



Fundamentos de Lenguaje C.

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de Programación

Grupo: 3

Alumna: Aguilar Lara Alexa Patricia

No. de Equipo de cómputo empleado: Máquina 44 Suecia

No. de lista o Brigada: 01

No. de cuenta: 316315515

Fecha de entrega: 30/09/2019

CALIFICACIÓN: _____

Observaciones:

Fundamentos de Lenguaje C

Objetivo:

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

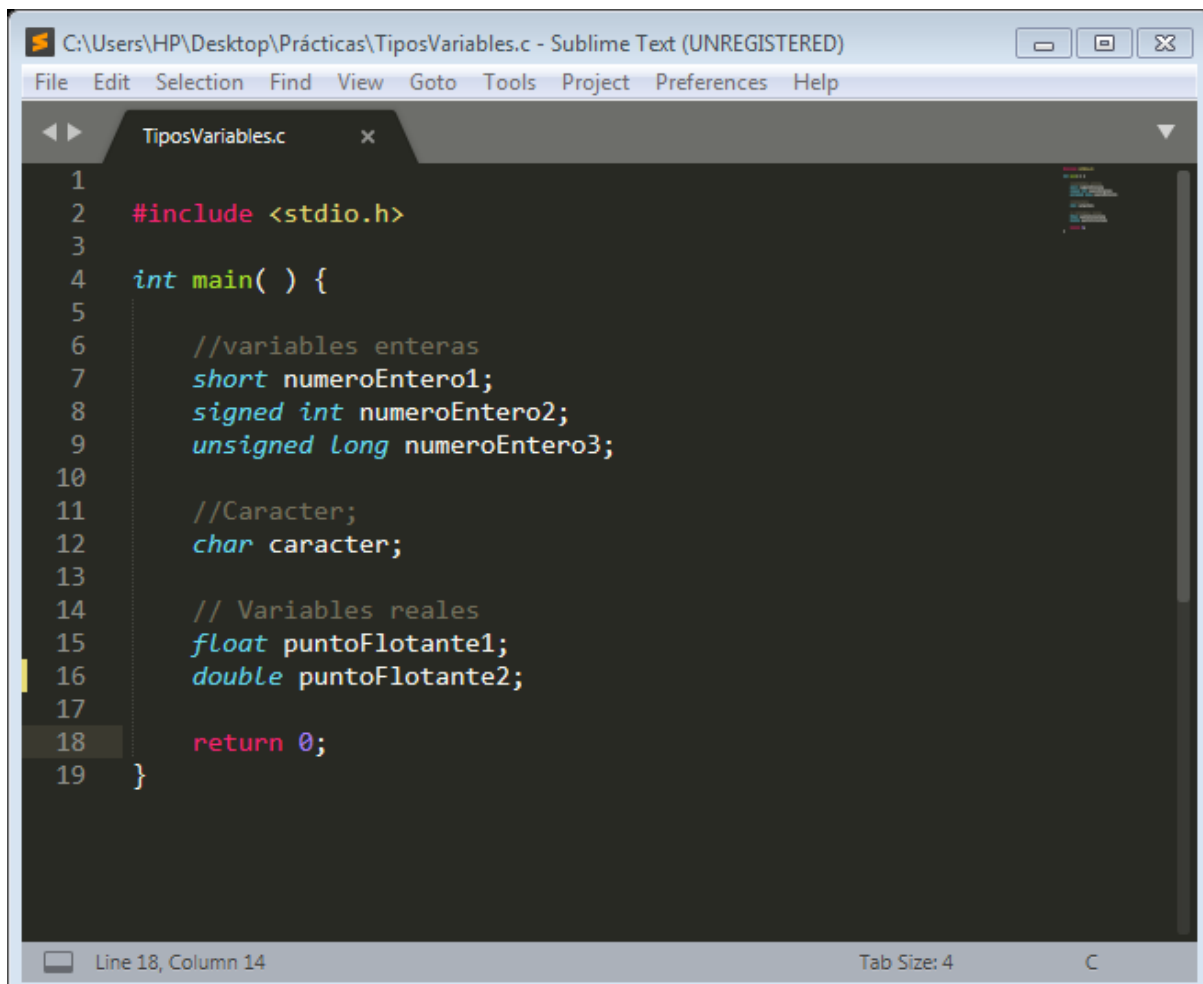
Tipos de variables

DATA TYPE	MEMORY (BYTES)	RANGE
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	$-(2^{63})$ to $(2^{63})-1$
unsigned long long int	8	0 to 18,446,744,073,709,551,615

Para los reales, se tienen también diferentes tipos de variables que asignan más bits para tener mayor rango y mayor precisión. Las variables reales siempre poseen signo.

