archivo de Entrada

```
%%VB
   Módulos  en Visual Basic
%%JAVA
   Clases en Java
%%PY
   Funciones y Procedimientos en Python
%%PROGRAMA
Sección de librerías de C
#include "VB"            //incluye el código en VB
#include "JAVA.*"        //incluye todas las clases declaradas en
                         //la sección de Java
#include "JAVA.nombre"   //incluye una clase en especifica
                         //declarada en la sección de Java
#include "PY"            //incluye el código en Python
Sección declaración de Constantes
Sección de Variables Globales

void main()
{
 // Programa principal
}
```

En los lenguajes VB, Java y Python se utilizarán de forma limitada, únicamente:

- Expresiones aritméticas
- mensajes a pantalla
- solicitud de datos a usuarios: intinput, floatinput, charinput
- ciclos (for, while, do while)
- manejo de condiciones (if, switch)
- declaración de variables
- asignaciones variables
- Funciones
- Procedimientos
- Clases (Java)

Además, los únicos tipos que se manejarán en estos lenguajes son entero, real y carácter. Las declaraciones de funciones, métodos en Visual Basic y métodos en Java se manejarán

de forma pública (PUBLIC).

El programa principal que se maneja en sintaxis de C se especifica a continuación:

Especificaciones del programa principal

Tipos de datos

Los tipos de datos que se utilizarán son:

- int
- char
- float

Arreglos

Arreglos de N dimensiones de cualquier tamaño. Los arreglos pueden ser de cualquier tipo.

Tipo arreglo[dim1];

Tipo arreglo[dim1][dim2][dim3];

Al momento de usar un arreglo, dentro de las dimensiones podrán venir expresiones aritméticas, arreglos, funciones, constantes, etc. Ejemplo:

int arreglo[25+4][CONSTANTE_ENTERA];

x = vector[5*4/7+8*9(4+2)][matriz[1][2*7]];

Constantes

Definición de constantes:

const tipo nombre_constante = valor;

const float pi = 3.14159;

const char c = 'X'; // X es una constante tipo char

const int X = 10; // X es un tipo int

Operadores

Todos los operadores aritméticos, números enteros y reales ambos pueden ser negativos,

los operadores aritméticos pueden ser: +, -, *, /, % (modulo ó residuo), = asignación y

paréntesis.

Ejemplo

var = (1 + 2 – 3 * 4 / 5) % (5 * -3);

Comentarios

Los comentarios de línea o de bloque que se hagan deberán de ser pasados a código tres

direcciones y al código ensamblador.

//Comentario de línea

/*Comentario

de bloque*/

Sentencia if

En la instrucción If, dentro de la condición pueden venir expresiones aritméticas y

relacionales (<, >, = =, !=), esto implica operadores && (and), || (or), !(not); con o sin Else.

if (condición)

{

SENTENCIAS;

}

else

{

SENTENCIAS;

}

Donde condición puede ser cualquier condición que involucre operadores aritméticos,

relacionales y lógicos, y cualquier tipo de variable. Si el valor resultante de la condición es

mayor que 0, la condición es VERDADERA, si el resultado de la condición es menor o igual

a 0 la condición es FALSA. Ejemplo:

(VarAr[1] + 1 > var2 * 2 && var3 != var4)

Sentencia switch

La instrucción Switch, con n case y con o sin Default.

switch (variable)

{

case valor1:

SENTENCIAS;

break;

case valor2:

SENTENCIAS;

break;

Ciclo for

for (variable = valor_inicial ; condición ; variable = variable + valor_de_aumento)

{

SENTENCIAS;

}

Valor_inicial y valor_de_aumento será un número entero o una variable tipo entero,

condición puede ser cualquier condición, refiérase a la parte condición de la instrucción if

Ciclo while

while (condición)

{

SENTENCIAS;

}

Condición puede ser cualquier condición, refiérase a la parte condición de la instrucción if

Ciclo do while

do

{

SENTENCIAS;

}

while (condición);

Condición puede ser cualquier condición, refiérase a la parte condición de la instrucción if.

Llamadas a funciones y procedimientos

Variable = VB.nombre_funcion(parámetros);

Variable = PY.nombre_funcion(parámetro);

VB.nombre_procedimiento(parámetros);

PY.nombre_procedimiento(parámetros);

scanf

Esta instrucción permitirá asignar valores a las variables.

scanf(“ mascara “, &variable);

Mascara se refiere a texto y al indicador de que tipo de dato leerá, ejemplo:

scanf(“Valor %d”, &var); //Leerá del teclado un valor para asignar a variable tipo int

scanf(“Valor %c”, &var); //Leerá del teclado un valor para asignar a variable tipo char

scanf(“Valor %f”, &var); //Leerá del teclado un valor para asignar a variable tipo float

printf

Esta instrucción permitirá desplegar mensajes y los valores de las variables

printf(“ texto “); ó printf(“El valor es mascara “, var);

En la primera instrucción solo se mostrará texto , en la segunda se mostrará el texto y el

valor que contenga la variable var , puede haber más de una variable por mostrar.

Mascara se refiere a texto y al indicador de que tipo de dato desplegará, ejemplo:

printf(“Valor %d”, var); //Desplegará el valor de una variable tipo int

printf (“Valor %c”, var); //Desplegará el valor de una variable tipo char

printf (“Valor %f”, var); //Desplegará el valor de una variable tipo float

clrscr

limpiar la pantalla, ejemplo:

clrscr();

getch

Esta instrucción leerá una tecla del teclado para poder seguir la ejecución del programa. El

valor que regresa puede o no ser asignado a una variable tipo char o int, si es asignado a

una variable tipo char la variable tendrá el carácter que se presionó; si es asignado a una

variable tipo int la variable tendrá el valor ASCII del carácter que se presionó.

Clases

Las clases se manejarán de la siguiente manera

Declaración

JAVA.Nombre_Clase Nombre_Var1, Nombre_Var2;

JAVA.Nombre_Clase Nombre_Var1(parámetros); // llamando a constructor

Llamada a métodos

Variable = JAVA.nombre_objeto.método(parámetros);