



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Alejandro Esteban Pimentel Alarcon

*Profesor:*

Fundamentos de programación

*Asignatura:*

*Grupo:*

07

*No de Práctica(s):*

Ameyalli Jocelyn Cureño Arvizu

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

10

*No. de Lista o Brigada:*

2020-1

*Semestre:*

03- octubre- 2019

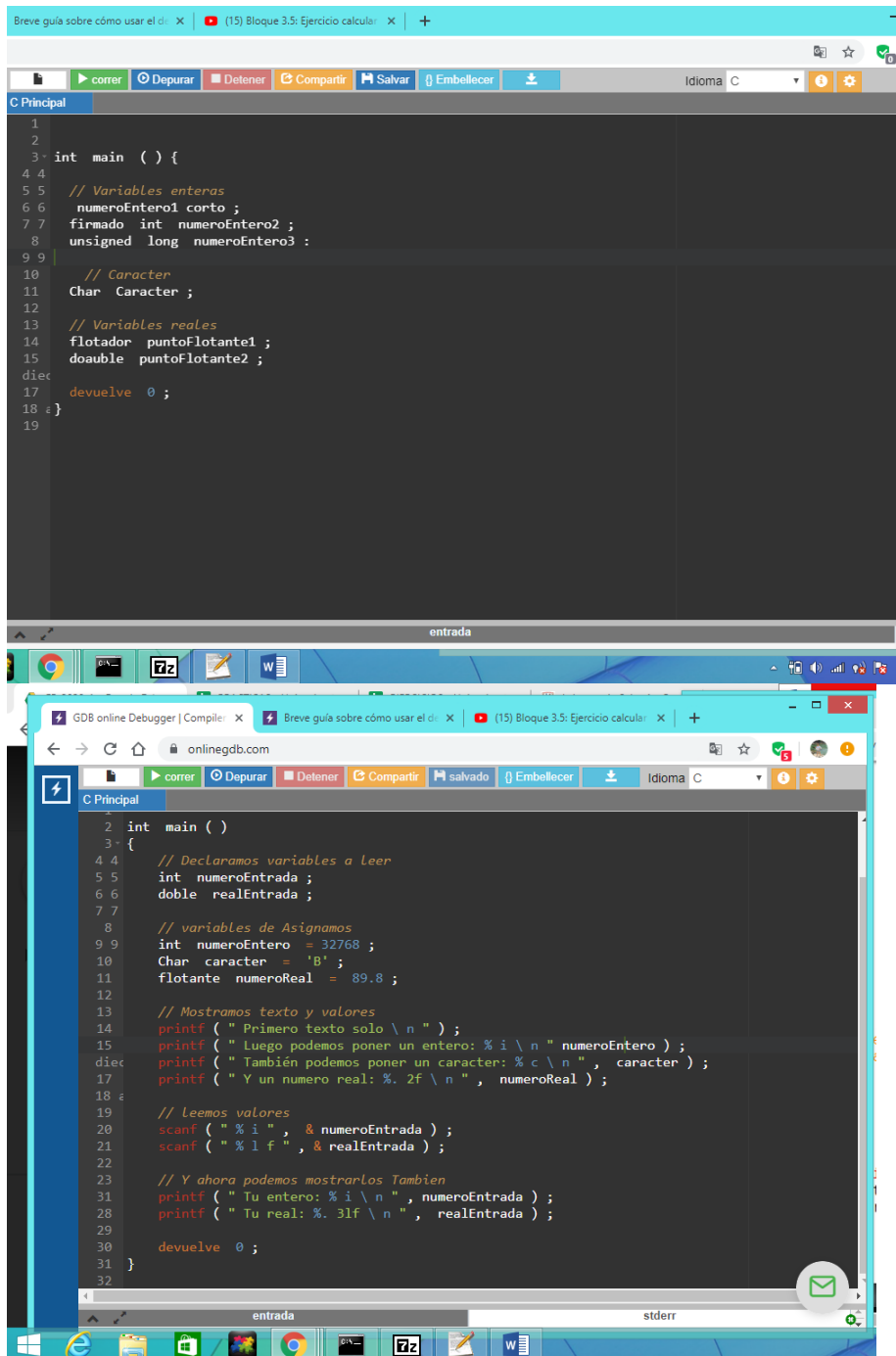
*Fecha de entrega:*

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## 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 demostrar valores de variables y expresiones.



```
1  
2  
3 int main ( ) {  
4  
5 // Variables enteras  
6 numeroEntero1 corto ;  
7 firmado int numeroEntero2 ;  
8 unsigned long numeroEntero3 :  
9  
10 // Caracter  
11 Char Caracter ;  
12  
13 // Variables reales  
14 flotador puntoFlotante1 ;  
15 doouble puntoFlotante2 ;  
16  
17 devuelve 0 ;  
18 }  
19
```

```
2 int main ( )  
3 {  
4 // Declaramos variables a Leer  
5 int numeroEntrada ;  
6 doble realEntrada ;  
7  
8 // variables de Asignamos  
9 int numeroEntero = 32768 ;  
10 Char caracter = 'B' ;  
11 flotante 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: %. 2f \n " , numeroReal ) ;  
18  
19 // Leemos valores  
20 scanf ( " %i " , & numeroEntrada ) ;  
21 scanf ( " %l f " , & realEntrada ) ;  
22  
23 // Y ahora podemos mostrarlos Tambien  
24 printf ( " Tu entero: %i \n " , numeroEntrada ) ;  
25 printf ( " Tu real: %. 3lf \n " , realEntrada ) ;  
26  
27 devuelve 0 ;  
28 }  
29
```

Debugger | Compiler | Breve guía sobre cómo usar el d... | (15) Bloque 3.5: Ejercicio calcular

onlinegdb.com

DB beta  
en línea  
allio907  
yecto  
ps  
nación  
amación  
on  
46,9K  
ck  
do con  
o de  
ara el  
recuentes  
e uso  
al GDB  
idad  
en línea

C Principal

```
1 // valores /
2
3 #include <stdio.h>
4
5 int main ( ) {
6
7     int dos , tres , cuatro , cinco ;
8     doble resultado ;
