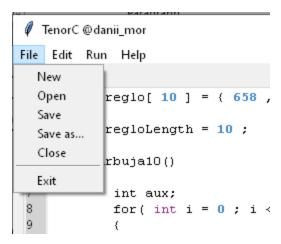
USER MANUAL

MENU BAR

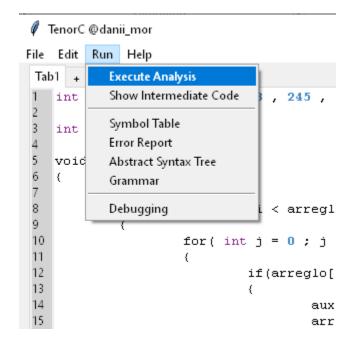


In this section we will be able to appreciate all the functions of our application. We will start with File that is used for:

- Create new
- Open file
- Save File
- Save as
- Close file
- Close application

The most important part of our application falls on this section. In Run we can do analysis and verify our options in different ways. In them we can see:

- Execute analysis
- Show Intermediate Code
- Symbol Table
- Error Report
- AST
- Grammar
- Debugging



EDITOR AREA

```
∅ TenorC @danii_mor
```

```
File Edit Run Help
Tab1 +
         arreglo[ 10 ] = { 658 , 245 , 654 , 956 , 5 , 754
   int
3
   int
         arregloLength = 10;
5
   void burbuja10()
6
7
            int aux;
8
            for ( int i = 0 ; i < arregloLength - 1 ; i++ )
10
                     for( int j = 0 ; j < arregloLength - i
11
12
                              if(arreglo[ j + 1 ] < arreglo[</pre>
13
14
                                       aux = arreglo[ j + 1 ]
15
                                       arreglo[ j + 1s ] = ar
16
                                       arrealof i l = aux:
```

In the text editor we can have multiple windows. So, you can work with multiple files at the same time.

It can also be seen that the line number is being indicated while we are advancing in the writing of our editor.

Seeing in more detail you can see the highlighting of colors. This to be effective we must apply spaces around each keyword.

The part of applying highlighting to pasted code in the editor is still in alpha.

Therefore, the space between the keywords must be manually applied for the editor to detect the colors.

In this example we can detail how this small air should be created between the keywords. This for the highlighting to take effect.

Also keep in mind that when there is text strings they generate a bug so you must have a space within the text to be able to detect where the highlighting of the text ends.

Symbol Table

| IDENTIFIER | TYPE | SIZE | VALUE | SCOPE | REFERENC |
|----------------|------|------|----------|--------|----------|
| var1 | INT | 0 | ID | GLOBAL | \$t0 |
| punteo | INT | 0 | ID | GLOBAL | \$t1 |
| Declaracion | VOID | 0 | FUNCTION | GLOBAL | LO |
| eraciones Basi | VOID | 0 | FUNCTION | GLOBAL | L1 |
| acionesAvanz | VOID | 0 | FUNCTION | GLOBAL | L2 |
| Aritmeticas | VOID | 0 | FUNCTION | GLOBAL | L3 |
| Logicas2 | VOID | 0 | FUNCTION | GLOBAL | L4 |
| BitABit | VOID | 0 | FUNCTION | GLOBAL | L5 |
| Logicas | VOID | 0 | FUNCTION | GLOBAL | L6 |
| relaciones1 | VOID | 1 | FUNCTION | GLOBAL | L7 |
| relaciones2 | VOID | 1 | FUNCTION | GLOBAL | L8 |
| Relacionales | VOID | 0 | FUNCTION | GLOBAL | L9 |
| main | INT | 0 | FUNCTION | GLOBAL | main |

Error Report

```
19 sino: $a0 = $a0 - 1:
20 $ra = $ra + 1; #1 🌶 Titus @danii_mor
                                                            \times
                                                     21 goto fact;
22 $t100 = - ;
                        Lexical error <@> at
23 ret11:
                        line:2, column:1
24
        $a0 = $a0 + 1
                        Syntax error at line:22,
25
        $v0 = $a0 *
                        column:11
26
        if ($ra==0)
27
        $ra = $ra -
Titus>> Error Report Generated
```

