



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Alejandro Pimentel Alarco

*Profesor:*

Fundamentos de Programacion

*Asignatura:*

3

*Grupo:*

7

*No de Práctica(s):*

Badillo Ruiz Evangelina

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

39

3069

*No. de Lista o Brigada:*

1

*Semestre:*

4 de octubre del 2019

*Fecha de entrega:*

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# PRÁCTICA 7

**Badillo Ruiz Evangelina No.cuenta 3069,nl. 4**

# OBJETIVO

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

# INTRODUCCIÓN

- Aprenderos a mostrar y leer las variables que ingresemos al lenguaje C
- Los tipos de valores que le podemos asignar a las variables y sus abreviaturas
- Como utilizar los operadores lógicos (and, or y not)
- Como utilizar los operadores (suma, resta, multiplicación y división)
- Todo en lenguaje C, en el editor de texto de nuestra preferencia ,pero lo compilaremos en la terminal de Linux.

# TIPO DE VARIABLES EN ENTEROS

## 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

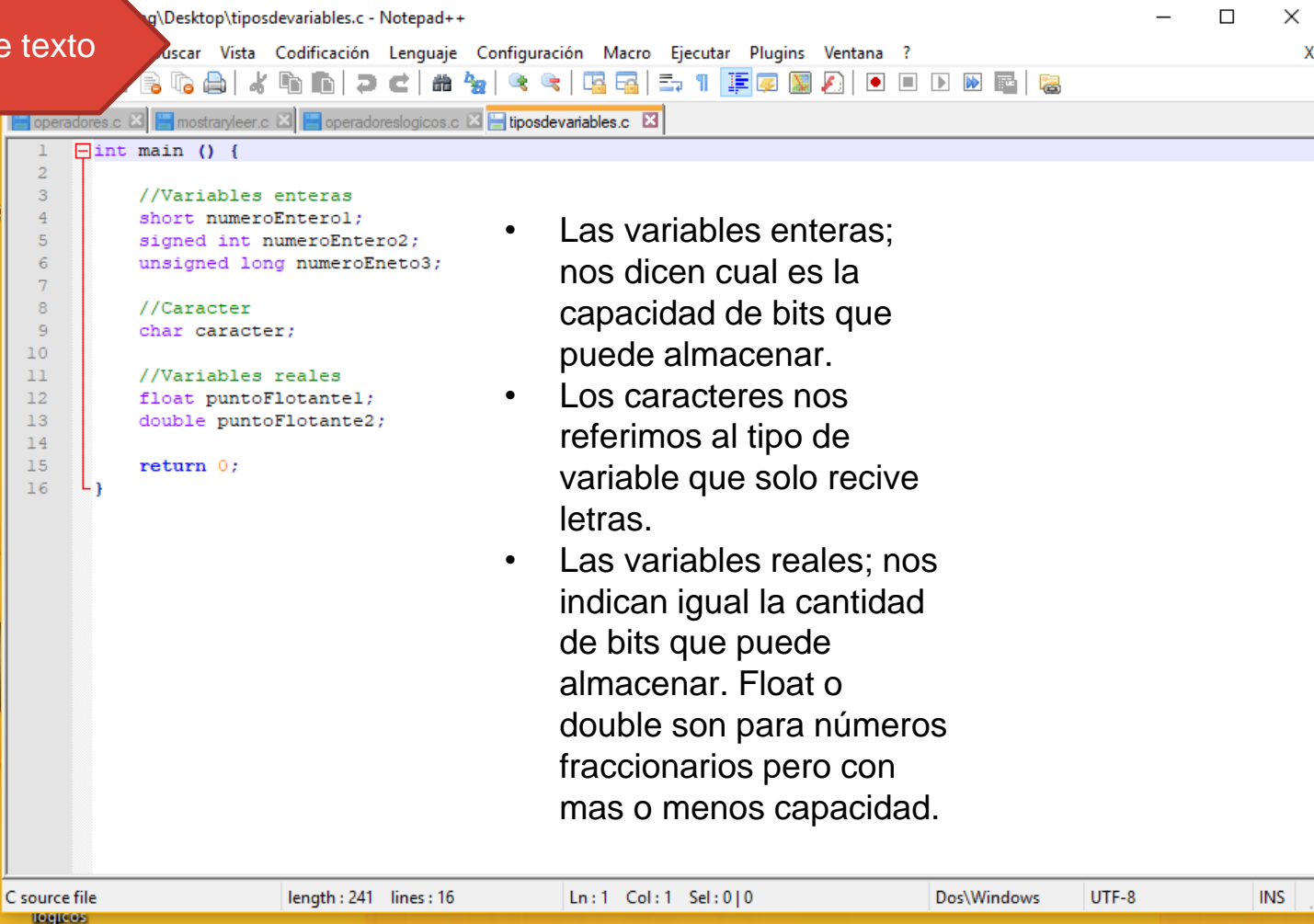
# TIPO DE VARIABLES EN REALES

## Tipos de variables

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.

<i>Tipo</i>	<i>Bits</i>	<i>Valor Mínimo</i>	<i>Valor Máximo</i>
<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

