



# Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Esteban Pimentel Alarcon
<i>Asignatura :</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	7
<i>Integrante(s):</i>	Flores Rodriguez Ricardo Manuel Valencia Moya José Alejandro
<i>No. de Equipo de cómputo empleado:</i>	51 y 52
<i>No. de Lista o Brigada:</i>	12 y 51
<i>Semestre:</i>	Primero 2019-2020
<i>Fecha de entrega:</i>	03 de Octubre del 2019

	---
<i>Observaciones:</i>	

**CALIFICACIÓN:** \_\_\_\_\_

### **Introducción**

Después de conocer y aprender estudiar los pseudo códigos y los diagramas de flujo, es momento de poner en práctica nuestros conocimientos teóricos en la práctica realizando algunos programas en c y correrlos para comprobar su función.

### **Objetivos**

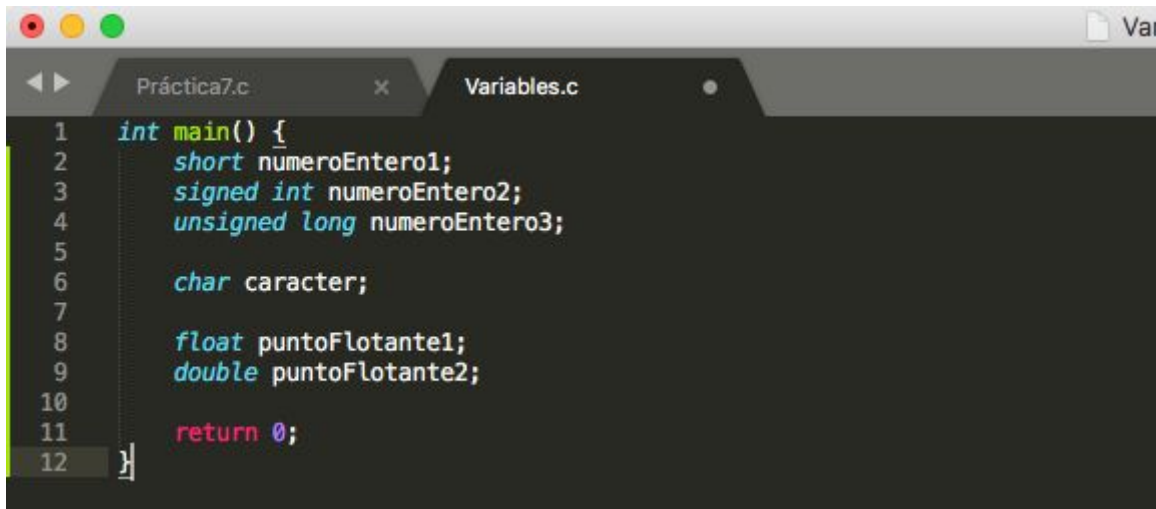
Elaborar programas en lenguaje C, utilizar las instrucciones de control de tipo de secuencia, para realizar la declaración de variables de diferentes tipos de datos.

En la práctica aprendimos que hay varios tipos de variables, para los reales, existen diferentes tipos de variables y siempre poseen signo por ejemplo:

Tipo	Bits
float	32
double	64
long double	80

## Actividad 1

En la primera actividad fuimos creando este programa con la ayuda del profesor y las diapositivas.

A screenshot of a code editor window. The window has two tabs: 'Práctica7.c' and 'Variables.c'. The 'Variables.c' tab is active. The code is written in C and defines a 'main' function. Inside the function, several variables are declared: 'numeroEntero1' (short), 'numeroEntero2' (signed int), 'numeroEntero3' (unsigned long), 'caracter' (char), 'puntoFlotante1' (float), and 'puntoFlotante2' (double). The function ends with 'return 0;'. The code is as follows:

