

Registro/Estructuras

⌵ Parcial

AutoEstudio

¿Qué es un registro en C (estructura)?

En C, un **registro** es definido con `struct`, y permite agrupar **diferentes tipos de datos** en una sola unidad lógica.



`strcpy` se usa en C para **copiar cadenas de texto** (strings) de un lugar a otro.

En C,

las cadenas son arreglos de caracteres, y no puedes asignarlas directamente con `=` como en otros lenguajes.



Sintaxis

```
strcpy(destino, origen);
```

- `destino` : el arreglo donde quieres guardar la cadena
- `origen` : la cadena que quieres copiar

✗ Esto NO funciona:

```
char nombre[30];
```

```
nombre = "Luis"; // ✗ Error: no se pueden asignar strings así en C
```



Usamos

`strcpy` :

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

```
char nombre[30];
```

```
strcpy(nombre, "Luis"); // ✓ Copia "Luis" dentro del arreglo nombre
```

Vamos a hacer un ejemplo de un **registro de estudiante** usando `struct` en C, que incluya los siguientes campos:

- `matricula` (entero)
- `nombre` (cadena)
- `apellido` (cadena)
- `edad` (entero)
- `esRegular` (valor booleano: `1` para regular, `0` para no regular)

✓ Código en C: Registro de estudiante

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

// Definimos el tipo de estructura para un estudiante
typedef struct {
    int matricula;
    char nombre[30];
    char apellido[30];
    int edad;
    int esRegular; // 1 = Sí, 0 = No
} Estudiante;

// Función para imprimir la información de un estudiante
void imprimirEstudiante(Estudiante e) {
    printf("Matricula: %d\n", e.matricula);
    printf("Nombre: %s %s\n", e.nombre, e.apellido);
    printf("Edad: %d\n", e.edad);
    printf("¿Es regular?: %s\n", e.esRegular ? "Sí" : "No");
}

int main() {
    // Creamos una variable tipo Estudiante
    Estudiante alumno1; //sin usar struct al principio

    // Llenamos los datos
    alumno1.matricula = 20250123;
    strcpy(alumno1.nombre, "Luis");
    strcpy(alumno1.apellido, "Martínez");
    alumno1.edad = 20;
    alumno1.esRegular = 1;

    // Mostramos los datos
```

```

    imprimirEstudiante(alumno1);

    return 0;
}

```

Explicación por partes

Parte del código	Qué hace
<code>struct Estudiante</code>	Define un nuevo tipo de dato que agrupa varios campos
<code>char nombre[30];</code>	Guarda una cadena (nombre), hasta 29 caracteres + <code>\0</code>
<code>strcpy()</code>	Copia una cadena a un arreglo de caracteres
<code>esRegular</code>	Se usa como booleano (0 o 1)
<code>imprimirEstudiante()</code>	Función que muestra el contenido del registro

¿Y si quiero varios estudiantes?

Puedes usar un **arreglo de estructuras**:

```

#define MAX 3

int main() {
    Estudiante alumnos[MAX];

    // Llenamos manualmente 3 estudiantes (puedes usar scanf también)
    alumnos[0] = (struct Estudiante){20250123, "Luis", "Martínez", 20, 1};
    alumnos[1] = (struct Estudiante){20250124, "Ana", "Gómez", 21, 1};
    alumnos[2] = (struct Estudiante){20250125, "Carlos", "Ruiz", 19, 0};

    // Imprimir todos
    int i = 0;
    while (i < MAX) {
        printf("\nEstudiante #%d\n", i + 1);
        imprimirEstudiante(alumnos[i]);
    }
}

```

```

        i++;
    }

    return 0;
}

```

✓ Para ingresar con teclado

```

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

// Definimos la estructura con typedef
typedef struct {
    int matricula;
    char nombre[30];
    char apellido[30];
    int edad;
    int esRegular; // 1 = Sí, 0 = No
} Estudiante;

// Función para pedir los datos de un estudiante
void capturarEstudiante(Estudiante *e) {
    printf("Ingrese la matrícula: ");
    scanf("%d", &e->matricula);
    getchar(); // limpiar el salto de línea

    printf("Ingrese el nombre: ");
    fgets(e->nombre, sizeof(e->nombre), stdin);
    e->nombre[strcspn(e->nombre, "\n")] = '\0'; // quitar el salto de línea

    printf("Ingrese el apellido: ");
    fgets(e->apellido, sizeof(e->apellido), stdin);
    e->apellido[strcspn(e->apellido, "\n")] = '\0';
}

```

```

printf("Ingrese la edad: ");
scanf("%d", &e→edad);

printf("¿Es regular? (1 = Sí, 0 = No): ");
scanf("%d", &e→esRegular);
}

// Función para imprimir un estudiante
void imprimirEstudiante(Estudiente e) {
    printf("Matrícula: %d\n", e.matricula);
    printf("Nombre: %s %s\n", e.nombre, e.apellido);
    printf("Edad: %d\n", e.edad);
    printf("¿Es regular?: %s\n", e.esRegular ? "Sí" : "No");
}

int main() {
    int n;
    printf("¿Cuántos estudiantes deseas registrar? ");
    scanf("%d", &n);

    Estudiante alumnos[n];
    int i = 0;

    printf("\n--- Captura de estudiantes ---\n");
    while (i < n) {
        printf("\nEstudiante #%d:\n", i + 1);
        capturarEstudiante(&alumnos[i]);
        i++;
    }

    printf("\n--- Lista de estudiantes registrados ---\n");
    i = 0;
    while (i < n) {
        printf("\nEstudiante #%d\n", i + 1);
        imprimirEstudiante(alumnos[i]);
        i++;
    }
}

```

```
}  
  
return 0;  
}
```