Tipo	Bits	Valor Mínimo	Valor Máximo
<i>float</i>	32	3.4 E-38	3.4 E38
<i>double</i>	64	1.7 E-308	1.7 E308
<i>long double</i>	80	3.4 E-4932	3.4 E4932

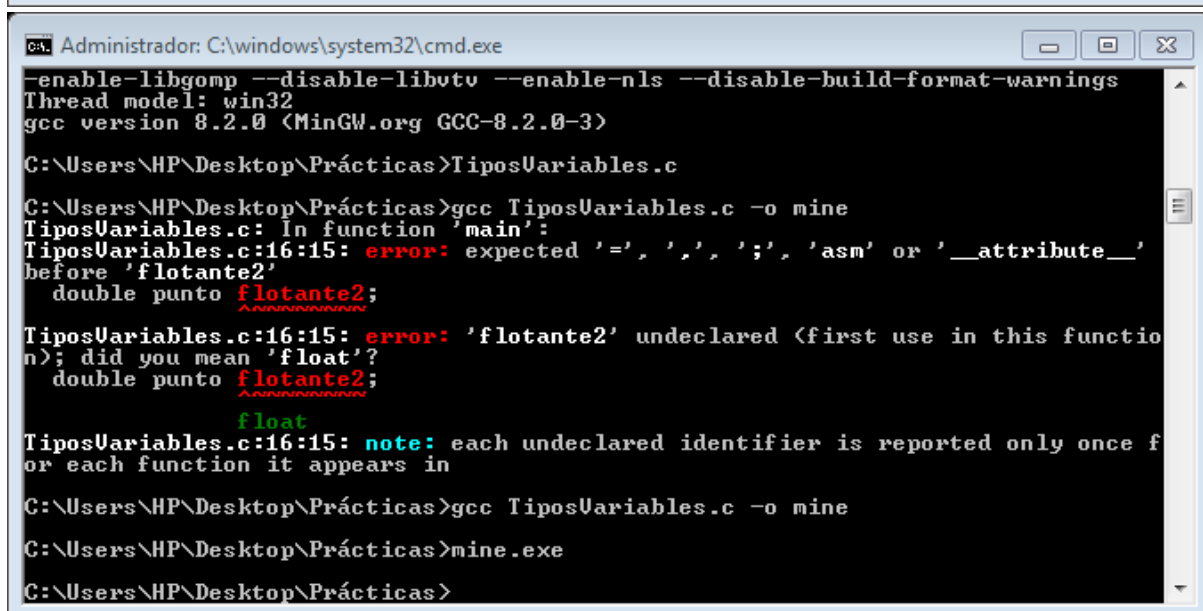
```
int main() {  
    // Variables enteras  
    short numeroEntero1;  
    signed int numeroEntero2;  
    unsigned long numeroEntero3;  
  
    // Caracter  
    char caracter;  
  
    // Variables reales  
    float puntoFlotante1;  
    double puntoFlotante2;  
  
    return 0;  
}
```



```
C:\Users\HP\Desktop\Prácticas\TiposVariables.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

TiposVariables.c x
1
2 #include <stdio.h>
3
4 int main( ) {
5
6     //variables enteras
7     short numeroEntero1;
8     signed int numeroEntero2;
9     unsigned long numeroEntero3;
10
11     //Caracter;
12     char caracter;
13
14     // Variables reales
15     float puntoFlotante1;
16     double puntoFlotante2;
17
18     return 0;
19 }
```

Line 18, Column 14 Tab Size: 4 C



```
C:\Windows\system32\cmd.exe
--enable-libgomp --disable-libvtv --enable-nls --disable-build-format-warnings
Thread model: win32
gcc version 8.2.0 (MinGW.org GCC-8.2.0-3)

C:\Users\HP\Desktop\Prácticas>TiposVariables.c

C:\Users\HP\Desktop\Prácticas>gcc TiposVariables.c -o mine
TiposVariables.c: In function 'main':
TiposVariables.c:16:15: error: expected '=', ',', ';', 'asm' or '__attribute__'
before 'flotante2'
    double punto flotante2;
                  ^~~~~~
TiposVariables.c:16:15: error: 'flotante2' undeclared (first use in this function);
did you mean 'float'?
    double punto flotante2;
                  ^~~~~~
TiposVariables.c:16:15: note: each undeclared identifier is reported only once for
each function it appears in

C:\Users\HP\Desktop\Prácticas>gcc TiposVariables.c -o mine

C:\Users\HP\Desktop\Prácticas>mine.exe

C:\Users\HP\Desktop\Prácticas>
```

En esta parte de la práctica repasamos los tipos de variables y los aplicamos al lenguaje C, elaborando nuestro primer programa, el cual, solo contenía los tipos de variables, ayudándonos a conocer como se aplican para este tipo de lenguaje. Se elaboró con ayuda de Sublime Text, capturando los datos proporcionados por el profesor.

