



Departamento de Ingeniería Informática
y Ciencias de la Computación
Universidad de Concepción

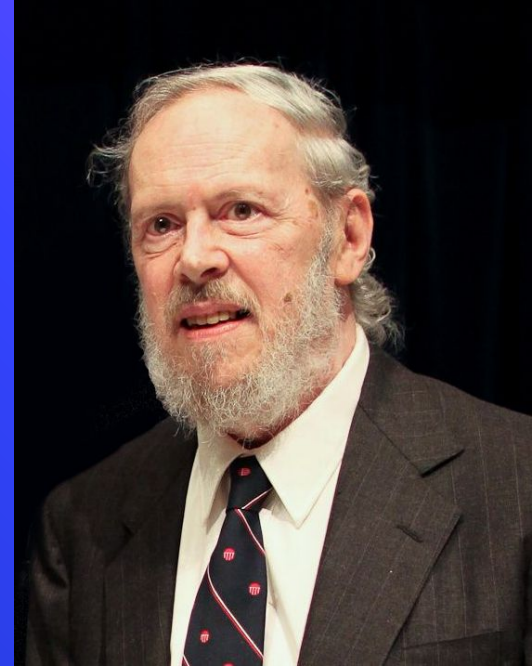
Laboratorio 1

Programación de Computadores (2025-2)



Lenguaje C

El lenguaje C es un lenguaje de programación de propósito general creado por Dennis Ritchie en 1972. Es utilizado para desarrollar software en sistemas embebidos, aplicaciones de escritorio, sistemas operativos y programación de bajo nivel.



Dennis Ritchie

“Hola mundo” en el lenguaje C

```
#include <stdio.h>

int main() {
    printf("Hola mundo\n");
    return 0;
}
```

“Hola mundo” en el lenguaje C

Directiva que indica la utilización de la biblioteca “stdio.h” usada en funciones de entrada y salida en la consola

stdio: standard input/output



```
#include <stdio.h>

int main() {
    printf("Hola mundo\n");
    return 0;
}
```

“Hola mundo” en el lenguaje C

Función principal “main”
del programa que
devuelve un dato del tipo
“int”




```
#include <stdio.h>

int main() {
    printf("Hola mundo\n");
    return 0;
}
```

“Hola mundo” en el lenguaje C

La función “printf”
imprime por pantalla el
mensaje “Hola mundo” y
agrega un salto de línea
con “\n”




```
#include <stdio.h>

int main() {
    printf("Hola mundo\n");
    return 0;
}
```

“Hola mundo” en el lenguaje C

Devuelve el valor “0” al sistema operativo, indicando que el programa se ejecutó correctamente



```
#include <stdio.h>

int main() {
    printf("Hola mundo\n");
    return 0;
}
```


¿Qué tipos de lenguaje hay?

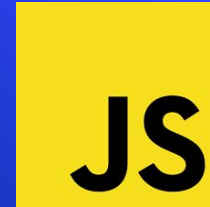
Compilado

El código se compila y se genera un archivo ejecutable



Interpretado

El código se va compilando línea a línea mientras se ejecuta



C es un lenguaje que requiere compilación

Los programas escritos en C deben traducirse a código de máquina (binario) para que sean ejecutados.

El compilador a utilizar en este curso será el GNU Compiler Collection (GCC)



C es un lenguaje que requiere compilación



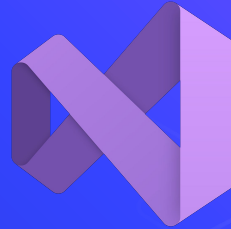
```
int main(){  
    printf("Hola mundo\n");  
    return 0  
}
```

00010101
10110101...

¿Dónde escribimos nuestro código fuente?

Existen programas especializados para escribir código, entre ellos los **entornos de desarrollo integrado (IDE)**.

Las IDEs integran el editor, compilador y otras herramientas útiles para el desarrollo.



Uso básico de GCC

En la terminal (consola de Linux):

```
user@equipo:~$ gcc holamundo.c
```

Si el programa es sintácticamente correcto, se genera un archivo ejecutable de nombre “a.out”, el cual se ejecuta en la terminal con “./a.out”:

```
user@equipo:~$ ./a.out  
Hola mundo
```

Se puede cambiar el nombre del ejecutable con el parámetro “-o [nombre]”

```
user@equipo:~$ gcc -o holamundo holamundo.c  
user@equipo:~$ ./holamundo  
Hola mundo
```

Comentarios en C

Hay dos formas de realizar comentarios en el código en C:

Comentarios para una línea:

```
int main()
{
    // Comentario
    printf("Hola mundo\n");

    return 0;
}
```

Comentario para múltiples líneas

```
int main()
{
    /*
        Comentario
        de multiples
        lineas
    */

    printf("Hola mundo\n");

    return 0;
}
```

Variables en C

Son espacios de memoria designados para almacenar valores de datos.

