



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* M.I. Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de Programación

*Grupo:* 3 Bloque: 136

*No de Práctica(s):* 7 Fundamentos de Lenguaje C

*Integrante(s):* Carranza Ochoa José David

*No. de Equipo de  
cómputo empleado:* No aplica

*No. de Lista o Brigada:* 6

*Semestre:* 2021-1

*Fecha de entrega:* 23/11/2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo:

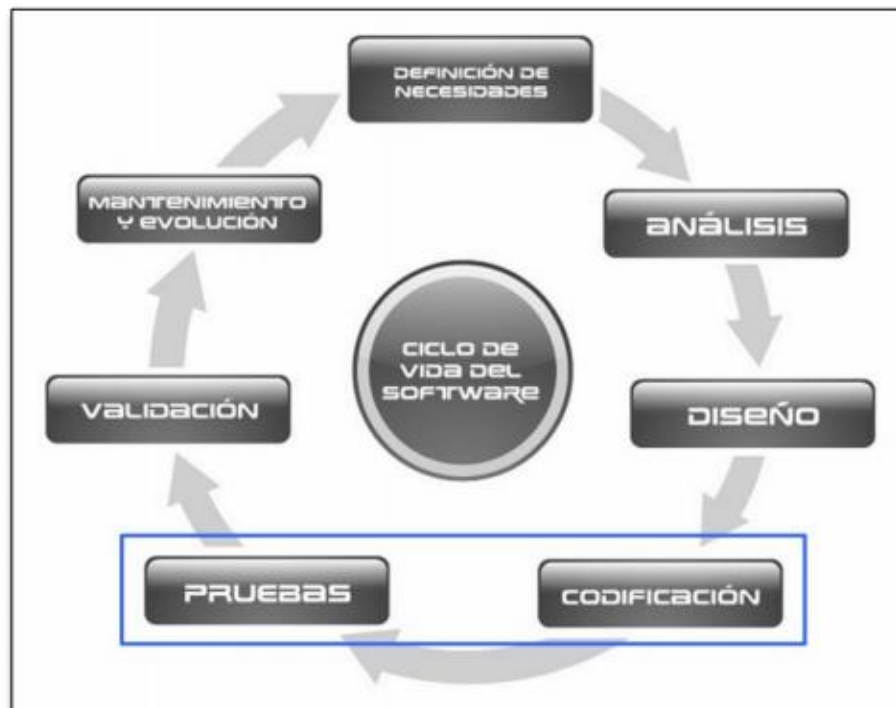
Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

# Introducción

Una vez que un problema dado ha sido analizado (se identifican los datos de entrada y la salida deseada), que se ha diseñado un algoritmo que lo resuelva de manera eficiente (procesamiento de datos), y que se ha representado el algoritmo de manera gráfica o escrita (diagrama de flujo o pseudocódigo) se puede proceder a la etapa de codificación.

La codificación se puede realizar en cualquier lenguaje de programación estructurada, como lo son Pascal, Python, Fortran o PHP. En este curso se aprenderá el uso del lenguaje de programación C.

Dentro del ciclo de vida del software, la implementación de un algoritmo se encuentra dentro de la etapa de codificación del problema. Esta etapa va muy unida a la etapa de pruebas:



**Figura 1:** Ciclo de vida del software, resaltando las etapas de *códificación y pruebas*, las cuales se cubrirán en esta práctica.