## En el editor de texto



The screenshot shows a Notepad++ window titled "C:\Desktop\tiposdevariables.c - Notepad++". The menu bar includes "Buscar", "Vista", "Codificación", "Lenguaje", "Configuración", "Macro", "Ejecutar", "Plugins", and "Ventana". The toolbar contains icons for file operations, editing, and execution. The tab bar shows four open files: "operadores.c", "mostranyleer.c", "operadoreslogicos.c", and "tiposdevariables.c". The main editor area displays the following C code:

```
1 int main () {  
2  
3     //Variables enteras  
4     short numeroEntero1;  
5     signed int numeroEntero2;  
6     unsigned long numeroEntero3;  
7  
8     //Caracter  
9     char caracter;  
10  
11    //Variables reales  
12    float puntoFlotante1;  
13    double puntoFlotante2;  
14  
15    return 0;  
16 }
```

The status bar at the bottom indicates "C source file", "length: 241 lines: 16", "Ln: 1 Col: 1 Sel: 0 | 0", "Dos\Windows", "UTF-8", and "INS".

- Las variables enteras; nos dicen cual es la capacidad de bits que puede almacenar.
- Los caracteres nos referimos al tipo de variable que solo recibe letras.
- Las variables reales; nos indican igual la cantidad de bits que puede almacenar. Float o double son para números fraccionarios pero con mas o menos capacidad.

