

Universidad de Morón
Escuela Superior de Ingeniería, Informática y Cs. Agroalimentarias

Asignatura:
(3362) Programación II

CLASE 8

**Estructura de Datos:
REGISTROS**



Form2

Agregar Registro

Nombre: Apellido:

Edad: Profesion:

Agregar

Prof. Lic. Sonia Zugna de Jausoro

Registros

- Son **estructuras de Datos** cuyos componentes **pueden** ser de *diferente tipo*.
- Para **declarar un tipo registro** se debe incluir:
el *nombre* y el *tipo* de cada componente del mismo.
- A cada componente, de un registro, se lo llama *Campo*.
- En C, se usa la palabra reservada *struct*, para indicar que el tipo de Dato, que se está definiendo es un **registro**.

Definir registro y cargar una variable

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>

using namespace std;

int main(void)
{
    // Defino la estructura del registro
    struct Persona
    {
        string nombre;
        int edad;
        float altura;
    };

    // Declaro la variable unaPersona de tipo Persona
    Persona unaPersona;

    // Cargo los datos del registro

    unaPersona.nombre = "Analia";

    unaPersona.edad = 19;

    unaPersona.altura = 1.68;

    // Imprimo los datos del registro
    cout << "El nombre es: " << unaPersona.nombre << endl;
    cout << "La edad es: " << unaPersona.edad << endl;
    cout << "La altura es: " << unaPersona.altura << endl;

    getch();
    return 0;
}
```

Copia un struct en otro

```
...
struct Ejemplo
{
    int campo1;
    float campo2;
};

int main()
{
    Ejemplo struct1, struct2, struct3;

    struct1.campo1=50;
    struct1.campo2=1.23;

    // copia campo a campo
    struct2.campo1= struct1.campo1;
    struct2.campo2= struct1.campo2;
    cout<<struct2.campo1<<endl<<struct2.campo2<<endl;

    // copia el struct completo
    struct3 = struct2;
    cout<<struct3.campo1<<endl<<struct3.campo2<<endl;

    getch();
    return(0);
}
```

Struct anidados

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
using namespace std;

struct Fecha {
    int dia;
    int mes;
    int anio;
};

struct DatosAlumno {
    int matricula;
    Fecha fechaNacim;
    float promedio;
};

int main() {

    DatosAlumno inscripto;

    inscripto.matricula = 1500;

    inscripto.fechaNacim.dia = 19;

    inscripto.fechaNacim.mes = 8;

    inscripto.fechaNacim.anio = 1996;

    inscripto.promedio = 7.5;

    cout<<"La matricula es: "<<inscripto.matricula<<endl;

    cout<<"Su fecha de nacimiento es: "<<inscripto.fechaNacim.dia<<"/"<<inscripto.fechaNacim.mes<<
        /"<<inscripto.fechaNacim.anio<<endl;

    cout << "El promedio del alumno es: " << inscripto.promedio ;

    getch();
    return 0;
}
```

Nota:

Los **struct** se pueden definir,
dentro del main o antes del
main.

```
// Programa que arma una Estructura Vector de Registros (STRUCT)
// conteniendo los datos de los alumnos de las carreras de Informática
// ( Ingeniería, código 20 - Licenciatura, código 30 - Tecnicatura, código 40 )
// Informa cantidad de alumnos inscriptos en cada carrera
// Informa Porcentaje de alumnos en Ingeniería y en Licenciatura
// Informa los nombres y los promedios de los alumnos menores de una determinada edad
```

```
#include <conio.h>
#include <iostream.h>
```

```
#define INGENIERIA 20
#define LICENCIATURA 30
#define TECNICATURA 40
```

```
// Definición del Registro estructura "Alumno"
```

```
struct Alumno {
    long matricula;
    char nombre[10];
    int edad;
    int codigoCarrera;
    float promedio;
};
```

```
int main() {
    int cantidadAlumnos;
    cout << "Qué cantidad de alumnos va a ingresar? ";
    cin >> cantidadAlumnos;
```