## Escritura en pantalla

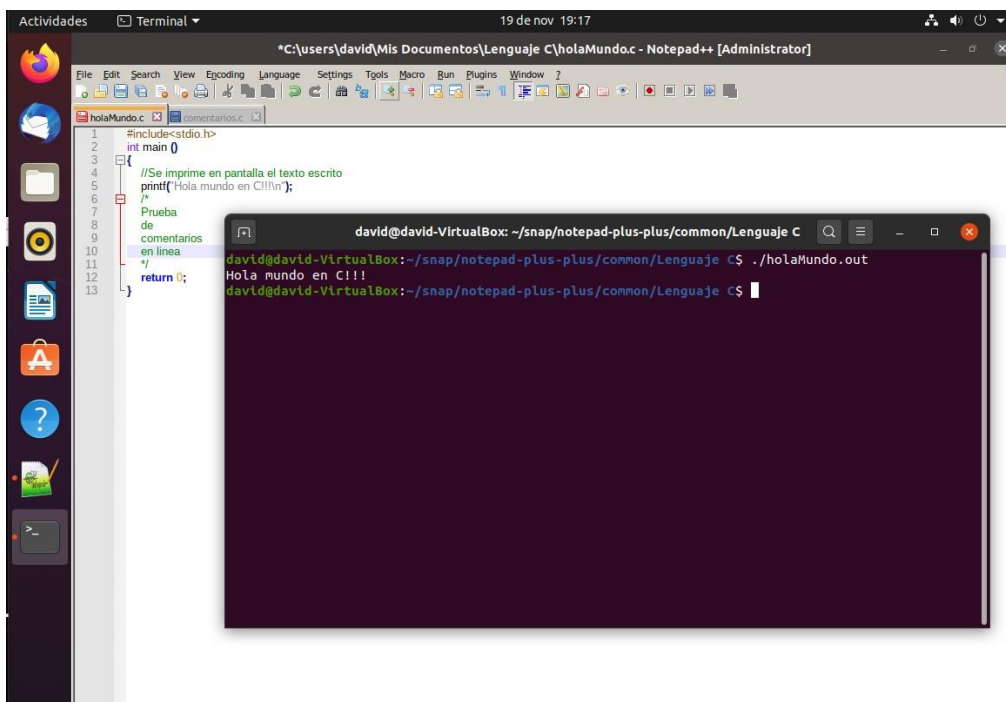
C presenta una gran variedad de maneras en las cuales se puede mostrar en formato de salida información, resultados o visualizaciones del código con ayuda de la pantalla, la función que realiza dicho proceso es *"printf"* siendo tan importante que todos los códigos lo conllevan, para expresar escritura en pantalla es necesario que nuestro programa tenga un objetivo inicial, anteriormente analizamos nuestro primer *"Hola Mundo"* quien contaba en si sintaxis con la función mencionada; es por eso que en esta ocasión revisaremos tipos de escrituras de datos, comentarios y secuencias de escape.

### Comentarios

Los comentarios en C se representan de dos formas; *"//"* todo lo que se escriba después de estos se expresan como comentarios solo por una línea, ahora para realizar comentarios en un bloque de líneas es necesario utilizar *"/\* (Texto expresado en varias líneas) \*/"* en donde todo lo que se escriba dentro del bloque quedará comentado.

Es de suma importancia realizar comentarios durante el desarrollo de un programa, ya que funcionan para distinguir lo que se efectúa en cada sección o en trabajos colaborativos para que otros usuarios comprendan el código mostrado.

Cabe recalcar que lo que se encuentra dentro de un comentario, no se mostrará en pantalla al momento de ejecutar el programa, es por eso que se debe tener precaución al comentar secciones.



The screenshot shows a Notepad++ window titled `*C:\users\david\Mis Documentos\Lenguaje C\holaMundo.c - Notepad++ [Administrator]`. The code in the editor is as follows:

```
1 #include<stdio.h>
2 int main ()
3 {
4     //Se imprime en pantalla el texto escrito
5     printf("Hola mundo en C!\n");
6     /*
7     Prueba
8     de
9     comentarios
10    en linea
11    */
12    return 0;
13 }
```

Overlaid on the Notepad++ window is a terminal window titled `david@david-VirtualBox: ~/snap/notepad-plus-plus/common/Lenguaje C`. The terminal shows the command `./holaMundo.out` being executed, which results in the output `Hola mundo en C!!!`.

## Tipos de datos

El lenguaje C presenta una gran variedad de formatos de datos para expresar valores que ingresan al programa, con esto, se presentan los siguientes:

Formato	Tipo de Dato
%d , %i , %ld, %li , %o , %x	Entero
%f , %lf , %e , %g	Flotante
%c , %d, %i, %o, %x	Carácter
%s	Cadena de Caracteres

Para enteros y caracteres se basa en lo siguiente

Tipo de dato	Número de bits	Valor mínimo	Valor máximo
signed char	8	-128	127
unsigned char	8	0	255
signed short	16	-32 768	32 767
unsigned short	16	0	65 535
signed int	32	-2 147 483 648	2 147 483 647
unsigned int	32	0	4 294 967 294
signed long	64	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
unsigned long	64	0	18 446 744 073 709 551 615
enum	16	-32 768	32 767