Mostrar y leer

Tipo de dato	Especificador de formato
Entero	%d, %i, %ld, %li, %o, %x
Flotante	%f, %lf, %e, %g
Carácter	%c, %d, %i, %o, %x
Cadena de caracteres	%s

```
#include <stdio.h>

int main() {

    //Declaramos variables a leer
    int numeroEntrada;
    double realEntrada;

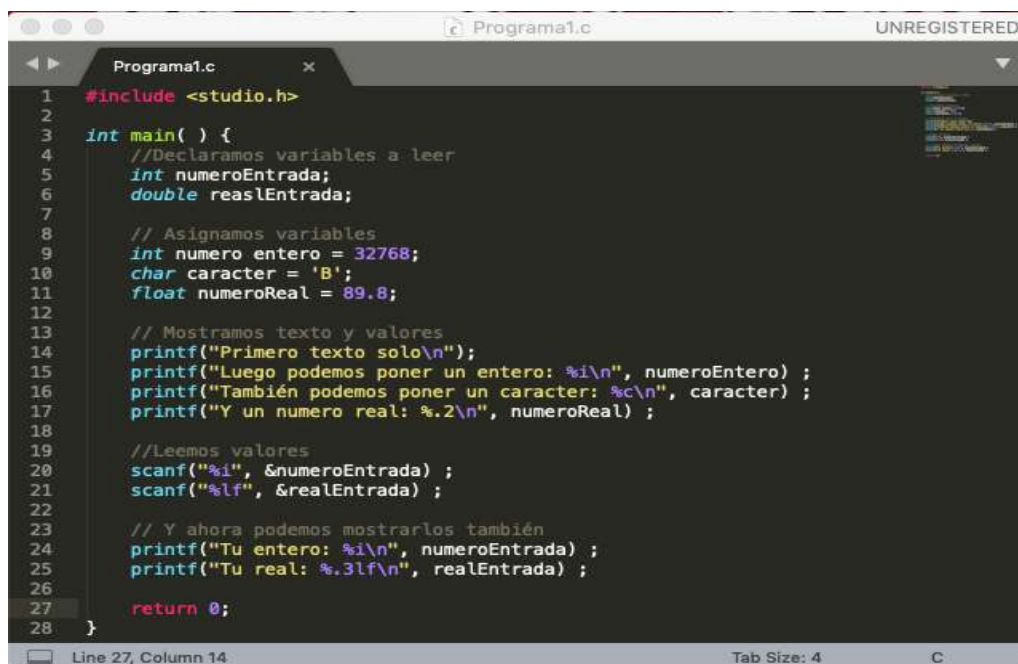
    // Asignamos variables
    int numeroEntero = 32768;
    char caracter = 'B';
    float numeroReal = 89.8;

    // Mostramos texto y valores
    printf("Primero texto solo\n");
    printf("Luego podemos poner un entero: %i\n", numeroEntero);
    printf("También podemos poner un carácter: %c\n", caracter);
    printf("Y un número real: %.2f\n", numeroReal);

    // Leemos valores
    scanf("%i", &numeroEntrada);
    scanf("%lf", &realEntrada);

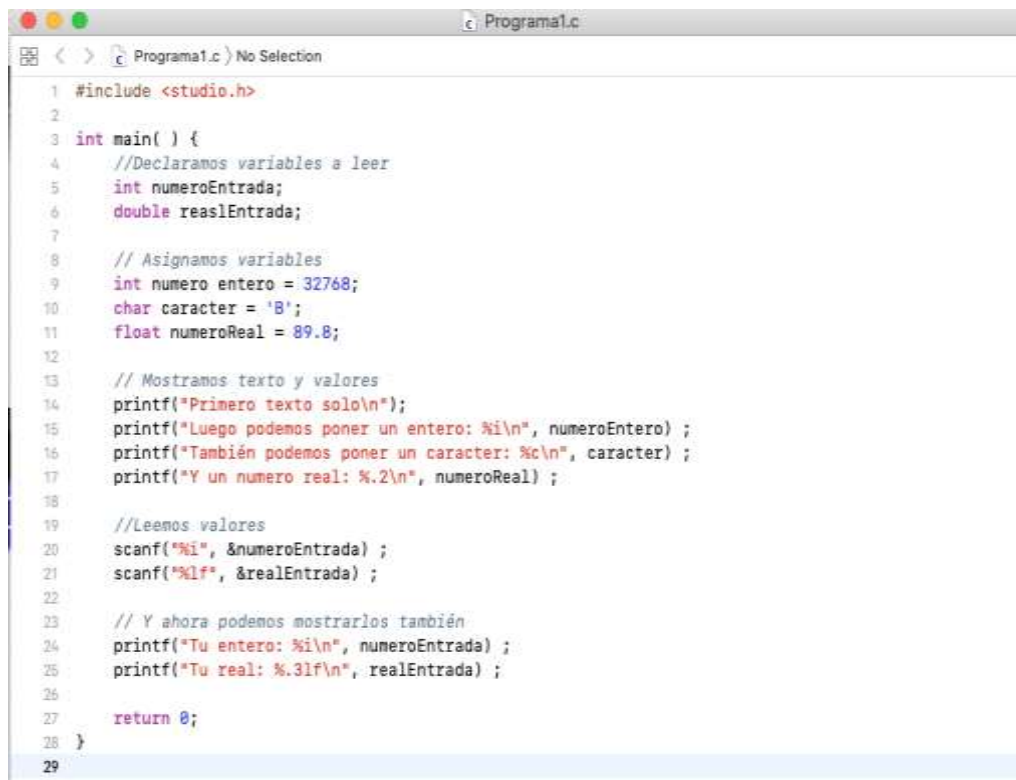
    // Y ahora podemos mostrarlos también
    printf("Tu entero: %i\n", numeroEntrada);
    printf("Tu real: %.3lf\n", realEntrada);

    return 0;
}
```