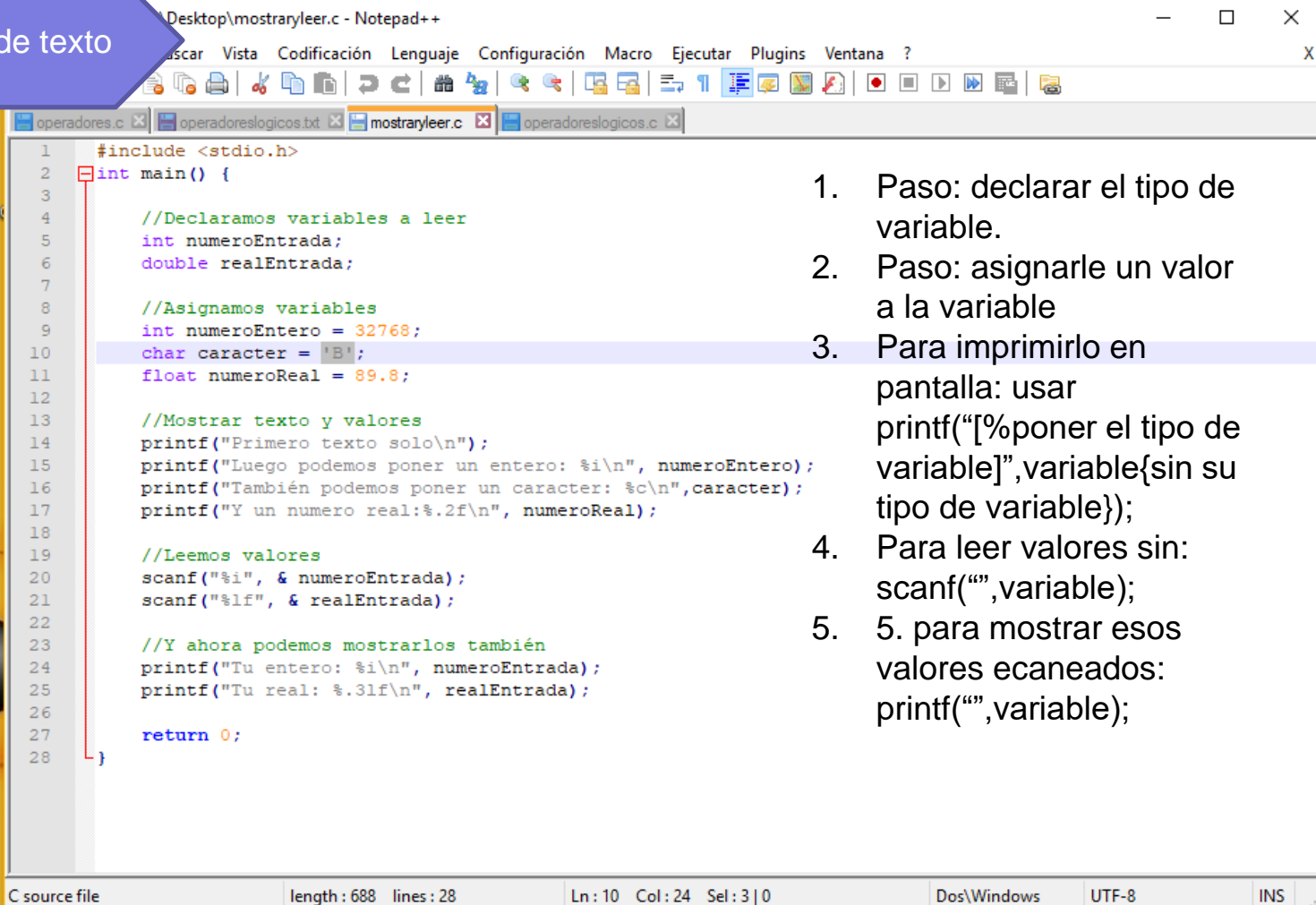
# TIPO DE DATO SIMPLIFICADO

## Mostrar y Leer

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



## En el editor de texto



```
#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;

    //Mostrar texto y valores
    printf("Primero texto solo\n");
    printf("Luego podemos poner un entero: %i\n", numeroEntero);
    printf("También podemos poner un caracter: %c\n", caracter);
    printf("Y un numero 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;
}
```

1. Paso: declarar el tipo de variable.
2. Paso: asignarle un valor a la variable
3. Para imprimirlo en pantalla: usar `printf("[%poner el tipo de variable]", variable{sin su tipo de variable});`
4. Para leer valores sin: `scanf("", variable);`
5. 5. para mostrar esos valores ecaneados: `printf("", variable);`

C source file      length : 688   lines : 28      Ln : 10   Col : 24   Sel : 3 | 0      Dos\Windows      UTF-8      INS

# MOSTRAR Y LEER

Cuando se compila

```
evangelina@evangelina-VirtualBox ~/Descargas $ ./mostraryleer
```