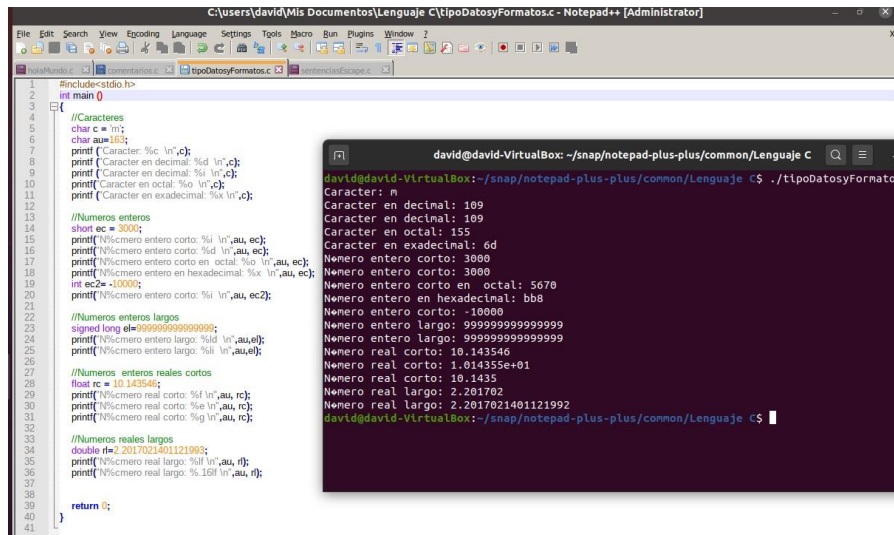
Ahora para números reales, aplican dos casos:

Tipo de dato	Número de bits	Valor mínimo	Valor máximo
float	32	3.4 E-38	3.4 E 38
double	64	1.7 E -308	1.7 E 308

La declaración de variables es fundamental en un código, ya que expresa valores que el desarrollador puede expresar obteniendo el valor que desea sin definir operación por operación, dicha declaración se define en nuestro “*printf*” en formato:

*TipoDeDato nombreDeVariable = dato;*

Ahora añadiendo algunos ejemplos obtenemos tanto caracteres, números enteros y números reales mostrando de lado izquierdo la sintaxis del código y del lado derecho la impresión en pantalla de los resultados, observando de nueva cuenta los comentarios incluidos.



The Notepad++ window displays a C program named `tipoDatosFormatos.c`. The program includes `<stdio.h>` and defines a `main` function. It uses `printf` to output various data types and values, including characters, integers, and floating-point numbers. The terminal window shows the output of the program, displaying the values of these variables in different formats (decimal, octal, hexadecimal, etc.).

```
#include<stdio.h>
int main ()
{
    //Caracteres
    char c = 'm';
    printf ("Caracter: %c \n",c);
    printf ("Caracter en decimal: %d \n",c);
    printf ("Caracter en octal: %o \n",c);
    printf ("Caracter en hexadecimal: %x \n",c);

    //Numeros enteros
    short ec = 3000;
    printf ("Númeroo entero corto: %i \n",au, ec);
    printf ("Númeroo entero corto: %d \n",au, ec);
    printf ("Númeroo entero corto en octal: %o \n",au, ec);
    printf ("Númeroo entero en hexadecimal: %x \n",au, ec);
    int ec2 = -10000;
    printf ("Númeroo entero corto: %i \n",au, ec2);

    //Numeros enteros largos
    signed long el = 9999999999999999;
    printf ("Númeroo entero largo: %ld \n",au, el);
    printf ("Númeroo entero largo: %i \n",au, el);

    //Numeros reales cortos
    float rc = 10.143546;
    printf ("Númeroo real corto: %f \n",au, rc);
    printf ("Númeroo real corto: %e \n",au, rc);
    printf ("Númeroo real corto: %g \n",au, rc);

    //Numeros reales largos
    double rl = 2.2017021401121992;
    printf ("Númeroo real largo: %lf \n",au, rl);
    printf ("Númeroo real largo: %l \n",au, rl);

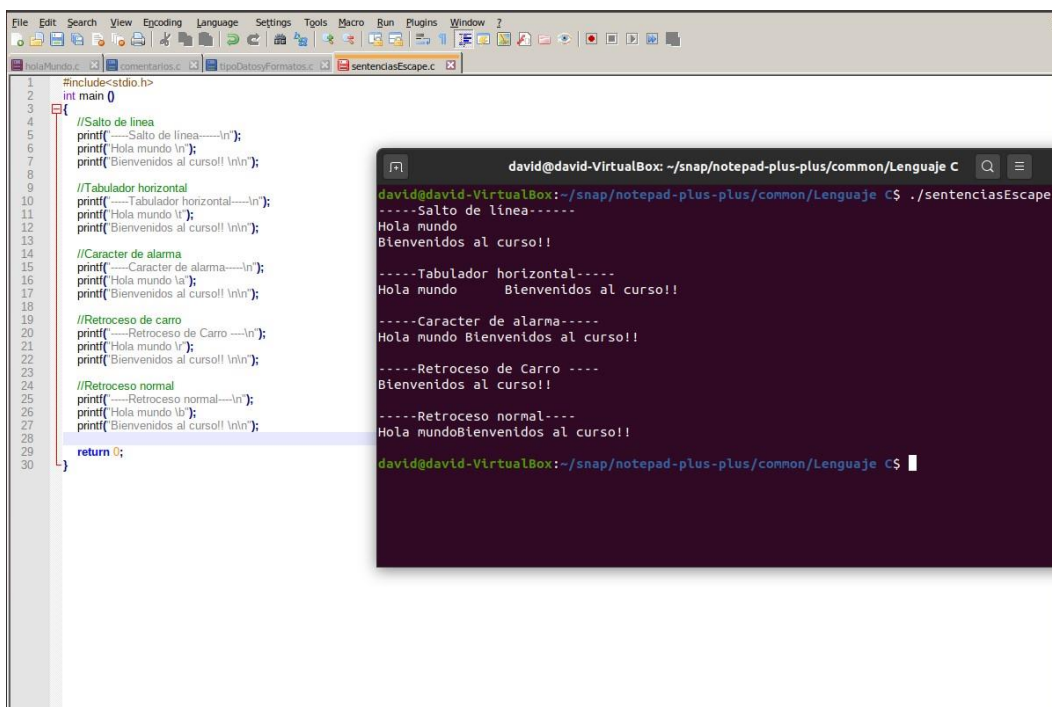
    return 0;
}
```

```
Caracter: m
Caracter en decimal: 109
Caracter en octal: 155
Caracter en hexadecimal: 6d
Númeroo entero corto: 3000
Númeroo entero corto: 3000
Númeroo entero corto en octal: 5670
Númeroo entero en hexadecimal: b98
Númeroo entero corto: -10000
Númeroo entero largo: 9999999999999999
Númeroo entero largo: 9999999999999999
Númeroo real corto: 10.143546
Númeroo real corto: 1.014355e+01
Númeroo real corto: 10.1435
Númeroo real largo: 2.201702
Númeroo real largo: 2.2017021401121992
```