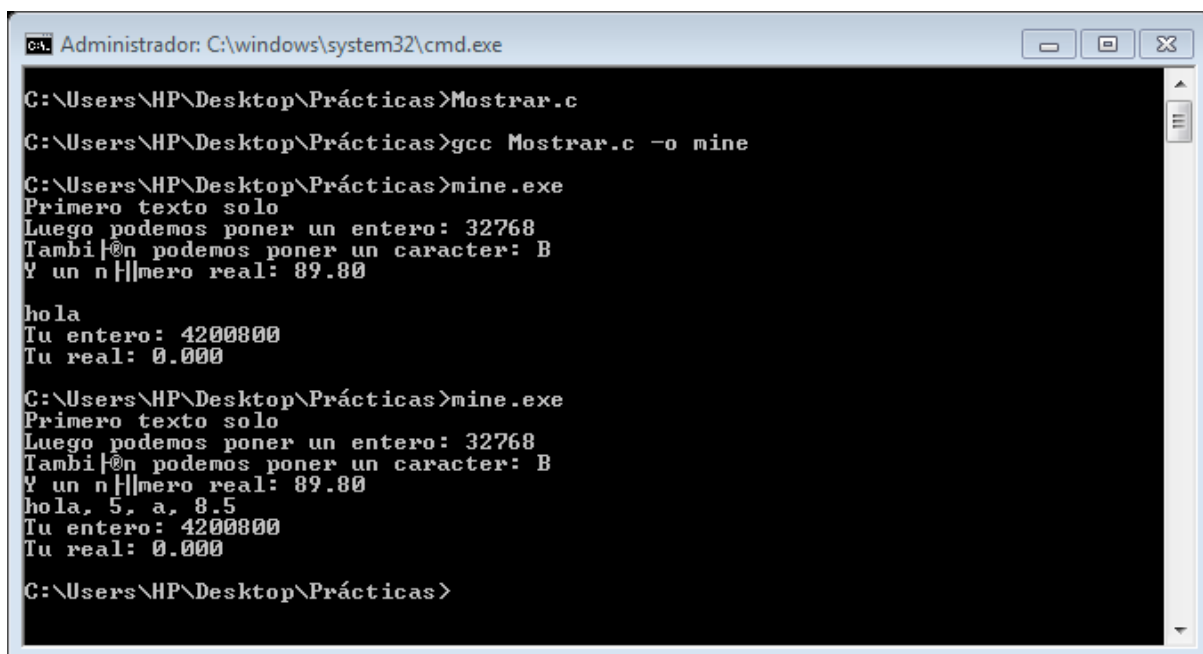


```
Programa1.c x
1 #include <stdio.h>
2
3 int main() {
4     //Declaramos variables a leer
5     int numeroEntrada;
6     double reaslEntrada;
7
8     // Asignamos variables
9     int numero entero = 32768;
10    char caracter = 'B';
11    float numeroReal = 89.8;
12
13    // Mostramos texto y valores
14    printf("Primero texto solo\n");
15    printf("Luego podemos poner un entero: %i\n", numeroEntero) ;
16    printf("También podemos poner un carácter: %c\n", caracter) ;
17    printf("Y un número real: %.2\n", numeroReal) ;
18
19    //Leemos valores
20    scanf("%i", &numeroEntrada) ;
21    scanf("%lf", &realEntrada) ;
22
23    // Y ahora podemos mostrarlos también
24    printf("Tu entero: %i\n", numeroEntrada) ;
25    printf("Tu real: %.3lf\n", realEntrada) ;
26
27    return 0;
28 }
```

