****

Universidad Tecnológica Centroamericana

Compiladores 2

Manual de Usuario de

Ing. Carlos Vallejo

LEONARDO MONTOYA 11741303

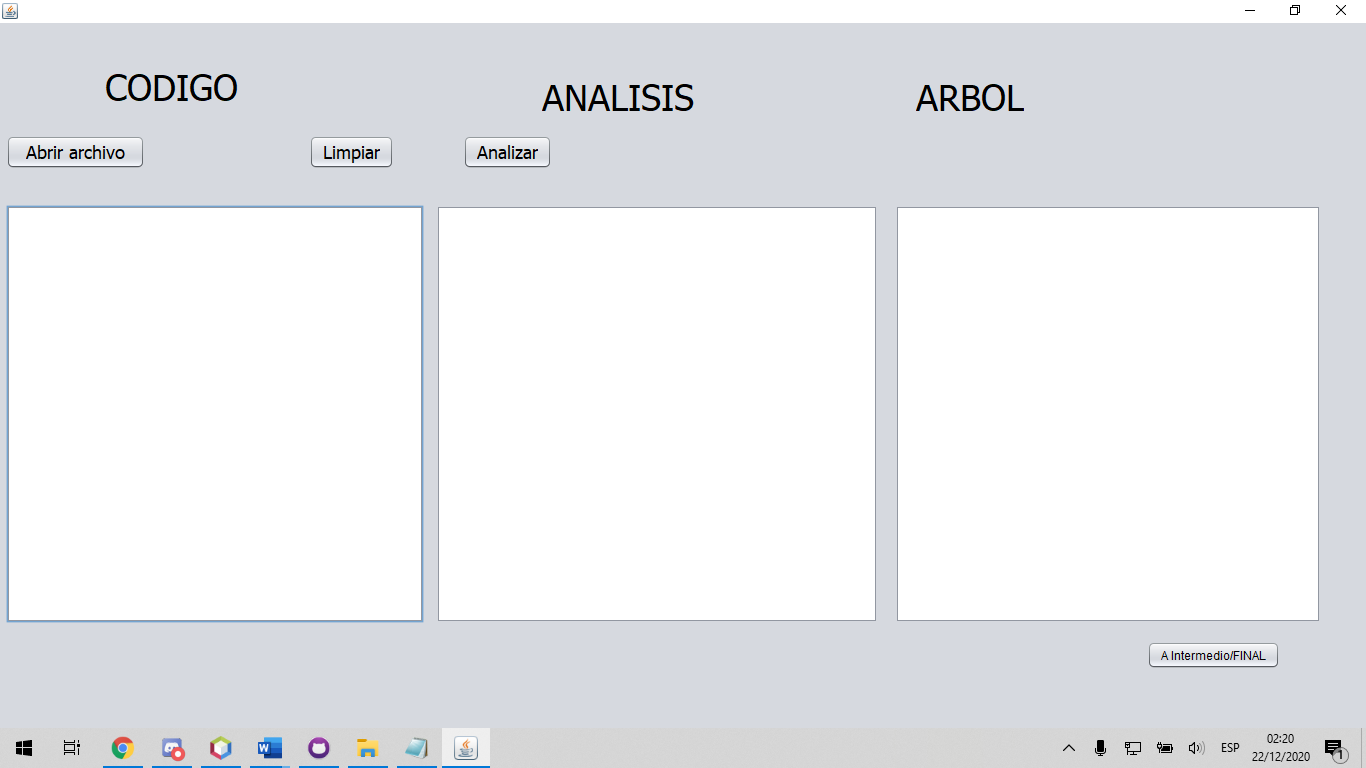
KENNETH NUÑEZ 11741149

ERICK CARRASCO 11641166

22 de Diciembre de 2020

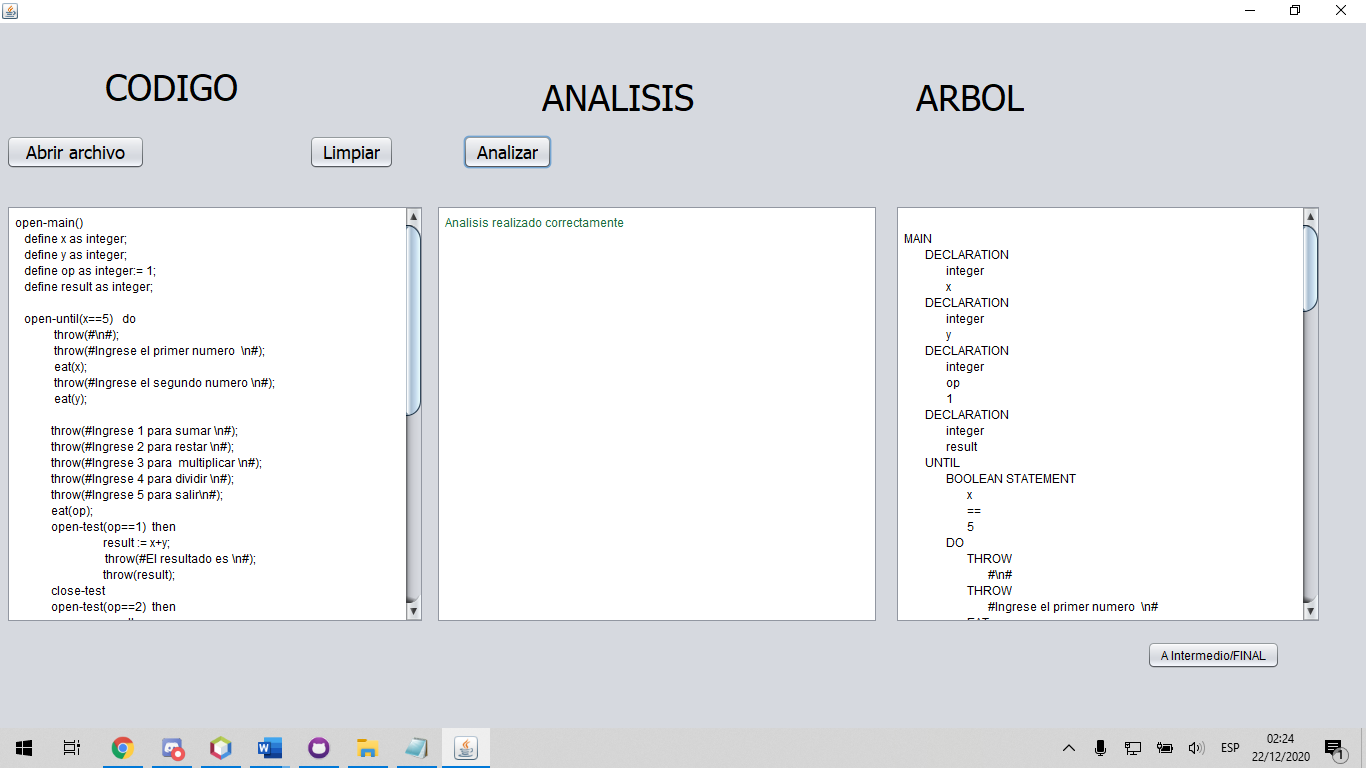
**Bienvenido a nuestro IDE**

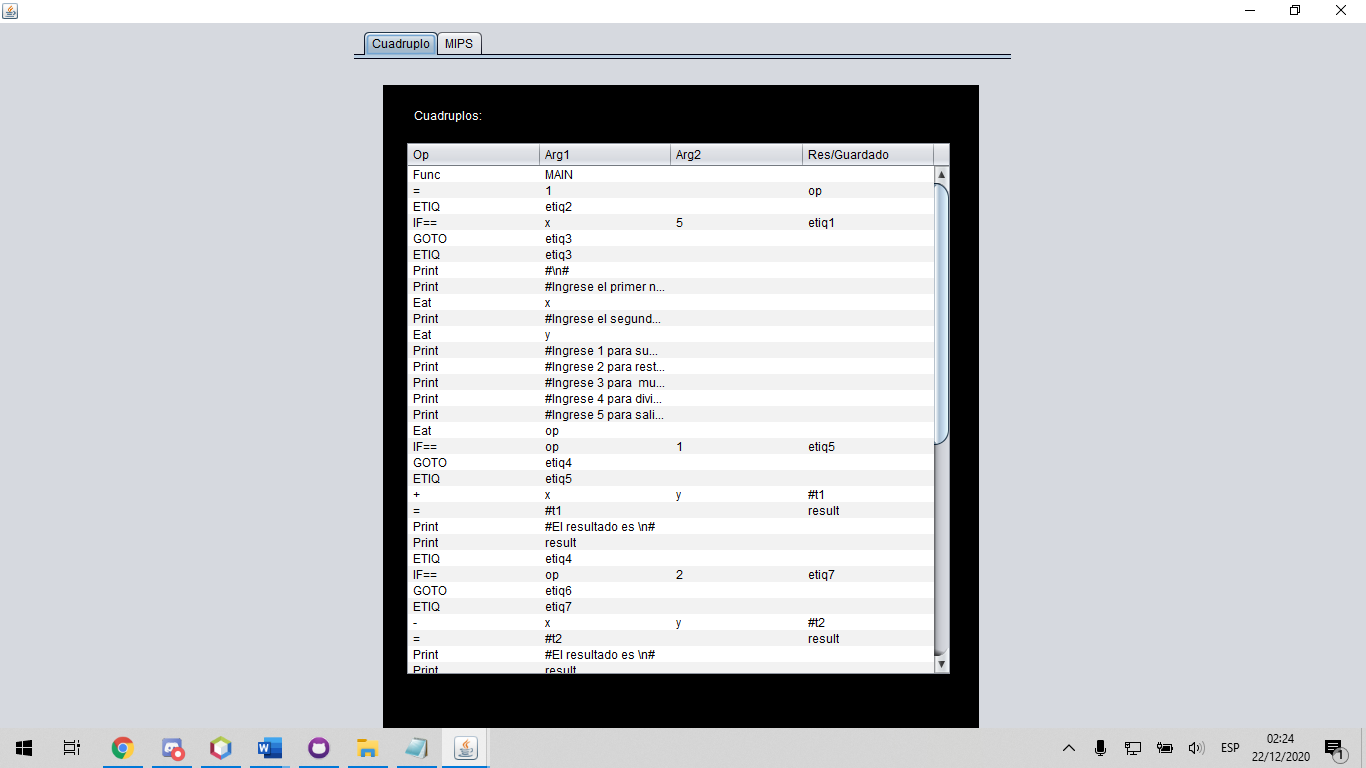
**Ejecuta el programa y disfrútalo.**

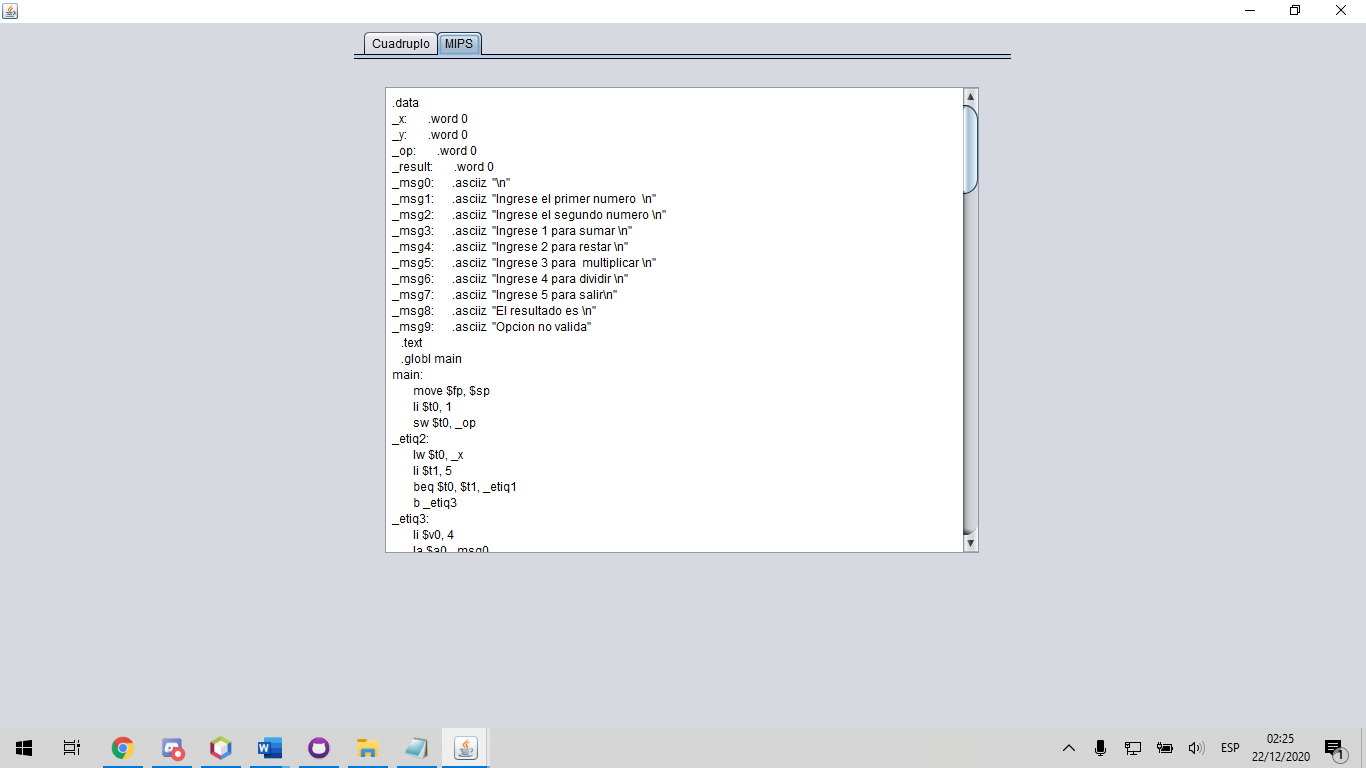


La interfaz cuenta con un text área que puede ser usado para editar en tiempo real el código.

¡Tan fácil como ingresar el código y correrlo!







**Descripción del lenguaje:**

**Tipos**

1. Entero: integer
