

# Guía práctica de C: Estructuras, Enumeraciones, Funciones y Control de Flujo

---

## Protección de funciones con `!= 0`

En C es común que las funciones devuelvan `0` si todo salió bien, y otros valores (como `-1`) si hubo un error.

```
#include <stdio.h>
#include <string.h>

// Función con verificación de errores
int saludar(const char* nombre) {
    if (nombre == NULL || strlen(nombre) == 0) {
        return -1; // Error
    }
    printf("Hola, %s\n", nombre);
    return 0; // Éxito
}

int main() {
    const char* usuario = "Alan";

    if (saludar(usuario) != 0) {
        printf("Error al ejecutar la función saludar\n");
        return 1;
    }

    return 0;
}
```

## switch-case

Permite tomar decisiones múltiples según el valor de una variable.

```
#include <stdio.h>

int main() {
    int opcion = 2;

    switch (opcion) {
        case 1:
            printf("Seleccionaste la opción 1\n");
            break;
        case 2:
            printf("Seleccionaste la opción 2\n");
            break;
    }
}
```

```
        default:
            printf("Opción no válida\n");
            break;
    }

    return 0;
}
```

Se recomienda usar break para evitar ejecuciones no deseadas

## typedef enum

Se usa para definir un conjunto de constantes enteras con nombres más legibles.

```
#include <stdio.h>

typedef enum {
    ROJO,
    VERDE,
    AZUL
} Color;

void mostrarColor(Color c) {
    switch (c) {
        case ROJO:
            printf("Color ROJO\n");
            break;
        case VERDE:
            printf("Color VERDE\n");
            break;
        case AZUL:
            printf("Color AZUL\n");
            break;
        default:
            printf("Color no reconocido\n");
            break;
    }
}

int main() {
    Color miColor = VERDE;
    mostrarColor(miColor);
    return 0;
}
```

## typedef struct

Sirve para definir nuevos tipos de datos que agrupan múltiples variables.

```
#include <stdio.h>

typedef struct {
    char nombre[50];
    int edad;
    float altura;
} Persona;

int main() {
    Persona p1 = {"Alan", 27, 1.75};

    printf("Nombre: %s\n", p1.nombre);
    printf("Edad: %d\n", p1.edad);
    printf("Altura: %.2f m\n", p1.altura);

    return 0;
}
```