Line 27, Column 14 Tab Size: 4 C



```
1 #include <stdio.h>
2
3 int main() {
4     //Declaramos variables a leer
5     int numeroEntrada;
6     double realEntrada;
7
8     // Asignamos variables
9     int numero entero = 32768;
10    char caracter = 'B';
11    float numeroReal = 89.8;
12
13    // Mostramos texto y valores
14    printf("Primero texto solo\n");
15    printf("Luego podemos poner un entero: %i\n", numeroEntero);
16    printf("También podemos poner un caracter: %c\n", caracter);
17    printf("Y un numero real: %.2\n", numeroReal);
18
19    //Leemos valores
20    scanf("%i", &numeroEntrada);
21    scanf("%lf", &realEntrada);
22
23    // Y ahora podemos mostrarlos también
24    printf("Tu entero: %i\n", numeroEntrada);
25    printf("Tu real: %.3lf\n", realEntrada);
26
27    return 0;
28 }
29
```



```
C:\Users\HP\Desktop\Prácticas>Mostrar.c
C:\Users\HP\Desktop\Prácticas>gcc Mostrar.c -o mine
C:\Users\HP\Desktop\Prácticas>mine.exe
Primero texto solo
Luego podemos poner un entero: 32768
Tambi@n podemos poner un caracter: B
Y un n||mero real: 89.80
hola
Tu entero: 4200800
Tu real: 0.000
C:\Users\HP\Desktop\Prácticas>mine.exe
Primero texto solo
Luego podemos poner un entero: 32768
Tambi@n podemos poner un caracter: B
Y un n||mero real: 89.80
hola, 5, a, 8.5
Tu entero: 4200800
Tu real: 0.000
C:\Users\HP\Desktop\Prácticas>
```

Esta parte de la práctica nos ayudó a conocer la manera en la que se muestran las variables en el lenguaje C para poder leerlas y mostrarlas, nuevamente capturamos los datos proporcionados por el profesor en la práctica en Sublime Text y se pudo compilar el programa, el cual nos dio como resultado las operaciones indicadas en el programa.