// Definición de la estructura Vector conformado por elementos del tipo “Alumno”

Alumno vAlumnos[cantidadAlumnos];

vAlumnos:	.nombre	.edad	.codCarrera	.promedio	.matricula	.nombre	.edad	.codCarrera	.promedio	.matricula	.nombre	.edad	.codCarrera	.promedio	.matricula	.nombre	.edad	.codCarrera	.promedio	.matricula
	vAlumnos[0]					vAlumnos[1]					vAlumnos[2]					vAlumnos[3]				

// Ingreso de datos para armar la Estructura

```
for (int i = 0; i < cantidadAlumnos; i++) {  
    cout << "---> Ingreso de datos para el Alumno " << i+1 << endl;  
    cout << "Matricula: ";  
    cin >> vAlumnos[i].matricula;  
    cout << "Nombre: ";  
    cin >> vAlumnos[i].nombre;  
    cout << "Edad: ";  
    cin >> vAlumnos[i].edad;  
    // Validación del ingreso del código de carrera  
    do {  
        cout << "Codigo de carrera (20 (Ing.), 30 (Lic.) o 40 (Tec.) ): ";  
        cin >> vAlumnos[i].codigoCarrera;  
    } while ( vAlumnos[i].codigoCarrera != INGENIERIA &&  
vAlumnos[i].codigoCarrera != LICENCIATURA && vAlumnos[i].codigoCarrera != TECNICATURA);  
  
    // Validación del ingreso del promedio  
    do {  
        cout << "Promedio: ";  
        cin >> vAlumnos[i].promedio;  
    }while (vAlumnos[i].promedio < 0.0 || vAlumnos[i].promedio > 10.0);  
} // del For
```


// Cálculo de cantidad de alumnos por carrera

```
int cantIng, cantLic, cantTec;  
cantIng = cantLic = cantTec = 0;
```

```
for (int a = 0; a < cantidadAlumnos; a++)  
{  
    if ( vAlumnos[a].codigoCarrera == INGENIERIA) {  
        cantIng++; }  
    else if ( vAlumnos[a].codigoCarrera == LICENCIATURA){  
        cantLic++; }  
    else{  
        cantTec++;  
    }  
}
```

// Informar cantidad de alumnos por carrera

```
cout << endl;  
cout << "Cantidad alumnos inscriptos en INGENIERIA: " << cantIng << endl;  
cout << "Cantidad alumnos inscriptos en LICENCIATURA: " << cantLic << endl;  
cout << "Cantidad alumnos inscriptos en TECNICATURA: " << cantTec << endl;
```

```
// Informar Porcentaje de alumnos en Ingenieria y en Licenciatura
```

```
cout << endl;
```

```
cout << "Porcentaje de alumnos inscriptos en INGENIERIA: " << ((float)cantIng  
                                     /((float)cantidadAlumnos) * 100 << " %" << endl;
```

```
cout << "Porcentaje de alumnos inscriptos en LICENCIATURA: " <<  
                                     ((float)cantLic/((float)cantidadAlumnos)*100 << " %" << endl;
```

```
// Informar los nombres y los promedios de los alumnos menores de una determinada edad
```

```
int edadTope;
```

```
cout << endl << "Ingrese una EDAD para informar datos de los alumnos que no la superan: ";
```

```
cin >> edadTope;
```

```
// recorrer la estructura buscando las edades menores a la ingresada
```

```
for(int e = 0; e < cantidadAlumnos; e++) {
```

```
    if (vAlumnos[e].edad <= edadTope) {
```

```
        cout << "Nombre: " << vAlumnos[e].nombre << " Promedio: " <<  
                vAlumnos[e].promedio << " Edad: " << vAlumnos[e].edad <<
```

```
endl;
```

```
    }
```

```
}
```

```
getch();
```

```
return 0;
```

```
}
```

