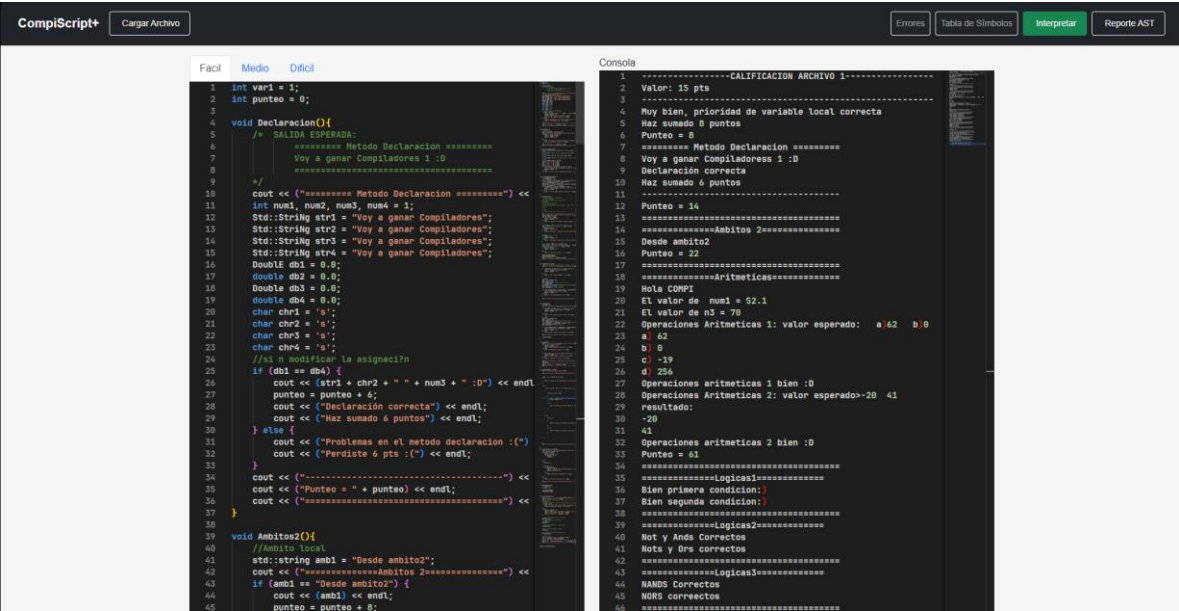


Manual de Usuario



este proyecto cuenta con las funcionalidades que se pueden observar acá

Como podemos ver cuenta con un botón que permite cargar un archivo en la pestaña que tengamos seleccionada actualmente

Se permite el manejo de múltiples pestañas

Y se cuenta con botones que mostraran los reportes al ejecutar el programa

Tabla de Símbolos

Identificador	Tipo	Valor
var1	Entero	0
arreglo1	Entero	0,0,0,0,0
arreglo2	Entero	0,0,1,2,0,0,5,1,0,0,8,0,0

en el reporte de tabla de símbolos se obtendrá una lista del Id el tipo y el valor del símbolo declarado en la parte del editor en el proyecto, a continuación, se describe el uso de las funciones del código que se manejará en este proyecto.

Comentarios

```
// Esto es un comentario de una sola línea

/* Esto es un comentario
Multilínea */
```

Con los comentarios el usuario puede utilizarlos para explicar, recordar o tener un mayor control sobre el código que está utilizando

```
<TIPO> identificador
<TIPO> id1, id2, id3, id4;
<TIPO> id1. id2, id3, id4 = <EXPRESION>
// Ejemplos
int numero;
int var1, var2, var3;
std::string cadena = "hola";
char var4 = 'a';
bool flag = true;
double a, b, c = 5.5;
```

Esta será la forma para declarar una variable, como podemos observar acá se utiliza un tipo y un id o varios ids, y estos pueden tener asignado o no un valor, si no tiene un valor asignado se le dará un valor por defecto a la variable

```
// Declaración de tipo 1
<TIPO> <ID> '[' ']' = new <TIPO> '[' <EXPRESION> ']' ';'
<TIPO> <ID> '[' ']' '[' ']' = new <TIPO> '[' <EXPRESION> ']' '['
<EXPRESION> ']' ';'

```

Se pueden declarar arreglos de una o dos dimensiones, para este se contarán con dos maneras de declararlo, en la primera se le especificara un tamaño y en la segunda se le colocara una lista de valores dentro de los []

```
<TIPO> <ID> '[' ']' = '[' <LISTAVALORES> ']' ';'
// Ejemplo
int vector1[] = new int[4];
char vector2[][] = new char[4][4];
std::string vector3[] = ["Hola", "Mundo"];
int vector4 [][] = [ [1, 2], [3, 4] ];
```

Para acceder a una posición de un vector se utilizará la siguiente sintaxis

```
<ID> '[' <EXPRESSION> '']'
```

// Ejemplos

```
std::string vector3[] = ["Hola", "Mundo"];
```

```
std::string valor3 = vector3[0]; // Almacena el valor "hola"
```

```
int vector4 [][] = [ [1, 2], [3, 4] ];
```

```
int valor4 = vector4[0][0]; // Almacena el valor 1
```

Esto puede ser muy útil cuando queremos hacer operaciones o funciones que involucran arreglos ya que podemos acceder individualmente a cada dato que contiene este arreglo

A continuación se colocara un ejemplo del AST generado con un ejemplo en el cual se observa como claramente de cada nodo salen las expresiones que este contiene según sean terminales o no terminales que serian las palabras reservadas, los símbolos o las expresiones ya interpretadas así como números, textos, etc.