Operadores

Operador	Operación	Uso	Resultado
+	Suma	125.78 + 62.5	188.28
-	Resta	65.3 - 32.33	32.97
*	Multiplicación	8.27 * 7	57.75
/	División	15 / 4	3.75
%	Módulo	4 % 2	0

```
#include <stdio.h>

int main() {

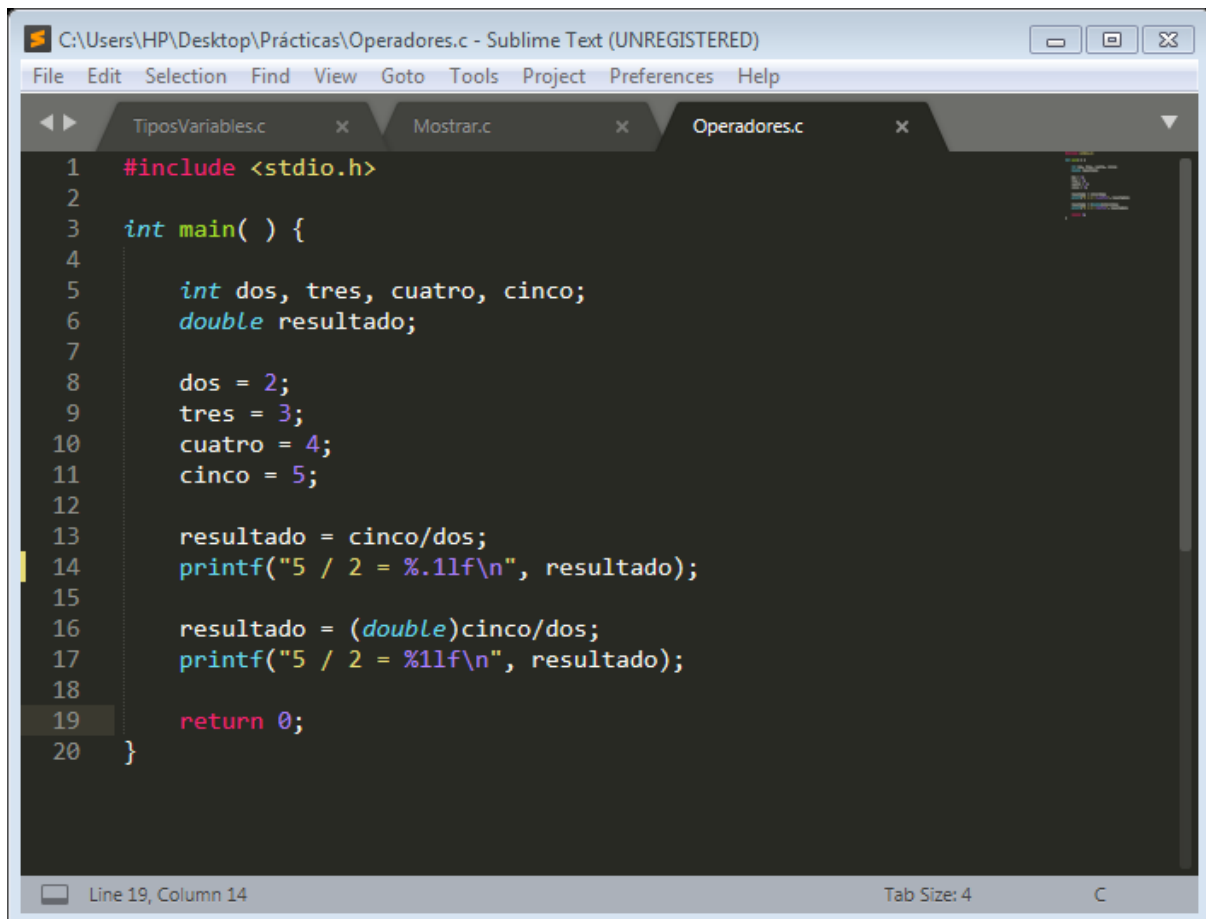
    int dos, tres, cuatro, cinco;
    double resultado;

    dos = 2;
    tres = 3;
    cuatro = 4;
    cinco = 5;

    resultado = cinco/dos;
    printf("5 / 2 = %.11f\n", resultado);

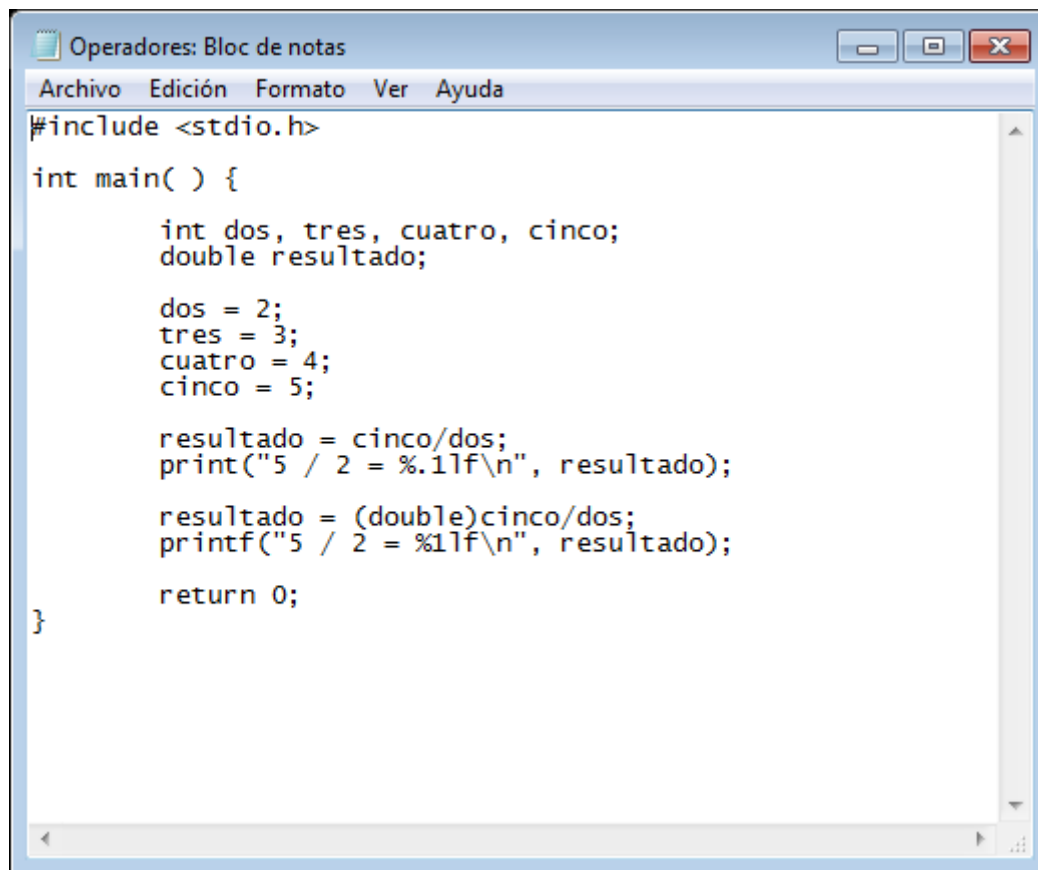
    resultado = (double)cinco/dos;
    printf("5 / 2 = %.11f\n", resultado);

    return 0;
}
```



```
C:\Users\HP\Desktop\Prácticas\Operadores.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
TiposVariables.c x Mostrar.c x Operadores.c x
1 #include <stdio.h>
2
3 int main( ) {
4
5     int dos, tres, cuatro, cinco;
6     double resultado;
7
8     dos = 2;
9     tres = 3;
10    cuatro = 4;
11    cinco = 5;
12
13    resultado = cinco/dos;
14    printf("5 / 2 = %.11f\n", resultado);
15
16    resultado = (double)cinco/dos;
17    printf("5 / 2 = %.11f\n", resultado);
18
19    return 0;
20 }
```

