

Unidad 4

Modularización



La **modularización** es dividir un programa grande en **partes más pequeñas y organizadas**, llamadas **módulos**. Cada módulo se escribe en **archivos diferentes**, y normalmente se separan en:

- **.h** → archivos **header** (declaraciones).
- **.c** → archivos de **implementación** (código de funciones).
- **main.c** → archivo principal que **usa los módulos**.



¿Por qué usar modularización?

- Hace el código más **ordenado** y **fácil de entender**.
- Permite **reutilizar funciones** en otros proyectos.
- Facilita el **trabajo en equipo**.
- Es más fácil de **probar** y **mantener**.



```
// funciones.h
// Declaración de funciones
#ifndef FUNCIONES_H
#define FUNCIONES_H

int esPar(int n);
int maximo(int a, int b);
float promedio(int a, int b);

#endif
```

```
// funciones.c
// Implementación de funciones
#include "funciones.h"

int esPar(int n) {
    return n % 2 == 0;
}

int maximo(int a, int b) {
    return (a > b) ? a : b;
}

float promedio(int a, int b) {
    return (a + b) / 2.0;
}
```

```
// main.c
#include <stdio.h>
#include "operaciones.h"

int main() {
    int x = 5, y = 3;

    int suma = sumar(x, y);
    int producto = multiplicar(x, y);

    printf("Suma: %d\n", suma);
    printf("Producto: %d\n", producto);

    return 0;
}
```

¿Cómo compilarlo?

Si estás usando la consola (por ejemplo, en Linux o usando `gcc`)

```
gcc main.c operaciones.c -o programa  
./programa
```

¿Por qué se usan archivos .h y .c con el mismo nombre?

En C, el archivo **.h** (header) y el archivo **.c** (implementación) suelen tener el mismo nombre porque trabajan juntos como un **módulo**. Pero **no es obligatorio** que tengan el mismo nombre; simplemente es una **buena práctica** para mantener tu código **organizado y fácil de entender**.



¿El orden afecta?

En la mayoría de los casos, NO afecta el orden:

Porque **gcc** primero **compila todos los .c por separado**, y **luego los enlaza** juntos en el ejecutable.

Es decir:

```
gcc main.c input.c operaciones.c -o  
programa
```

es igual que:

```
gcc operaciones.c input.c main.c -o  
programa
```

Siempre que todas las dependencias estén satisfechas, el programa va a compilar igual.

