



## Tipos de Datos definidos por el programador

Hasta ahora hemos visto los tipos de datos simples definidos en c sharp: char, int, float, void. Como hemos podido observar estos tipos de datos contienen un conjunto de valores con las mismas características.

Existen sin embargo otros tipos de datos que nos permiten almacenar un conjunto de datos de distintos tipos, por ejemplo si deseáramos almacenar para un grupo de alumnos su nombre, apellidos, edad, dirección, promedio, peso, estatura.

A este nuevo tipo de datos le llamamos enumeraciones y estructuras.

### Enumeraciones

Una enumeración es un tipo especial que permite agrupar un conjunto de valores en una lista asignándoles un identificador numérico de tipo entero (puede ser únicamente de byte, short, int o long).

Declaración de una Enumeración

Para declarar una enumeración debe usar la palabra clave enum seguida del nombre de la enumeración:

```
enum nombre_enumeracion {  
    Dato1,  
    Dato2,  
    Dato3,  
    Dato4,  
    DatoN  
}
```

Ejemplo:

```
enum planetas {  
    Mercurio,  
    Venus,  
    Tierra,  
    Marte,  
    Júpiter,  
    Saturno,  
    Urano,  
    Neptuno  
}  
public enum colores { rojo, blanco, negro, azul, rosa, verde, amarillo}
```

Por default el sistema le asigna numeración consecutiva desde 0 hasta N-1. Para el primer ejemplo Mercurio = 0, Venus = 1 y Neptuno = 7. Sin embargo se le puede asignar un valor distinto de la siguiente forma:



#### TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

```
enum planetas {  
    Mercurio=2,  
    Venus=4,  
    Tierra=6,  
    Marte=8,  
    Júpiter=10,  
    Saturno=12,  
    Urano=14,  
    Neptuno=16  
}
```

Para hacer uso de una enumeración se puede hacer a partir de su valor o de su identificador numérico:

**nombre\_enumeracion nombre\_cadena = nombre\_enumeracion.DatoN //Con su valor**

**int nombre\_variable = (int) nombre\_enumeracion.DatoN //Con su identificador numérico**

Ejemplo:

```
planetas miPlaneta = planetas.Tierra;
```

```
int miPlanetaEntero = (int) planetas.Tierra;
```

Si quiero obtener todos los elementos de mi enumeración lo puedo hacer utilizando

```
Array A = nombre_enumeracion.GetValues(typeof(nombre_enumeracion));
```

### Concepto de estructura

- Una **estructura** es una colección de uno o más elementos, cada uno de los cuales puede ser de un tipo de dato diferente.
- Cada elemento de la estructura se denomina **miembro**.
- Una estructura puede contener un número ilimitado de miembros.
- A las estructuras también se las llama **registros**.

Una estructura es un tipo de dato creado por el usuario, por tanto, es necesario *definirla* antes de poder utilizarla. Una vez definido, podremos crear variables de tipo estructura.

#### Declaración de una estructura

Una estructura se declara de la siguiente manera;

```
struct nombre_estructura  
{  
    public tipo_dato1 dato1;  
    public tipo_dato2 dato2;  
    public tipo_dato3 dato3;  
    public tipo_datoN datoN;  
}
```



#### TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

Ejemplos:

```
struct alumno
{
    public string nombre;
    public int edad;
    public float peso;
}

struct libro
{
    public string editorial;
    public string autor;
    public string titulo;
    public int paginas;
}
```

#### Uso de una estructura

Una vez que se ha declarado la estructura la puede utilizar como si fuera cualquier otro tipo de dato (int, string, float por ejemplo).

**nombre\_estructura nombre\_variable;**

Ejemplo:

```
alumno Alumno1;
alumno miAlumno, otroAlumno;
libro miLibroEscolar;
libro miLibroFavorito;
```

Después de crear una variable del tipo de estructura se puede acceder a sus elementos poniendo después del nombre de la variable un punto y posteriormente el campo.

```
nombre_variable.dato1 = valor;
```

Ejemplo:

```
Alumno1.nombre="Juan Perez";
Alumno1.edad = 15;
Alumno1.peso = 50.200F
Console.WriteLine("Nombre: {0}, Edad: {1}, Peso: {2}", Alumno1.nombre, Alumno1.edad,
Alumno1.peso);
miAlumno.nombre = "Rosa Gómez";
otroAlumno.nombre = "Juan Carlos Estrada";
miLibroEscolar.titulo="Curso de Programación C#";
miLibroEscolar.editorial ="Alfaomega";
miLibroEscolar.autor = "Francisco Ceballos";
```



#### TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

#### Ejercicios de la Práctica

Mediante estos ejercicios el alumno aprenderá a utilizar enumeraciones y estructuras para representar grupos de elementos que pueden ser de diferentes tipos de datos.