Primero texto solo

Luego podemos poner un entero: 32768

También podemos poner un caracter: B

Y un numero real:89.80

—

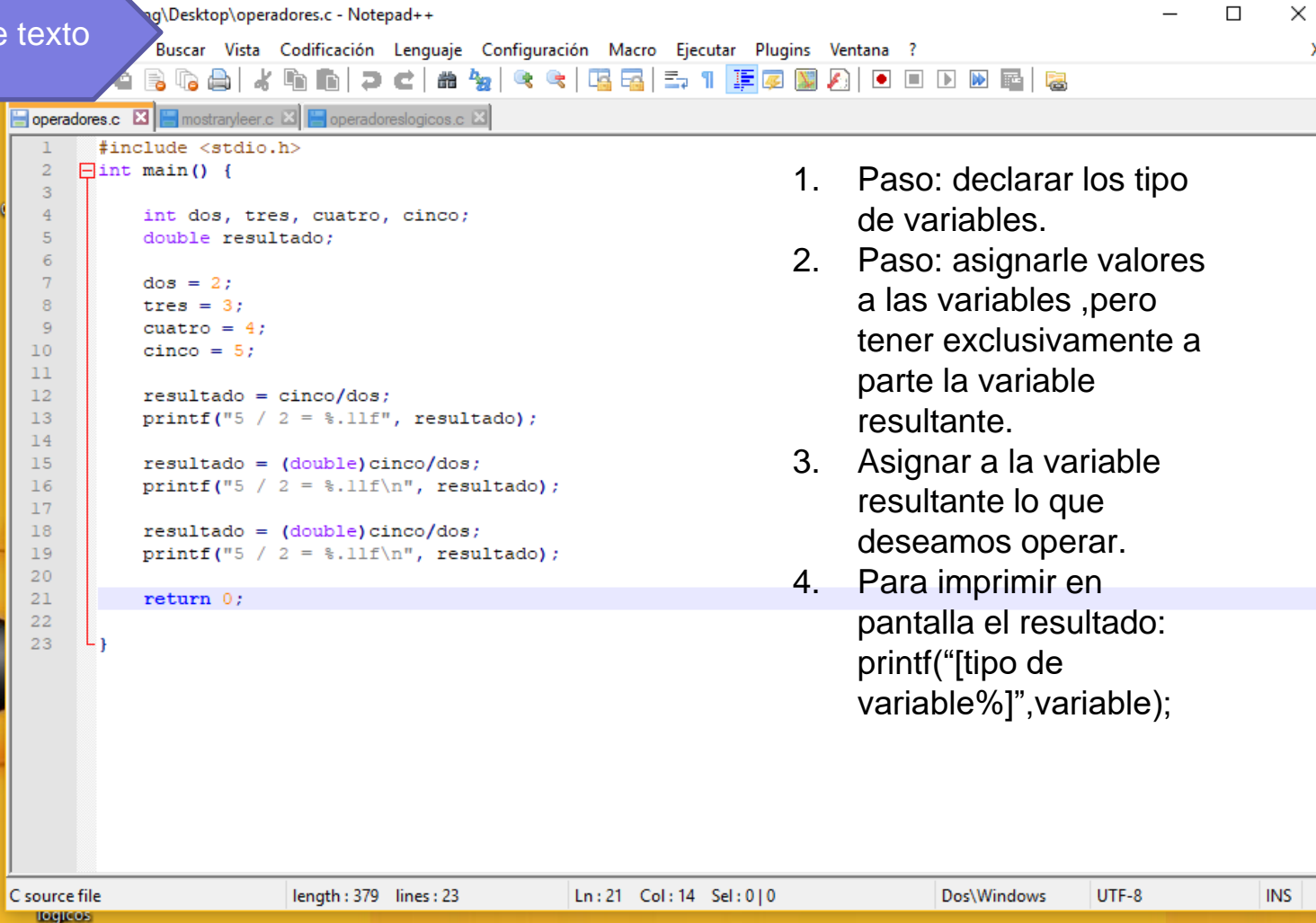


# OPERADOR

## Operadores

<i>Operador</i>	<i>Operación</i>	<i>Uso</i>	<i>Resultado</i>
+	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

## En el editor de texto



```
1  #include <stdio.h>
2  int main() {
3
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 = %.11f", resultado);
14
15     resultado = (double)cinco/dos;
16     printf("5 / 2 = %.11f\n", resultado);
17
18     resultado = (double)cinco/dos;
19     printf("5 / 2 = %.11f\n", resultado);
20
21     return 0;
22
23 }
```

