

Tema 4: Tipos Avanzados de Datos

Oscar Perpiñán Lamigueiro - David Álvarez

1 Vectores

2 Matrices

3 Cadenas de caracteres

4 Estructuras

Estructuras en C

Permiten almacenar valores de diferentes tipos bajo un mismo identificador.

```
struct identificador
{
    tipo_1 comp_1;
    tipo_2 comp_2;
    ...
    tipo_n comp_n;
};
```

```
typedef struct
{
    tipo_1 comp_1;
    tipo_2 comp_2;
    ...
    tipo_n comp_n;
} identificador;
```

identificador Nombre de la estructura

tipo_n Tipo de datos del componente comp_n.

comp_n Nombre de la variable/vector componente n-ésimo de la estructura.

Ejemplo con struct

```
#include <stdio.h>

struct contacto
{
    char nombre[30];
    int telefono;
    int edad;
    float notas[5];
};

int main()
{
    struct contacto person1;

    return 0;
}
```

```
#include <stdio.h>

typedef struct
{
    char nombre[30];
    int telefono;
    int edad;
    float notas[5];
} contacto;

int main()
{
    contacto person1;

    return 0;
}
```

Inicialización de valores en estructuras

Si no se especifica el identificador de cada componente la asignación se realiza en orden

```
#include <stdio.h>

typedef struct
{
    char nombre[50];
    char apellidos[50];
    int matricula;
} ficha;

int main ()
{
    ficha alumno1 = {"David", "Alvarez Sanchez", 1234};

    return 0;
}
```

Asignación de valores en estructuras

```
typedef struct {  
    int day;  
    int month;  
    int year;  
} date;  
  
int main () {  
    date d1, d2, d3;  
    // Asignacion por componentes  
    d1.day = 31;  
    d1.month = 12;  
    d1.year = 1999;  
  
    // Asignacion por copia  
    d3 = d1;  
    return 0;  
}
```

Asignación de cadenas en estructuras

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

typedef struct {
    char nombre[50];
    char apellidos[50];
    int matricula;
} ficha;

int main ()
{
    ficha alumno1, alumno2, alumno3;
    // Para asignar cadenas usamos strcpy o un código equivalente
    strcpy(alumno1.nombre, "David");
    strcpy(alumno1.apellidos, "Alvarez Sanchez");
    alumno1.matricula = 1234;
    return 0;
}
```

Escritura de componentes de una estructura con scanf

```
#include <stdio.h>

typedef struct
{
    char nombre[50];
    char apellidos[50];
    int matricula;
    float notas[2];
} ficha;
```

// Continúa el código a la derecha ...

```
int main()
{
    ficha alumno;
    printf("Nombre:");
    scanf("%s", alumno.nombre);
    printf("Apellidos:");
    scanf("%49[^\n]", alumno.apellidos);
    printf("Numero de matricula:");
    scanf("%i", &alumno.matricula);
    printf("Nota matematicas:");
    scanf("%f", &alumno.notas[0]);
    printf("Nota fisica:");
    scanf("%f", &alumno.notas[1]);

    return 0;
}
```


Estructuras dentro de estructuras

Una estructura puede contener otras estructuras.

```
#include <stdio.h>

typedef struct
{
    int d, m, a;
} fecha;

typedef struct
{
    char nombre[50];
    char apellidos[50];
    int matricula;
    fecha fNacimiento;
} ficha;
```

```
// ... continuacion del código de la izquierda
int main()
{
    ficha alumno1, alumno2;

    alumno1.fNacimiento.d = 31;
    alumno1.fNacimiento.m = 12;
    alumno1.fNacimiento.a = 1999;

    alumno2.fNacimiento = (fecha){1, 1, 2000};

    return 0;
}
```

Vector de estructuras

Después de definir una estructura, se pueden generar vectores basados en esa estructura.

```
#include <stdio.h>

typedef struct {
    int day;
    int month;
    int year;
} date;

int main() {
    date fechas[3] = { // Vector de 3 fechas
        {1, 1, 1999},
        {31, 12, 2000},
        {15, 5, 1980}
    };
    return 0;
}
```

Vector de estructuras

La asignación de valores sigue las mismas reglas que para vectores de tipos simples (mediante corchetes []).

```
#include <stdio.h>

typedef struct
{
    int day;
    int month;
    int year;
} date;

int main() {
    date fechas[3]; // Vector de 3 fechas
    fechas[0].day = 1;
    fechas[1] = (date) {31, 12, 1999};
    fechas[2] = fechas[1];
    return 0;
}
```