1. Crea una nueva solución llamada **Practica14\_enum\_y\_struct** y llama a tu proyecto **estructura\_ejemplo**. Dentro del proyecto escribe el código de la parte inferior.

1.1. A continuación ejecútalo y anota tus observaciones.

1.2. Comenta las partes que creas necesarias // o /\* \*/

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace estructura_ejemplo
{
    class Program
    {
        struct alumno
        {
            public string nombre;
            public int edad;
            public float[] evaluaciones;
        }
        static void Main(string[] args)
        {
            Program objeto_estructura = new Program();
            Console.WriteLine("*****DATOS PARA UNA ESTRUCTURA ALUMNO*****");
            objeto_estructura.estructura_alumnos();
            Console.WriteLine("*****DATOS PARA UN ARREGLO DE ESTRUCTURA ALUMNO*****");
            objeto_estructura.arreglo_estructura_alumnos();
        }
        public void estructura_alumnos()
        {
            alumno miAlumno;
            float prom;
            miAlumno.nombre = "Erika Dominguez";
            miAlumno.edad = 25;
            miAlumno.evaluaciones = new float[3];
            miAlumno.evaluaciones[0] = 8.5F;
            miAlumno.evaluaciones[1] = 9.3F;
            miAlumno.evaluaciones[2] = 5.7F;
            prom = this.calcula_promedio(miAlumno.evaluaciones);
            Console.WriteLine("Nombre del alumno:{0}, edad: {1}, promedio: {2}", miAlumno.nombre, miAlumno.edad, prom);
        }
        public void arreglo_estructura_alumnos()
        {
            alumno[] miArregloAlumnos;
            float prom;
            int cont = 0;
            miArregloAlumnos = new alumno[3];
            miArregloAlumnos[0].nombre = "Juan Perez";
            miArregloAlumnos[0].edad = 23;
            miArregloAlumnos[0].evaluaciones = new float[3];
            miArregloAlumnos[0].evaluaciones[0] = 10.0F;
            miArregloAlumnos[0].evaluaciones[1] = 7.5F;
            miArregloAlumnos[0].evaluaciones[2] = 8.3F;
            miArregloAlumnos[1].nombre = "Carlos Fuentes";
            miArregloAlumnos[1].edad = 22;
            miArregloAlumnos[1].evaluaciones = new float[3];
            miArregloAlumnos[1].evaluaciones[0] = 9.0F;
            miArregloAlumnos[1].evaluaciones[1] = 8.5F;
            miArregloAlumnos[1].evaluaciones[2] = 8.9F;
            miArregloAlumnos[2].nombre = "Alicia Suarez";
```



#### TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

```
miArregloAlumnos[2].edad = 24;
miArregloAlumnos[2].evaluaciones = new float[3];
miArregloAlumnos[2].evaluaciones[0] = 10.0F;
miArregloAlumnos[2].evaluaciones[1] = 6.8F;
miArregloAlumnos[2].evaluaciones[2] = 7.3F;
foreach (alumno al in miArregloAlumnos)
{
    cont++;
    Console.WriteLine("Alumno " + cont);
    prom = this.calcula_promedio(al.evaluaciones);
    Console.WriteLine("Nombre del alumno:{0}, edad: {1}, promedio: {2}", al.nombre, al.edad, prom);
}
}