2. Boolean: boolean
3. Char: character
4. Arreglo: tipo[]

**Declaración**

1. define id as tipo;
2. define h1 as integer;
3. define classic as character[9];

**Asignación**

1. h1 := 2;
2. define h2 as character := 'x';

**OPERADORES RELACIONALES**

**& AND**

**| OR**

**! NOT**

**<**

**>**

**<=**

**=>**

**==**

**!=**

**Comentarios:**

* Una línea: //
* Multilínea:/\*

\*/

**Lectura y escritura:**

* eat (id); //LECTURA
* throw(id); //ESCRITURA
* throw(#CADENA#)//Impresión de Strings
* throwln(id);ESCRITURA

**IF:**

open-test (cond) then

b1;

close-test

**IF ELSE:**

open-test (cond) then

b1;

or

b2;

close-test

**FOR:**

open-for (/\*integer i:= num; condición;incremento\*/) do

//Sentencias

close-for

**until:**- cuando la condición se cumple para

open-until (/\*condición\*/) do

//Sentencias

close-until

**SWITCH:**

open-match(x) do

when A := statement;

stop;

close-match

**Funciones:**

open-method tipo f()

close-method

**main:**

open-main()

close-main

**Ejemplo:**

open-main()

define x as integer;

define y as integer;

define op as integer:= 1;

define result as integer;

open-until(x==5) do

throw(#\n#);

throw(#Ingrese el primer numero \n#);

eat(x);

throw(#Ingrese el segundo numero \n#);

eat(y);

throw(#Ingrese 1 para sumar \n#);

throw(#Ingrese 2 para restar \n#);

throw(#Ingrese 3 para multiplicar \n#);

throw(#Ingrese 4 para dividir \n#);

throw(#Ingrese 5 para salir\n#);

eat(op);

open-test(op==1) then

result := x+y;

throw(#El resultado es \n#);

throw(result);

close-test

open-test(op==2) then

result := x-y;

throw(#El resultado es \n#);

throw(result);

close-test

open-test(op==3) then

result := x\*y;

throw(#El resultado es \n#);

throw(result);

close-test

open-test(op==4) then

result := x/y;

throw(#El resultado es \n#);

throw(result);

close-test

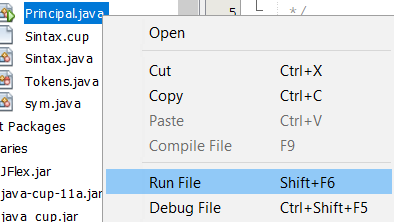
open-test(op>5) then

throw(#Opcion no valida#);