9
10    dos = 2 ;
11    tres = 3 ;
12    cuatro = 4 ;
13    cinco = 5 ;
14
15    resultado = cinco / dos ;
16    printf ( " 5/2 = %.1lf \n " , resultado ) ;
17
18    resultado = ( doble )
19
20    devuelve 0 ;
21 }
22
```

entrada stderr

Debugger | Compiler | Breve guía sobre cómo usar el d... | (16) Bloque 3.5: Ejercicio calcular

onlinegdb.com

DB beta  
en línea para c...  
allio907  
yecto  
ps  
nación  
amación  
on  
46,9K  
ck  
do con  
o de  
ara el  
es de

C Principal

```
1 // operadores logicos /
2
3 #include <stdio.h>
4
5 int main ( ) {
6
7     int num1 , num2 , res ;
8     char c1 , c2 ;
9
10    num1 = 7 ;
11    num2 = 15 ;
12    c1 = 'h' ;
13    c2 = 'H' ;
14
15    printf ( " ¿num1 es menor a num2? -> \t%d \n " , num1 < num2 ) ;
16    printf ( " ¿c1 es igual a c2? -> \t%d \n " , c1 == c2 ) ;
17    printf ( " ¿c1 es diferente a c2? -> \t%d \n " , c1 != c2 ) ;
18
19    res = num1 < num2 && c1 == 'h' ;
20    printf ( " ¿num1 < num2 Y c1 es igual a 'h'? -> \t%d \n " , res ) ;
21
22    res = c1 == 's' || c2 == 'H' ;
23    printf ( " ¿c1 es igual a 's' O c2 a 'H'? -> \t%d \n " , res ) ;
24
25    devuelve 0 ;
26 }
27
28
```

entrada stderr

## CONCLUSIONES

Para empezar a programas es bueno identificar los tipos de variables y en qué caso usar cada una (tanto en las reales como en las enteras); los cuales, en su mayoría cuentan con una más formas para representarlos.

Los operadores es otro de los símbolos importantes en la programación en C y también es importante saber identificar y no confundir del pseudocódigo y no es menos importante saber que los operadores tienen diferente función de las comparaciones. Todo esto para completar una buena edificación del programa.

Para poder verificar que el programa tenga la estructura correcta debemos guardar el texto recordando siempre poner al final de nombre del archivo un `.c` para luego correrlo en la terminal