Grammar

```
TenorC @danii_mor
                                                            🛭 TenorC @danii_...
                                                                                       >
             Edit
                  Run Help
                                                           s -> code
         Tab1 +
                                                           statement -> is_array_term = expression
                                                           expression -> expression + expression
         2
            int
                   var1 = 1;
                                                           expression -> expression - expression
         3
                   punteo = 0;
            int
                                                           expression -> expression * expression
                                                           expression -> expression / expression
         5
            void Declaracion()
                                                           expression -> expression & expression
         6
                                                           expression -> expression | expression
         7
                  printf ( "===== Metodo De
                                                           expression -> expression ^ expression
         8
                  int n4 = 2;
                                                           expression -> expression < expression
         9
                  double db4 = 0.0;
                                                           expression -> expression > expression
         10
                  double db1 = db4:
                                                           expression -> expression && expression
         11
                  char chr4 = 's';
                                                           expression -> expression || expression
         12
                  if(db1 == db4){
                                                           expression -> expression != expression
         13
                       printf ( "Declaraciones
                                                           expression -> expression == expression
         14
                       punteo = punteo + 5;
                                                           expression -> expression >= expression
         15
                  }else {
                                                           expression -> term
         16
                       printf ( "Problemas en
                                                       e1
                                                           term -> - expression
         17
                                                           expression -> ( INT) expression
                                                           expression -> ( expression )
                                                           expression -> ~ expression
                                                           expression ->! expression
              -----CALIFICACION------
                                                           term -> NUMBER
          ====== Metodo Declaracion =======
                                                           term -> NUMBER "." NUMBER
         Declaraciones Bien :D
                                                           factor -> STRING
                                                           factor -> is_array_term
               ========Aritmeticas===
          TenorC @danii mor
        File Edit Run Help
AST
                                                                                     X
                                         TenorC @danii_mor
         Tab1 +
           int
                 var1 = 1;
            int punteo = 0;
         5
            void Declaracion()
         6
         7
                 printf ( "====
         8
                 int n4 = 2;
                                                          FUNCTION
         9
                 double db4 = 0.0
         10
                 double db1 = db4
         11
                char chr4 = 's';
         12
                 if(db1 == db4) {
         13
                     printf ( "Dec.
         14
                     punteo = punt
                                                            Logicas2
                                             void
                                                                                  S
         15
                 }else {
         16
                     printf ( "Prol
                                                                                                olu
         17
         <
                  -CALIFICACION--
                                                 ASSIGN
                                                                             DECLARE
               ==== Metodo Declaracion
        Declaraciones Bien :D
                                     <
                        =Aritmeticas
```

Debugg

```
TenorC @danii_mor
   Edit Run Help
Tab1 +
   int
        var1 = 1;
3
   int
        punteo = 0;
5
                                        TenorC @danii_mor
6
       Execute Now:
                                   Executed Before:
7
                                                      ____
8
         Line: 1
                                       Line: 0
9
10
       double
                db1 = db4;
       char chr4 = 's';
11
       if(db1 == db4){
12
            printf ( "Declaraciones Bien :D\n" );
13
14
            punteo = punteo + 5;
15
       }else {
16
            printf ("Problemas en el metodo declar
17
        --CALIFICACION----
```

Info

```
TenorC @danii_mor
File Edit Run Help
Tab1 +
   int
         var1 = 1;
3
        punteo = 0;
5
                                 Х
        TenorC @danii_mor
6
     Augus intermediate
7
                                            laracion ======
     code by Engr. Espino
8
     TenorC 1.23.2a
9
     Developed by
10
     @danii_mor
11
      201314810
12
            printf ( "Declaraciones Bien :D\n" );
13
14
            punteo = punteo + 5;
15
        }else {
16
            printf ( "Problemas en el metodo declaracio
17
             ===relacionales1=====
```