

CURSOS
INTERSEMESTRALES



PROTECO

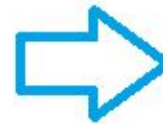
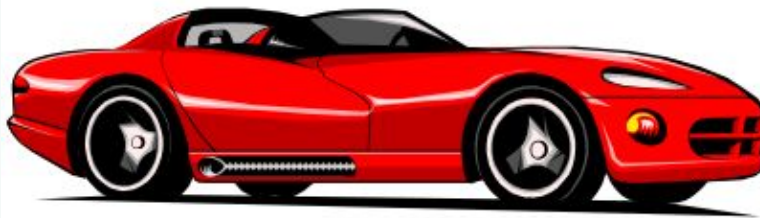
C Intermedio

Estructuras

¿Qué es una Estructura?

Es una colección de uno a más tipos de elementos denominados miembros, cada uno de los cuales puede ser un tipo de dato diferente.

Estructura: Carro



■ Miembros

- color -> **char**
- velocidad -> **float**
- numRuedas -> **int**
- motor -> **char**



¿Qué es una estructura?

Supongamos que se desea almacenar los datos de una colección de discos compactos (CD) de música. Estos datos pueden ser:

- Título
- Artista
- Núm. de Canciones
- Precio
- Fecha de Compra

```
struct coleccion_CD {  
    char titulo[30];  
    char artista[25];  
    int numCanciones;  
    float precio;  
    char fecha_compra[20];  
};
```



Declaración de una Estructura

```
struct <nombreDeLaEstructura> {  
    <tipoDeDato>    <nombreDeMiembro>  
    <tipoDeDato>    <nombreDeMiembro>  
    <tipoDeDato>    <nombreDeMiembro>  
    ...  
    <tipoDeDato>    <nombreDeMiembro>  
};
```

```
struct coleccion_CD {  
    char titulo[30];  
    char artista[25];  
    int numCanciones;  
    float precio;  
    char fecha_compra[20];  
};
```



Declarando variables struct:

Forma 1:

```
struct coleccion_CD {  
    char titulo[30];  
    char artista[25];  
    int numCanciones;  
    float precio;  
    char fecha_compra[20];  
};  
  
int main(){  
    struct coleccion_CD CD1,CD2,CD3;  
}
```

Forma 2:

```
struct coleccion_CD {  
    char titulo[30];  
    char artista[25];  
    int numCanciones;  
    float precio;  
    char fecha_compra[20];  
}CD1,CD2,CD3;
```



Typedef

Se usa para dar a un tipo de dato, un nuevo nombre (alias).

typedef unsigned char BYTE;

Después de esta definición de typedef, el identificador podrá ser usado como una abreviatura para el tipo **unsigned char**.

BYTE bi, b2;



Typedef

El uso de typedef en estructuras define un nuevo tipo de dato para después usar dicho tipo de dato para definir variables de estructuras directamente.

```
typedef struct paciente{  
    //Miembros  
}P;
```

```
P p; //Declaración de variable de tipo struct  
paciente
```



¿Qué es una Estructura Anidada?

Es una estructura dentro de otra.

```
struct info_direccion{
    char direccion[30];
    char ciudad[20];
    char provincia[20];
};

struct empleado{
    char nombre[20];
    struct info_direccion dir_empleado;
    double salario;
};
```



¿Qué es una Estructura Anidada?

empleado				
nombre	dir_empleado			salario
	dirección	ciudad	provincia	

```
struct info_direccion{  
    char direccion[30];  
    char ciudad[20];  
    char provincia[20];  
};  
  
struct empleado{  
    char nombre[20];  
    struct info_direccion dir_empleado;  
    double salario;  
};
```



Uniones

La definición de "unión" es similar a la de "estructura", La diferencia entre las dos es que en una estructura, los miembros ocupan diferentes áreas de la memoria, pero en una unión, los miembros ocupan la misma área de memoria.

```
union {  
    int i;  
    double d;  
} u;
```



Uniones

El programador puede acceder a través de "u.i" o de "u.d", pero no de ambos al mismo tiempo.

Cómo "u.i" y "u.d" ocupan la misma área de memoria, modificar uno modifica el valor del otro, algunas veces de maneras impredecibles.

El tamaño de una unión es el de su miembro de mayor tamaño.



Enumeraciones

Una enumeracion (enum) es un tipo definido con constante de tipo entero.

En la declaracion de un tipo enum creamos una lista de tipo de datos que se asocian con + las constantes enteras 0, 1, 2, 3, 4, 5...su forma de definirlas es la siguiente:

```
enum  
{  
    enumerador1, enumerador2, ...  
    enumeradorn  
};
```



Enumeraciones

Al declarar la enum se puede asociar a los tipos de datos a valores constantes en vez de la asociacion que por defecto se realiza (0, 1, 2, ...), se utiliza entonces este formato:

```
enum Nombre
{
    enumerador1 = valor_constante1,
    enumerador2 = valor_constante2,
    ...
    enumeradorn = valor_constanten,
};
```



Enumeraciones

Un ejemplo de una enum:

```
enum Boolean  
{  
    FALSE,  
    TRUE  
};
```

Se definen dos constantes para las constantes true y false con valores iguales a 0 para False y 1 para True.