Line 19, Column 14 Tab Size: 4 C



```
#include <stdio.h>

int main( ) {

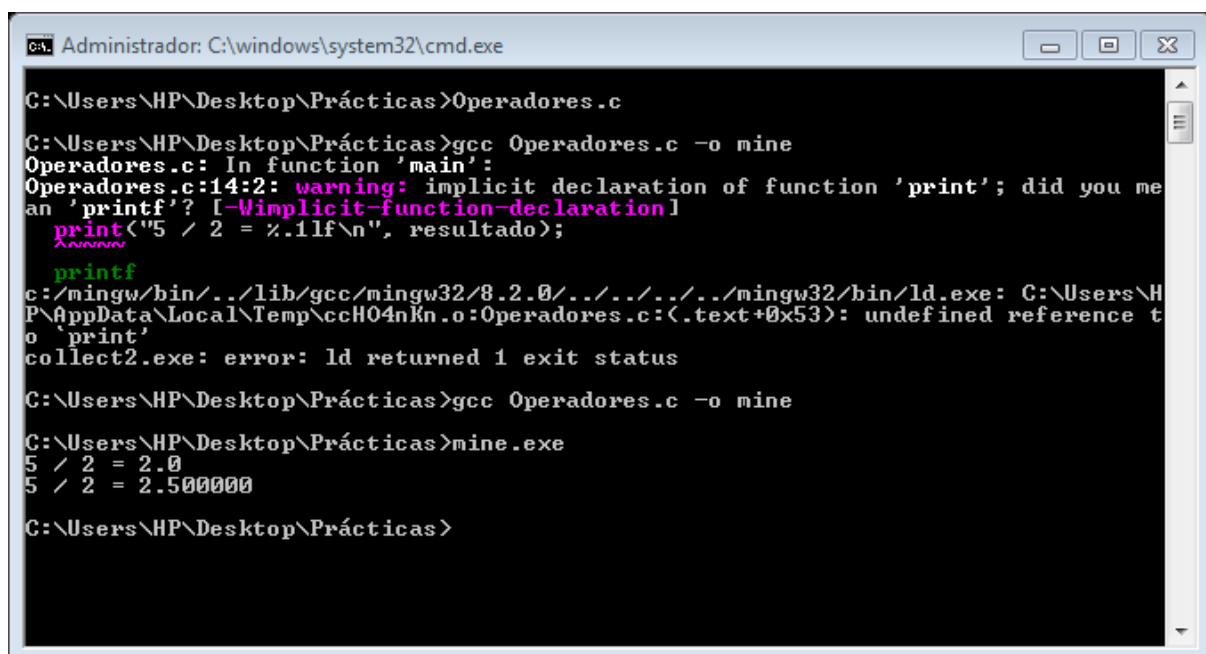
    int dos, tres, cuatro, cinco;
    double resultado;

    dos = 2;
    tres = 3;
    cuatro = 4;
    cinco = 5;

    resultado = cinco/dos;
    print("5 / 2 = %.1lf\n", resultado);

    resultado = (double)cinco/dos;
    printf("5 / 2 = %.1lf\n", resultado);

    return 0;
}
```



```
C:\Users\HP\Desktop\Prácticas>Operadores.c
C:\Users\HP\Desktop\Prácticas>gcc Operadores.c -o mine
Operadores.c: In function 'main':
Operadores.c:14:2: warning: implicit declaration of function 'print'; did you mean
an 'printf'? [-Wimplicit-function-declaration]
    print("5 / 2 = %.1lf\n", resultado);
    ^~~~~
c:/mingw/bin/../lib/gcc/mingw32/8.2.0/../../../../mingw32/bin/ld.exe: C:\Users\H
P\AppData\Local\Temp\ccH04nKn.o:Operadores.c:(.text+0x53): undefined reference t
o 'print'
collect2.exe: error: ld returned 1 exit status
C:\Users\HP\Desktop\Prácticas>gcc Operadores.c -o mine
C:\Users\HP\Desktop\Prácticas>mine.exe
5 / 2 = 2.0
5 / 2 = 2.500000
C:\Users\HP\Desktop\Prácticas>
```

Para esta parte nos familiarizamos con los operadores que ya conocíamos, aplicándolos nuevamente en C, en este caso, el programa capturado y obtenido, nos dio como resultado después de compilarlo las operaciones indicadas en el programa, en este caso divisiones.