Ejercicio: Matriz de elementos struct

```
1 // Ejercicio Profesionales por Consultorio por Turno - Matriz y Struct.
2 // Consultorios y Turnos: numerados de 1 a 3
3 // Las filas de la matriz son Consultorios y Las columnas son Turnos.
4 // Calcular e imprimir, por cada Turno, La mayor CANTIDAD de horas trabajadas en el mes.
5
6 #include <iostream>
7 #include <conio.h>
8 #include <stdio.h>
9 #include <iomanip>
10 using namespace std;
11
12 struct Fecha {
13     int dia;
14     int mes;
15     int anio;
16 };
17
18 struct DatosProfesional {
19     int matricula;
20     Fecha fechaIncor; // fecha de incorporación del profesional a La institución
21     float cantHs; // cantidad de horas trabajadas en el mes
22 };
23
24 int main ()
25 {
26     int f, c;
27     int cantCons= 3;
28     int cantTurnos=3;
29     float mayorCantHs;
30
31     // define La matriz de elementos formato struct
32     DatosProfesional mat[cantCons][cantTurnos];
33
34     // Ingreso de datos de Los Profesionales
35     for(f=0;f<cantCons;f++){
36         for(c=0;c<cantTurnos;c++){
37             cout<<"Ingrese la matricula del Profesional del Consultorio: "<< f+1 <<" del turno: " << c+1 <<endl;
38             cin>> mat[f][c].matricula;
39             cout<<"Ingrese el dia de incorporacion del profesional "<<endl;
40             cin>> mat[f][c].fechaIncor.dia;
41             cout<<"Ingrese el mes de incorporacion del profesional "<<endl;
42             cin>> mat[f][c].fechaIncor.mes;
43             cout<<"Ingrese el anio de incorporacion del profesional "<<endl;
44             cin>> mat[f][c].fechaIncor.anio;
45             cout<<"Ingrese cantidad de horas trabajadas por el profesional "<<endl;
46             cin>> mat[f][c].cantHs;
47         }
48     }
49     // recorre La matriz POR COLUMNA (son Los Turnos) buscando La MAYOR de horas trabajadas
50     for(c=0;c<cantTurnos;c++){
51         mayorCantHs=mat[0][c].cantHs; // inicializo el campo de comparación con el PRIMER elemento de La Columna
52
53         for(f=1;f<cantCons;f++){ // compara con Los elementos RESTANTES de La columna (desde fila 1)
54             if( mat[f][c].cantHs > mayorCantHs)
55                 mayorCantHs = mat[f][c].cantHs;
56         }
57         cout<<"La mayor cantidad de horas trabajadas por Profesional/es del turno "<< c+1 <<" es: "<< mayorCantHs <<endl;
58     }
59     getch();
60     return(0);
61 }
62
63
64
```

Matriz de elementos de tipo de dato *struct*

Datos de los PROFESIONALES. Matriz de: **CONSULTORIOS X TURNOS**

mat															
0				1				2							
0	123	12	12	2001	7	240	3	6	2001	7	185	8	5	2001	6
		día	mes	anio			día	mes	anio			día	mes	anio	
	matricula	fechaIncor				cantHs	matricula	fechaIncor			cantHs	matricula	fechaIncor		
1	190	9	1	2002	6	125	15	5	2003	4	145	10	9	2003	7
		día	mes	anio			día	mes	anio			día	mes	anio	
	matricula	fechaIncor				cantHs	matricula	fechaIncor			cantHs	matricula	fechaIncor		
2	200	7	4	2004	5	222	12	12	2001	9	178	8	5	2001	5
		día	mes	anio			día	mes	anio			día	mes	anio	
	matricula	fechaIncor				cantHs	matricula	fechaIncor			cantHs	matricula	fechaIncor		

Matriz de: 3 Consultorios x 3 Turnos