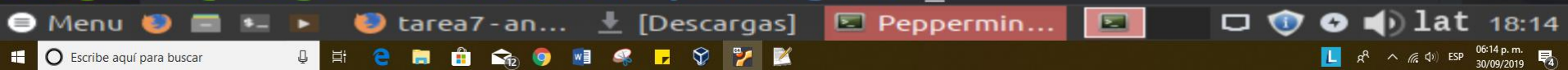
1. Paso: declarar los tipo de variables.
2. Paso: asignarle valores a las variables ,pero tener exclusivamente a parte la variable resultante.
3. Asignar a la variable resultante lo que deseamos operar.
4. Para imprimir en pantalla el resultado: `printf("[tipo de variable%]",variable);`

C source file      length: 379 lines: 23      Ln: 21 Col: 14 Sel: 0|0      Dos\Windows      UTF-8      INS

# OPERADORES

Cuando se compila

```
evangelina@evangelina-VirtualBox ~/Descargas $ ls
mostraryleer      operadores.c      tiposdevariables-1
mostraryleer.c    operadoreslogicos.c  tiposdevariables-1.c
evangelina@evangelina-VirtualBox ~/Descargas $ gcc operadores.c -o operadores
evangelina@evangelina-VirtualBox ~/Descargas $ ./operadores
5 / 2 = 2.05 / 2 = 2.5
5 / 2 = 2.5
evangelina@evangelina-VirtualBox ~/Descargas $ _
```



# COMPARACIÓN

## Comparaciones

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

# OPERADORES LÓGICOS

## Operadores lógicos

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

## En el editor de texto

q:\Desktop\operadoreslogicos.c - Notepad++

Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?

operadores.c x operadoreslogicos.txt x mostrayleer.c x operadoreslogicos.c x

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

1. Paso: declarar el tipo de variable.
2. Paso: asignarle un valor a la variable
3. Para imprimirlo en pantalla: usar printf("[%poner el tipo de variable]",variable[<,>==,!=,&&,||]variable);

Podemos en el mismo printf meter el operador.

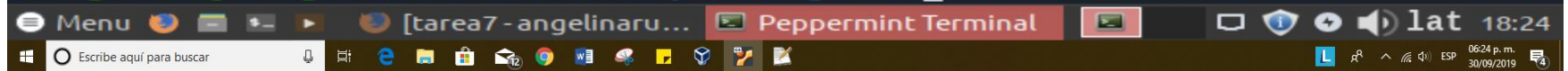
C source file length: 519 lines: 23 Ln: 23 Col: 2 Sel: 0 | 0 Dos\Windows UTF-8 INS



# OPERADORES LÓGICOS

Cuando se compila

```
evangelina@evangelina-VirtualBox ~/Descargas $ ls
mostraryleer      operadores      operadoreslogicos  tiposdevariables-1
mostraryleer.c    operadores.c    operadoreslogicos.c tiposdevariables-1.c
evangelina@evangelina-VirtualBox ~/Descargas $ ./operadoreslogicos
¿ 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
evangelina@evangelina-VirtualBox ~/Descargas $ _
```



# CONCLUSIÓN

- Aprendimos en esta práctica primero a declarar los tipos de variables que necesitamos, una vez decidido con que tipo de números vamos a utilizar ahora le pondremos nombre a nuestra variable.
- Si deseamos asignarle un valor a nuestra variable solo tenemos que reescribirla con todo y su tipo de variable e igualarla a lo que queremos asignarle.
- Para imprimir en pantalla se utiliza: `printf(“%[tipo de variable]”,variable);`
- Para escanear un valor se utiliza: `scanf(“%[tipo de variable]”,variable);`
- Para operar , asigna variable, nombre de la variable, asignar operación a la variable, utilizar `printf(“%[tipo de variable]”,variable);` o , asigna variable, nombre de la variable, `scanf(“%[tipo de variable]”,variable);`, utilizar `printf(“%[tipo de variable]”,variable);`
- Para operadores lógicos, dentro de `printf` hacer la operación lógica.