Comparaciones

Operador	Operación	Uso	Resultado
<code>==</code>	Igual que	<code>'h' == 'H'</code>	Falso
<code>!=</code>	Diferente a	<code>'a' != 'b'</code>	Verdadero
<code><</code>	Menor que	<code>7 < 15</code>	Verdadero
<code>></code>	Mayor que	<code>11 > 22</code>	Falso
<code><=</code>	Menor o igual	<code>15 <= 22</code>	Verdadero
<code>>=</code>	Mayor o igual	<code>20 >= 35</code>	Falso

Operadores lógicos

Operador	Operación
<code>!</code>	No
<code>&&</code>	Y
<code> </code>	O

```
#include <stdio.h>

int main() {
    int num1, num2, res;
    char c1, c2;

    num1 = 7;
    num2 = 15;
    c1 = 'h';
    c2 = 'H';

    printf("%d num1 es menor a num2 ? -> %d\n", num1 < num2);
    printf("%d c1 es igual a c2 ? -> %d\n", c1 == c2);
    printf("%d c1 es diferente a c2 ? -> %d\n", c1 != c2);

    res = num1 < num2 && c1 == 'h';
    printf("%d num1 < num2 Y c1 es igual a 'h' ? -> %d\n", res);

    res = c1 == 's' || c2 == 'H';
    printf("%d c1 es igual a 's' O c2 a 'H' ? -> %d\n", res);

    return 0;
}
```


C:\Users\HP\Desktop\Prácticas\Operadoreslog.c - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

TiposVariables.c x Mostrar.c x Operadores.c x Operadoreslog.c x

```
3 int main( ) {
4
5     int num1, num2, res;
6     char c1, c2;
7
8     num1 = 7;
9     num2 = 15;
10    c1 = 'h';
11    c2 = 'H';
12
13    printf("¿ num1 es menor a num2 ? -> \t%d\n", num1<num2);
14    printf("¿ c1 es igual a c2 ? -> \t%d\n", c1==c2);
15    printf("¿ c1 es diferente a c2 ? -> \t%d\n", c1!=c2);
16
17    res = num1 < num2 && c1 == 'h';
18    printf("¿ num1 < num2 y c1 es igual a 'h' ? -> \t%d\n", res);
19
20    res = c1 == 's' || c2 == 'H';
21    printf("¿c1 es igual a 's' 0 c2 a 'H'? -> \t%d\n", res);
22
23    return 0;
24 }
```

Line 17, Column 21 Tab Size: 4 C

Operadoreslog: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>

int main( ) {

    int num1, num2, res;
    char c1, c2;

    num1 = 7;
    num2 = 15;
    c1 = 'h';
    c2 = 'H';

    printf("¿ num1 es menor a num2 ? -> \t%d\n", num1<num2);
    printf("¿ c1 es igual a c2 ? -> \t%d\n", c1==c2);
    printf("¿ c1 es diferente a c2 ? -> \t%d\n", c1!=c2);

    res = num1 < num2 && c1 == 'h';
    printf("¿ num1 < num2 y c1 es igual a 'h' ? -> \t%d\n", res);

    res = c1 == 's' || c2 == 'H';
    printf("¿c1 es igual a 's' 0 c2 a 'H'? -> \t%d\n", res);

    return 0;
}
```

```
Administrator: C:\windows\system32\cmd.exe
Operadoreslog.c: In function 'main':
Operadoreslog.c:17:15: error: 'num' undeclared (first use in this function); did
you mean 'num2'?
    res = num1 < num 2 && c1 == 'h';
                  ^~~
                  num2
Operadoreslog.c:17:15: note: each undeclared identifier is reported only once fo
r each function it appears in
Operadoreslog.c:17:18: error: expected ';' before numeric constant
    res = num1 < num 2 && c1 == 'h';
                  ^~
                  ;

C:\Users\HP\Desktop\Prácticas>gcc Operadoreslog.c -o mine
C:\Users\HP\Desktop\Prácticas>Operadoreslog.c
C:\Users\HP\Desktop\Prácticas>gcc Operadoreslog.c -o mine
C:\Users\HP\Desktop\Prácticas>mine.exe
num1 es menor a num2 ? ->      1
c1 es igual a c2 ? ->        0
c1 es diferente a c2 ? ->     1
num1 < num2 y c1 es igual a 'h' ? ->      1
c1 es igual a 's' 0 c2 a 'H'? ->      1

C:\Users\HP\Desktop\Prácticas>
```

Para esta última parte usamos los operadores lógicos, nuevamente aplicados en C, el programa obtenido y capturado nos dio como resultado, las comparaciones de diferentes valores capturados, tal y como se mostraba en el programa, obteniendo resultados de verdadero y falso.

En conclusión, después de realizar cada actividad, pudimos familiarizarnos con el lenguaje en C, brindándonos las herramientas necesarias para elaborar programas de este tipo, aprendimos también a compilar estos programas y verificar si son correctos o no para lo que queremos lograr con ellos, el uso de estos nos facilita la resolución de diferentes problemas, sabiendo cómo realizar correctamente un programa y de no ser correcto también se nos brindaron las herramientas necesarias para corregirlo.

Capturamos los programas en lenguaje C proporcionados en la práctica por el profesor, utilizamos una declaración de variables diferentes, reconociendo los enteros, reales y los caracteres, y llevamos a cabo funciones externas de entrada y salida que nos ayudaron a mostrar, leer y asignar datos a estas, por lo cual se cumple con el objetivo inicial de la práctica.