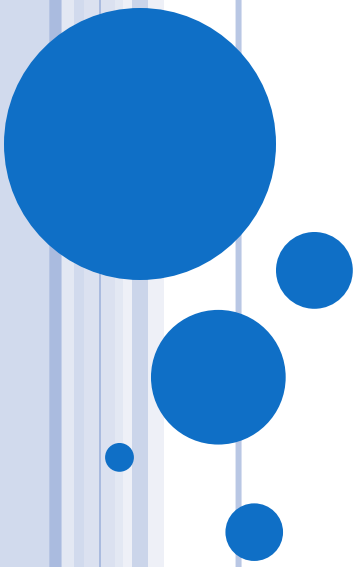
Estructura de Datos



Estructuras anidadas

Uniones

Anidación de estructuras (registros)

```
struct   info_dir
{
  char direccion[25];
  char ciudad[20];
  char providencia[20];
  int cod_postal;
}
```

```
struct   cliente
{
  char nombre_cliente[30];
  struct info_dir direccion_cliente;
  double saldo;
}
```

```
struct   empleado
{
  char nombre_emp[30];
  struct info_dir direccion_emp;
  double salario;
}
```

cliente:

nombre_cliente

Info_dir

- direccion
- ciudad
- povincia
- cod_postal

saldo

empleado:

nombre_emp

Info_dir

- direccion
- ciudad
- provincia
- cod_postal

salario

EJEMPLO DE ESTRUCTURA ANIDADA

personaEmpleado

persona

fecha

```
struct fecha
{
  int dia, mes, anio;
}
```

```
struct persona
{
  char nombre [20];
  int edad;
  float estatura;
  int peso;
  struct fecha fec;
}
```

```
struct   personaEmpleado
{
  struct persona unapersona;
  float salario;
  int horas_semana;
```

Union

- Las uniones son un tipo especial de estructuras que permiten almacenar elementos de diferentes tipos en las mismas posiciones de memoria, aunque evidentemente no simultáneamente. Así, las uniones son similares a las estructuras, pero con la diferencia de que en las uniones sus campos se almacenan solapándose unos con otros en la misma ubicación de memoria; al contrario que en las estructuras, donde los campos se almacenan unos seguidos de otros en posiciones contiguas de memoria.

- En esencia, las uniones sirven para ahorrar espacio en memoria. Para almacenar los miembros de una unión, se requiere una zona de memoria igual a la que ocupa el miembro de mayor tamaño en la unión.

- Todos los miembros son almacenados en el mismo espacio de memoria y comienzan en la misma dirección. El valor almacenado es sobre escrito cada vez que se asigna un valor al mismo miembro o a un miembro diferente, aquí radica la diferencia con las estructuras.

- Las uniones pueden albergar diferentes tipos de datos, pero sólo uno, de entre todos los posibles, al mismo tiempo. Lo cual indica que sólo uno de ellos puede estar activo en cada momento.

Union

Una unión es una variable la cual podría guardar (en momentos diferentes) objetos de diferentes tamaños y tipos. C emplea la sentencia unión para crear uniones por ejemplo:

```
union numero
{
short shortnumero;
long longnumero;
double floatnumero;
} unumero;
```

con lo anterior se define una unión llamada numero y una instancia de esta llamada **unumero**.

numero es la etiqueta de la unión y tiene el mismo comportamiento que la etiqueta en la estructura.