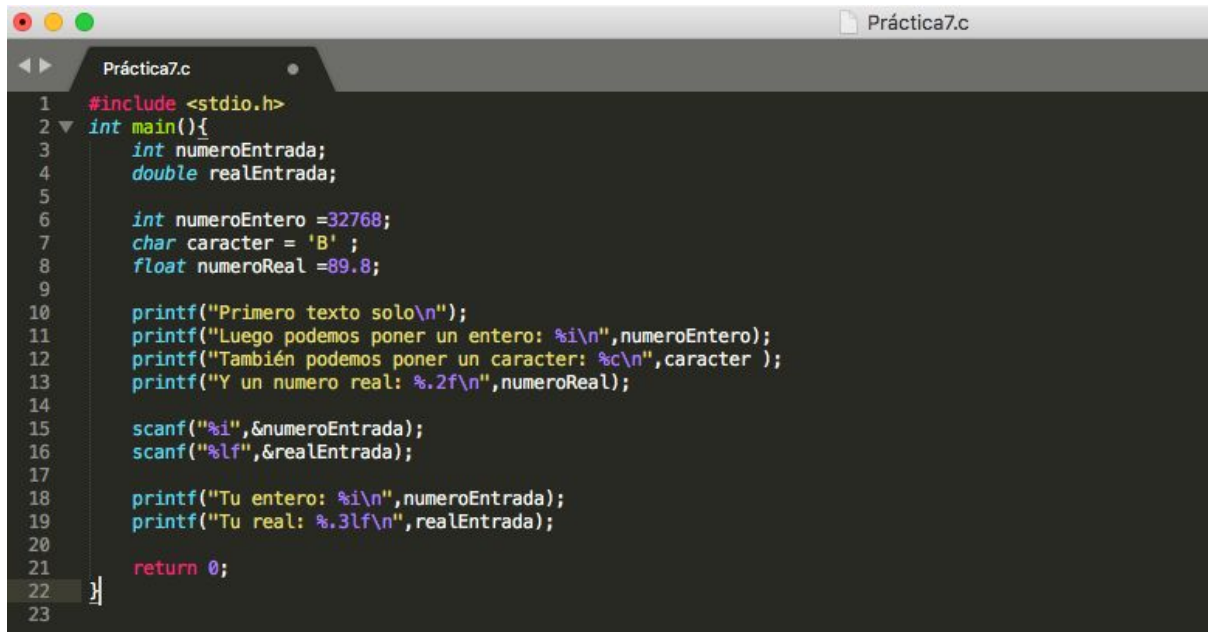
```
1 int main() {
2     short numeroEntero1;
3     signed int numeroEntero2;
4     unsigned long numeroEntero3;
5
6     char caracter;
7
8     float puntoFlotante1;
9     double puntoFlotante2;
10
11     return 0;
12 }
```

Y así como hay varios tipos de variables, tenemos que especificar cada una para los diferentes tipos de datos

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

## Actividad 2

En esta actividad ocupamos mas comandos y variables, incluyendo números enteros y caracteres, probando en la terminal lo escrito en el programa.



```
1  #include <stdio.h>
2  int main(){
3      int numeroEntrada;
4      double realEntrada;
5
6      int numeroEntero =32768;
7      char caracter = 'B' ;
8      float numeroReal =89.8;
9
10     printf("Primero texto solo\n");
11     printf("Luego podemos poner un entero: %i\n",numeroEntero);
12     printf("También podemos poner un caracter: %c\n",caracter );
13     printf("Y un numero real: %.2f\n",numeroReal);
14
15     scanf("%i",&numeroEntrada);
16     scanf("%lf",&realEntrada);
17
18     printf("Tu entero: %i\n",numeroEntrada);
19     printf("Tu real: %.3lf\n",realEntrada);
20
21     return 0;
22 }
23
```

```
[iMac-de-mac:Desktop mac$ ./main
Primero texto solo
Luego podemos poner un entero: 32768
También podemos poner un caracter: B
Y un numero real: 89.80
```

Y existen operaciones que son las siguientes:

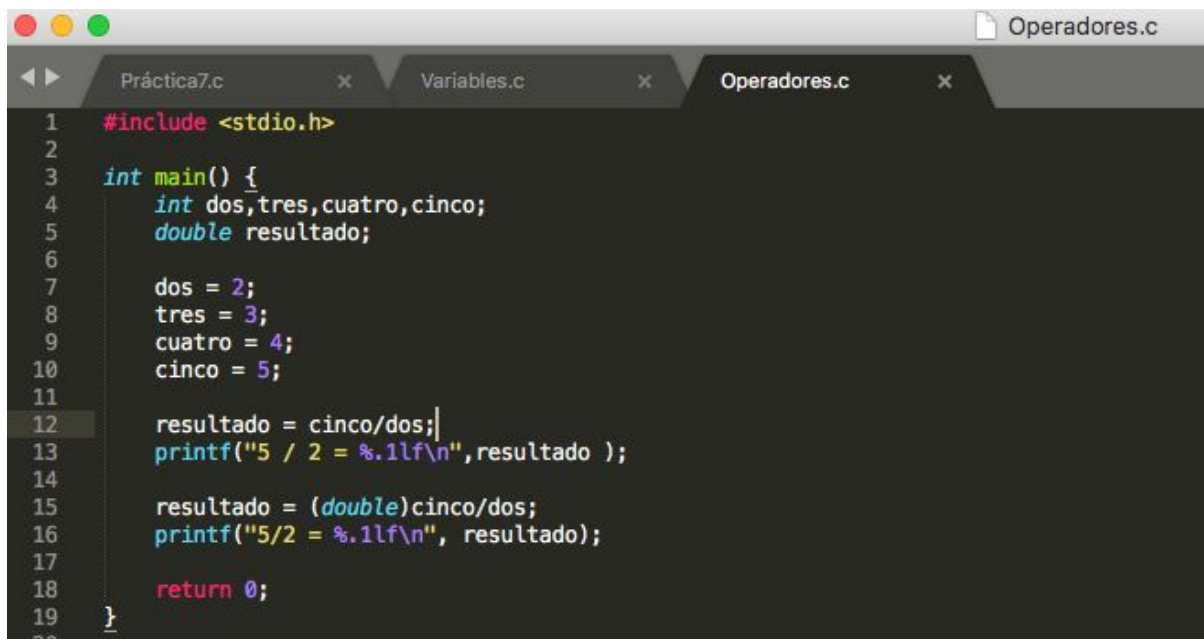
- Suma
- Resta
- Multiplicación
- División
- Módulo

Pero cada operación necesita un operador correspondiente los cuales son los siguientes:

Operador	Operación	Ejemplo de uso
+	Suma	114+16=130
-	Resta	256-81=175
*	Multiplicación	78*65=5070
/	División	15/4=3.75
%	Módulo	4%2=0

### Actividad 3

En esta actividad desarrollamos un programa en donde incluimos una operación lógica para ponerla en práctica la práctica de arriba



```

1  #include <stdio.h>
2
3  int main() {
4      int dos,tres,cuatro,cinco;
5      double resultado;
6
7      dos = 2;
8      tres = 3;
9      cuatro = 4;
10     cinco = 5;
11
12     resultado = cinco/dos;
13     printf("5 / 2 = %.1lf\n",resultado );
14
15     resultado = (double)cinco/dos;
16     printf("5/2 = %.1lf\n", resultado);
17
18     return 0;
19 }

```

```

iMac-de-mac:Desktop mac$ gcc Operadores.c -o main
iMac-de-mac:Desktop mac$ ./main
5 / 2 = 2.0
5/2 = 2.5

```

Estos son ejemplos de comparaciones y operadores lógicos:

Operador	Operación	Operador	Operación	Uso	Resultado
		==	Igual que	'h' == 'H'	Falso
		!=	Diferente a	'a' != 'b'	Verdadero
!	No	<	Menor que	7 < 15	Verdadero
&&	Y	>	Mayor que	11 > 22	Falso
	O	<=	Menor o igual	15 <= 22	Verdadero
		>=	Mayor o igual	20 >= 35	Falso

#### Actividad 4

En esta cuarta y última actividad realizamos otro programa incluyendo algunos operadores lógicos para practicar con ellos.

```

1  #include <stdio.h>
2
3  int main () {
4      int num1, num2, res;
5      char c1, c2;
6
7      num1=7;
8      num2=15;
9      c1= 'h';
10     c2= 'H';
11
12     printf("\n num1 es menor a num2 ? -> \t%d\n",num1<num2 );
13     printf("\n c1 es igual a c2 ? -> \t%d\n",c1==c2 );
14     printf("\n c1 es diferente a c2 ? -> \t%d\n",c1!=c2 );
15
16     res = num1 < num2 && c1 == 'h' ;
17     printf("\n num1 < num2 Y c1 es igual a 'h' ? -> \t%d\n",res );
18
19     res = c1 == 's' || c2 == 'H';
20     printf("\n c1 es igual a 's' 0 c2 a 'H' ? -> \t%d\n",res );
21
22     return 0;
23
24 }
25

```

## **Conclusiones**

Los textos en c son más cómodos y fáciles a comparación de los pseudo códigos o los diagramas de flujo, y es una parte fundamental para la programación básica ya que en base a códigos en c se pueden crear muchas cosas.

Estos programas que se crean a partir de c te ayudan a corroborar tus errores con los colores que te marca en cada renglón.