



# **INSTITUTO TECNOLÓGICO BELTRÁN**

Centro de Tecnología e Innovación



Algoritmos y estructuras de datos – Lic. Yaps David P.

## FUNCIÓN PRINTF

Una funcionalidad básica para poder desarrollar nuestros primeros programas en C será la de mostrar mensajes por pantalla. Esto nos permitirá mostrar mensajes al usuario del programa, visualizar el resultado de cálculos, etc. Hay varias formas de mostrar mensajes por pantalla con C, vamos a comenzar estudiando el uso de una de ellas.

La salida de datos por pantalla más básica la conseguimos a través de la función `printf`. La sintaxis básica para esta función es la siguiente:

```
printf ("texto1 %codigoFormato1 texton %codigoFormaton", variable1, variablen);
```

## FUNCIÓN PRINTF - FORMATEADORES

Formateador	Salida
%d ó %i	entero en base 10 con signo (int)
%u	entero en base 10 sin signo (int)
%o	entero en base 8 sin signo (int)
%x	entero en base 16, letras en minúscula (int)
%X	entero en base 16, letras en mayúscula (int)
%f	Coma flotante decimal de precisión simple (float)
%lf	Coma flotante decimal de precisión doble (double)
%e	La notación científica (mantisa / exponente), minúsculas (decimal precisión simple ó doble)
%E	La notación científica (mantisa / exponente), mayúsculas (decimal precisión simple ó doble)
%c	caracter (char)
%s	cadena de caracteres (string)

## FUNCIÓN PRINTF

Se puede introducir la cantidad de texto y la cantidad de variables a mostrar por pantalla que se desee. A continuación, se representan algunos ejemplos más utilizados:

```
#include <stdio.h>

int main() {
    int numEntero=56; long numLong =44322155;
    float numFloat=2.36; double numDouble=2.545466;
    char miChar='c'; char cadena []= "Esto es una cadena";
    printf ("numEntero vale %d\n", numEntero);
    printf ("numLong vale %ld\n", numLong);
    printf ("numFloat vale %f\n", numFloat);
    printf ("numDouble vale %lf\n", numDouble);
    printf ("miChar vale %c\n", miChar);
    printf ("cadena vale %s\n", cadena);
    return 0;
}
```

## FUNCIÓN SCANF

En la función `scanf`, por cada argumento (variable) que se le pase, se debe escribir un especificador de formato, que establecerá el formato de entrada por teclado del argumento. La forma más sencilla de escribir un especificador de formato en la función `scanf` es con el carácter tanto por ciento (%) seguido de un carácter de conversión que indique el tipo de dato del argumento. De momento, no es necesario conocer todos los especificadores de formato que se pueden usar en la función `scanf`, pero sí, los más usados, que son los mismos que en la función `printf`:

```
scanf( "%d", &numero);
```

## FUNCIÓN SCANF

A continuación se presenta un ejemplo:

```
#include <stdio.h>

int main()
{
    int numero;

    printf( "\n  Introduzca un numero entero: " );
    scanf( "%d", &numero );

    return 0;
}
```

## FUNCIÓN STRCMP

Vamos a ver una función más relacionadas con `string`. En este caso, `strcmp`, que nos servirá para comparar dos cadenas y así saber si son iguales o son diferentes. La función `strcmp` recibe dos parámetros que serán `dos string`, y si son iguales nos devolverá un 0, y si no son iguales devolverá cualquier otro número. Veamos un ejemplo a continuación:

```
#include <stdio.h>

int main()
{
    char nombre[20] = "Jose";
    char apellido[20] = "Perez";

    if (strcmp(nombre, apellido) == 0) {
        printf("Las cadenas son iguales");
    }else{
        printf("Las cadenas son distintas");
    }
}
```



# **INSTITUTO TECNOLÓGICO BELTRÁN**

Centro de Tecnología e Innovación