Cada variable tiene un nombre o identificador (único).

Las variables tienen *tipos de datos* que determinan el tipo de valor que pueden almacenar y el tamaño de memoria a reservar.

```
char letra = 'a';  
int numero = 1024;  
char frase[25] = "Programar es divertido";
```

DATO



Tipos de datos básicos en C

Tipo de Dato	Tamaño (bits/bytes)	Símbolo	Rango
char	8 bits / 1 byte	%c	-128 a 127
short	16 bits / 2 bytes	%d	-32.768 a 32.767
int	32 bits / 4 bytes	%d	-2.147.483.648 a 2.147.483.647
long	64 bits / 8 bytes	%ld	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
float	32 bits / 4 bytes	%f	1.2×10^{-38} a 3.4×10^{38}
double	64 bits / 8 bytes	%lf	2.3×10^{-308} a 1.7×10^{308}

*valores aproximados

Unsigned values

Válido para valores enteros (int, char, short, etc.)

En C, por defecto, los tipos numéricos son con signo (signed).
Eso significa que usan 1 bit para representar el signo (positivo o negativo).

char usa 8 bits → rango -128 a 127.

Si le agregas la palabra unsigned, esos mismos bits se usan solo para valores positivos.

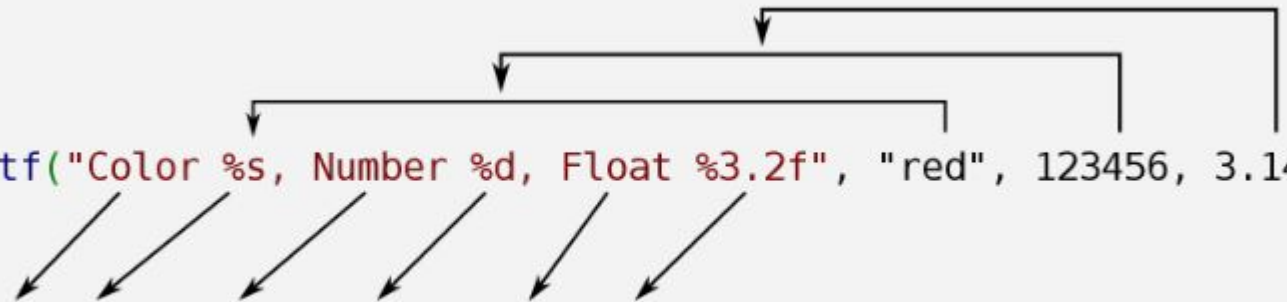
unsigned char usa 8 bits → rango 0 a 255.

char x = -1; → 11111111

unsigned char y = 255; → 11111111

Uso de printf con variables

Para imprimir los valores de las variables se debe especificar el formato usando el caracter %



```
Input: printf("Color %s, Number %d, Float %3.2f", "red", 123456, 3.14);
```

```
Output: Color red, Number 123456, Float 3.14
```

Ejercicio con printf

Escribir el código mostrado, asegurarse de que compile correctamente.

Modificar los valores de las variables y los tipos de identificador para observar el comportamiento del programa.

```
#include <stdio.h>

int main() {

    char c = 'k';
    short s = 16;
    int i = 32;
    long l = 64;
    float f = 32.32;
    double d = 64.64;
    char frase[15] = "Hola mundo";

    printf("c: %c, s: %d\n", c, s);
    printf("El valor de int i es: %d\n", i);
    printf("El valor de long l es: %ld\n", l);
    printf("f: %f, d: %lf\n", f, d);
    printf("%s\n", frase);

    return 0;
}
```

Función scanf

Esta función permite leer datos desde la consola.

Estos datos pueden ser almacenados en variables, indicando su dirección de memoria con el símbolo **&**

También se requiere el uso del especificador de formato (%).

```
#include <stdio.h>

int main(){

    // Ejemplo de uso de scanf
    int numero;
    printf("Ingrese un numero: ");
    scanf("%d", &numero);
    printf("El numero ingresado es: %d\n", numero);

    return 0;
}
```

Ejercicio con printf y scanf

Escriba un programa en C para calcular el área de un rectángulo.

El programa debe pedirle al usuario ingresar el ancho y el largo del rectángulo y debe entregar como resultado el área del rectángulo.