close-test

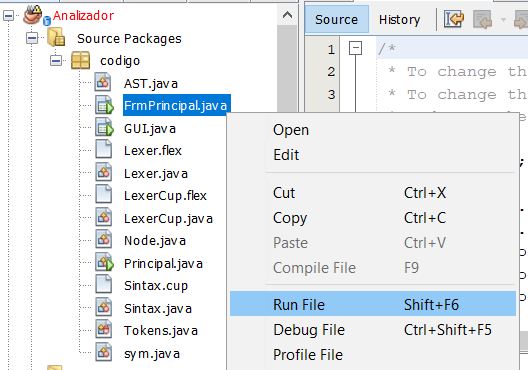
close-until

close-main

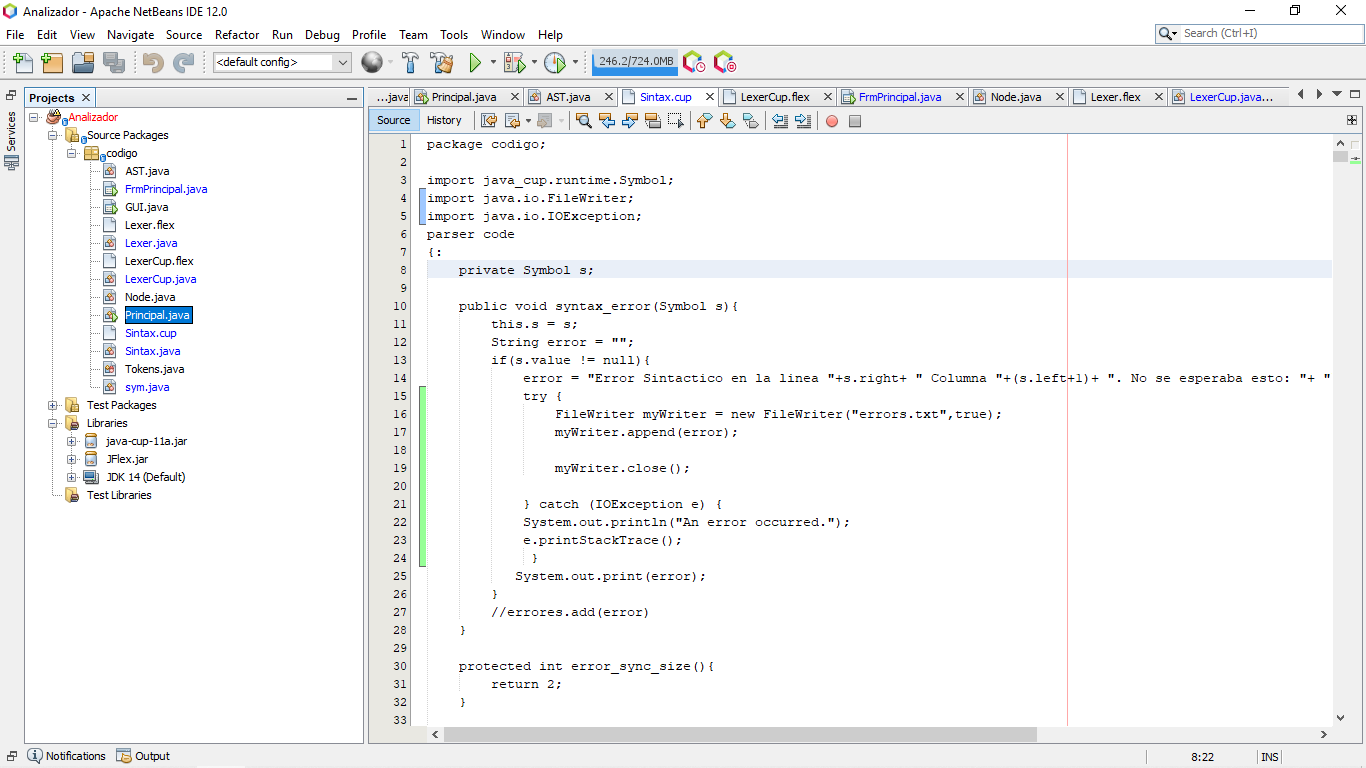


Si se modifica la gramatica ejecuta el Principal.java para actualizar el analizador lexico y sintactico.

Posteriormente ejecuta el FrmPrincipal.java y disfruta del IDE.



///////////////



Importar en este orden.