

Estructuras en C/C++

Mag. Ing. Nancy López

Estructuras

- Las estructuras nos permiten agrupar datos de distinto tipo bajo un mismo nombre.
- Son llamadas también registros y están muy relacionadas con los registros de las bases de datos, incluso sus datos se denominan campos.

Estructuras

- **Definición**

```
struct nombreEstructura{  
    tipoDato1 nombre;  
    tipoDato2 nombre;  
}variableEstructura[tamaño];
```

Estructuras

- Ejemplo

```
struct persona{  
    char nombre[20], apellido[20];  
    char domicilio[20], cedula[9];  
    int nacim[3];  
};
```

Estructuras

- Para poder usar la estructura defino una variable de ese tipo.

```
struct persona{  
    char nombre[20], apellido[20];  
    char domicilio[20], cedula[9];  
    int nacim[3];  
} per[100] ;
```

Estructuras

- Otra forma de definir la variable estructura:

```
struct persona{  
    char nombre[20], apellido[20];  
    char domicilio[20], cedula[9];  
    int nacim[3];  
};
```

- Dentro del main:

```
persona per[100];
```

El nombre de la estructura se usa sólo para definir la estructura, luego se usa la variable de tipo estructura.

Estructuras

- La variable de tipo estructura es la que se usará para acceder a los campos de la estructura:

```
gets(per[0].nombre);  
gets(per[0].domicilio);  
cin>>per[0].nacim[0];  
cin>>per[0].nacim[1];  
cin>>per[0].nacim[2];
```

Estructuras anidadas

- Se puede definir una estructura y utilizarla dentro de otra. Por ejemplo:

```
struct fecha{  
    int dia, mes, anio;  
};  
struct persona{ ...  
    fecha nacim;  
}per[100];
```


Estructuras anidadas

- Para utilizar los valores:

```
cin>>per[0].nacim.dia;
```

```
cin>>per[0].nacim.mes;
```

```
cin>>per[0].nacim.anio;
```

Ejemplo estructuras

2. Realizar la declaración de tipo registro para un automóvil: los campos son marca, modelo, año, color, N° de puertas y precio, hacer un programa que cargue y visualice los datos ingresados.

Ejemplo estructura

```
struct automovil{ char marca[20], modelo[20],  
    matricula[8], color[10];  
    int anio;  
}car[100]; //fin estructura  
  
void main()  
{  
    int i, cant=0, opcion, enc;  
    char seguir, matBus[9], nuevo[20];
```

```
do{  
    cout<<"Ingrese marca";  
    gets(car[cant].marca);  
    cout<<"Ingrese modelo";  
    gets(car[cant].modelo);  
    cout<<"Ingrese color";  
    gets(car[cant].color);  
    cout<<"Ingrese matricula";  
    gets(car[cant].matricula);  
    cant++;  
    cout<<"Desea seguir ingresando automoviles?";  
    cin>>seguir;  
}while(seguir=='s' && cant<100);
```

```
cout<<"Ingrese matricula de Automovil a modificar";  
  gets(matBus);  
  enc=-1;  
  for(i=0;i<cant;i++)  
    if(stricmp(matBus,car[i].matricula)==0)  
    {
```

```
      cout<<car[i].marca<<"\t"<<car[i].modelo<<"\t"<<car[i].color<<"\t"<<car[i].matricula<<"\n";  
      enc=i;  
    }  
  if(enc==-1)  
    cout<<"Automovil no encontrado";  
  else  
  {  
    //el else está en la otro hoja  
  }
```

//marca

```
cout<<"Ingrese nueva marca (ENTER para dejar igual)";  
gets(nuevo);  
if (strlen(nuevo)>0)  
    strcpy(car[enc].marca, nuevo);
```

//modelo

```
cout<<"Ingrese nuevo modelo (ENTER para dejar igual)";  
gets(nuevo);  
if (strlen(nuevo)>0)  
    strcpy(car[enc].modelo, nuevo);
```

//color

```
cout<<"Ingrese nuevo color (ENTER para dejar igual)";  
gets(nuevo);  
if (strlen(nuevo)>0)  
    strcpy(car[enc].color, nuevo);
```

//matricula

```
cout<<"Ingrese nueva matricula (ENTER para dejar igual)";  
gets(nuevo);  
if (strlen(nuevo)>0)  
    strcpy(car[enc].matricula, nuevo);  
clrscr();  
cout<<"Los nuevos datos son:"<<endl;  
cout<<car[enc].marca<<"\t"<<car[enc].modelo<<"\t"<<car[enc].color<<"\t"<<car[enc].matricula<<"\n";  
getch();
```

```
cout<<"Ingrese matricula de Automovil a buscar";  
    gets(matBus);  
    enc=0;  
    for(i=0;i<cant;i++)  
        if(stricmp(matBus,car[i].color)==0)  
        {  
  
            cout<<car[i].marca<<"\t"<<car[i].modelo<<"\t"<<car[i].  
            color<<"\t"<<car[i].matricula<<"\n";  
            enc++;  
        }  
    if(enc==0)  
        cout<<"Automovil no encontrado";
```

Bubble Sort para estructuras

La burbuja (Bubble Sort) para estructuras sigue el mismo formato pero, dado que guardamos todos los datos bajo un mismo nombre (la variable de tipo estructura), el intercambio es más simple.

Ejemplo

- En el caso del automóvil, si definimos la variable de tipo estructura como global, podemos usar una función para ordenar, de lo contrario, lo ponemos dentro del código:

```
struct automovil{ char marca[20], modelo[20];  
                  char matricula[8], color[10];  
                  int anio;  
} car[100]; //fin estructura
```

Ejemplo (cont)

- En el caso de las estructuras, lo que se intercambia es la estructura entera, por lo tanto, la variable auxiliar debe ser del mismo tipo que la estructura.
- Suponemos que, en la opción 3-Mostrar automóviles, se deberá mostrar ordenado por marca.

Ejemplo (cont)

```
void burbuja(int cant)
{
    automovil aux; //definimos una variable de tipo
                    //estructura automóvil

    int i, j;
    for(i=0; i<cant; i++)
        for(j=0; j<cant-1; j++)
            if(stricmp(car[j].marca, car[j+1].marca)>0)
            {
                aux=car[j]; //intercambiamos la estructura
                car[j]=car[j+1];
                car[j+1]=aux;
            }
}
```