

## Matrices

También pueden llamarse arreglos de dos dimensiones.

Una matriz es un conjunto de vectores del mismo tipo de dato que se guardan de forma contigua en la memoria. Es decir, una matriz es un vector de vectores.

- ⊛ Solo permiten almacenar variables/vectores que tengan el mismo tipo de dato.
- ⊛ Tienen un tamaño fijo, y se le asigna cuando se declara la matriz
- ⊛ Las variables/vectores se guardan de manera contigua en la memoria.

```
int matriz[10][3];
```

### Ejemplo

```
int matriz[10][3];
int tope_fil = 0;
int tope_col = 0;
int total = 0;
for (int i = 0; i < 10; i++){
    for (int j = 0; j < 3; j++){
        matriz[i][j] = i*j;
        total++;
    }
    tope_fil++;
}
tope_col = total / tope_fil;
```

```
for (int i = 0; i < tope_fil; i++){
    for (int j = 0; j < tope_col; j++){
        printf("%i", matriz[i][j]);
    }
    printf("\n");
}
```

Structs Un struct en C es una estructura de datos que permite agrupar diferentes tipos de datos en un mismo lugar.

### Ejemplo

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

typedef struct persona {
    char inicial_nombre;
    char inicial_apellido;
    int edad;
    int cantidad_hijos;
    bool trabaja;
} persona_t;
```

OBS: el typedef sirve para darle un "alias" al struct persona, en este caso persona\_t

```
//DECLARAR//
//FORMA 1//
persona_t homero = {'H', 'S', 39, 3, true};

//FORMA 2//
persona_t homero;
homero.inicial_nombre = 'H';
homero.inicial_apellido = 'S';
homero.edad = 39;
homero.cantidad_hijos = 3;
homero.trabaja = true;
```

OBS: persona\_t va a funcionar como cualquier otro tipo de dato.



Para acceder a los campos de un struct se usa .

### Pasaje por referencia

```
void envejecer_persona(persona_t *persona) {
    persona->edad += 1; // Se usa "->" porque "h" es un puntero
}

void mostrar_homero(persona_t *persona) {
    printf("Homero tiene %d años, persona->edad);
}

envejecer_persona(&homero);
```

- ⊛ Uso del operador ->: cuando persona\_t \*persona es un puntero, se usa persona->edad en lugar de persona.edad. Esto es equivalente a (\*persona).edad.
- ⊛ Paso por referencia (&homero): se envía la dirección de memoria de homero (persona\_t), permitiendo modificarlo sin copiar la estructura.