## Secuencias de escape

De la misma forma, C expresa secuencias para la impresión de textos tales como:

- `\n` Al estar incluida, genera un salto de línea
- `\t` Genera una tabulación horizontal
- `\a` Expresa un carácter de alarma siendo un ligero sonido
- `\r` Retroceso de carro, también expresa un ligero sonido, pero suprime la línea anterior
- `\b` Retroceso normal, suprime una línea de comandos



The Notepad++ window displays a C program named `sentenciasEscape.c`. The program includes `<stdio.h>` and defines a `main` function. It uses `printf` to output various escape sequences, including `\n` (newline), `\t` (horizontal tab), `\a` (alarm), `\r` (carriage return), and `\b` (backspace). The terminal window shows the output of the program, demonstrating the effect of these escape sequences on the text display.

```
#include<stdio.h>
int main ()
{
    //Salto de línea
    printf ("----Salto de línea----\n");
    printf ("Hola mundo \n");
    printf ("Bienvenidos al curso!! \n\n");

    //Tabulador horizontal
    printf ("----Tabulador horizontal----\n");
    printf ("Hola mundo \t");
    printf ("Bienvenidos al curso!! \n\n");

    //Caracter de alarma
    printf ("----Caracter de alarma----\n");
    printf ("Hola mundo \a");
    printf ("Bienvenidos al curso!! \n\n");

    //Retroceso de carro
    printf ("----Retroceso de Carro ----\n");
    printf ("Hola mundo \r");
    printf ("Bienvenidos al curso!! \n\n");

    //Retroceso normal
    printf ("----Retroceso normal----\n");
    printf ("Hola mundo \b");
    printf ("Bienvenidos al curso!! \n\n");

    return 0;
}
```

```
----Salto de línea----
Hola mundo
Bienvenidos al curso!!

----Tabulador horizontal----
Hola mundo    Bienvenidos al curso!!

----Caracter de alarma----
Hola mundo
Bienvenidos al curso!!

----Retroceso de Carro ----
Bienvenidos al curso!!

----Retroceso normal----
Hola mundo
Bienvenidos al curso!!
```

## Conclusiones:

Ahora pude distinguir de manera acertada los elementos que forman parte de C, los tipos de Datos y algunas especificaciones para realizar la impresión en pantalla al llamar a los datos, así como variables que se declaren, la escritura es sencilla se manejar y aprendí que no es lo mismo pensar en cómo actuará el problema a plasmarlo con una codificación definida por reglas.

Me gustó la parte de secuencias de escape al ser más didáctica con respecto a los comentarios y tipos de datos, no queriendo decir que no sean importantes

## Bibliografía

- ♣ El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- ♣ Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura 1]. Consulta: junio de 2015. Disponible en:  
<https://www.mindmeister.com/es/273953719/aseguramiento-de-la-calidad-delsoftware-sqa>