float calcula_promedio(params float[] val)
{
    float resultado;
    float suma = 0F;
    int n = 0;
    try
    {
        foreach (float v in val)
        {
            suma = suma + v;
            n++;
        }
        resultado = suma / n;
        return resultado;
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
        return 0F;
    }
}
}
```



#### TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

2. Crea un nuevo proyecto y llámalo enumeraciones\_ejemplo.

2.1. A continuación ejecútalo y anota tus observaciones.

2.2. Comenta las partes que creas necesarias // o /\* \*/

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace enumeraciones_ejemplo
{
    class Program
    {
        public enum semana
        {
            lunes,
            martes,
            miercoles,
            jueves,
            viernes,
            sabado,
            domingo
        }
        static void Main(string[] args)
        {
            Program misemana = new Program();
            Console.WriteLine("*****Manipulacion directa de la semana*****");
            misemana.manipular_enumeracion();
            Console.WriteLine("\n*****Mostrar enumeracion completa*****");
            misemana.mostrar_enumeracion_completa();
            Console.WriteLine("\n*****Funcion con enumeracion*****");
            misemana.dia_hoy();
        }
        public void manipular_enumeracion()
        {
            Console.WriteLine(semana.domingo);
            Console.WriteLine((int)semana.domingo);
            Console.WriteLine(semana.lunes);
            Console.WriteLine((int)semana.lunes);
        }
        public void mostrar_enumeracion_completa()
        {
            Array arregloSemana;
            arregloSemana = semana.GetValues(typeof(semana));
            foreach (semana dia in arregloSemana)
            {
                Console.WriteLine(dia);
                Console.WriteLine((int)dia);
            }
        }
        public void dia_hoy()
        {
            Array arregloSemana;
            //Para una fecha determinada uso
            //DateTime dateValue = new DateTime(2008, 6, 11);
            DateTime fhoy = DateTime.Today;//calculo la fecha de hoy
            arregloSemana = semana.GetValues(typeof(semana));
            foreach (semana dia in arregloSemana)
            {
                if ((int)fhoy.DayOfWeek == (int)dia)
                    Console.WriteLine("Hoy es {0}", dia);
            }
        }
    }
}
```



**Ejercicio de clase.**

3. El siguiente ejercicio deberá realizarse en equipos de tres personas. Deberán realizar un sistema que simule el préstamos de libros en una biblioteca.

Deberán considerar dos estructuras: alumnos y libros. Cada estructura deberá contener la información que consideren necesaria.

El sistema contará con un método llamado préstamos que validará que el libro esté disponible y que el alumno exista. Para realizar sus pruebas deberán dar del alta 5 libros y 5 alumnos.

Al final deberá guardar en un archivo la relación de los libros que fueron prestados.

**NOTA:** Utilicen enumeraciones para determinar el área del libro.

Ejemplo:

Título: A, autor: AA, ID: 4567

Título: B, autor: BB, ID: 326

Alumno: Mario Gómez, No. Cuenta: 345678

Alumno: María Hernández, No. Cuenta: 334567



### Un ejemplo completo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace estructura_cine
{
    class Program
    {
        struct peliculas
        {
            public enum generos { terror=1, accion, comedia, musical, infantil }
            public enum clasificaciones { A=1, B, C }
            public int idpelicula;
            public string titulo;
            public generos genero;
            public string duracion;
            public clasificaciones clasificacion;
            public peliculas(int ID, string Titulo, generos Genero, string Duracion, clasificaciones Clasificacion)
            {
                idpelicula = ID;
                titulo = Titulo;
                genero = Genero;
                duracion = Duracion;
                clasificacion = Clasificacion;
            }
        }
        struct salas
        {
            public int idsala;
            public int idpelicula;
            public string hora;
            public string observaciones;

            public salas(int IDsala, int IDpelicula, string HR, string OBS)
            {
                idsala = IDsala;
                idpelicula = IDpelicula;
                hora = HR;
                observaciones = OBS;
            }
        }
        struct taquilla
        {
            public int idboleto;
            public int idsala;
            public string pelicula;
            public string hora;

            public taquilla(int IDboleto,int IDsala, string Pelicula, String HR)
            {
                idboleto=IDboleto;
                idsala=IDsala;
                pelicula=Pelicula;
                hora=HR;
            }
        }
        static void Main(string[] args)
        {
            Program miCine = new Program();
            /*****Ejemplo cargando una estructura desde teclado *****/
            peliculas miPeli;
            miPeli = miCine.inserta_pelicula(1);
            Console.WriteLine("Titulo: {0}", miPeli.titulo);
            Console.WriteLine("Genero: {0}",miPeli.genero);
            Console.WriteLine("Clasificacion: {0}", miPeli.clasificacion);
            Console.ReadKey();
            Console.Clear();
            /*****
            salas[] misSalas = miCine.inserta_salas();
            peliculas[] misPelis = miCine.inserta_peliculas();
            peliculas pelicula_seleccionada;
            salas sala_seleccionada;
```



#### TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

```
Console.WriteLine("Seleccione pelicula:");
int i=0;
foreach (peliculas p in misPelis)
{
    Console.WriteLine("{0}{1},Gen:{2},Clas:{3},Dur:{4}", i+1, p.titulo, p.genero, p.clasificacion, p.duracion);
    i++;
}
int idmovie = Convert.ToInt32(Console.ReadLine());
pelicula_seleccionada = misPelis[idmovie-1];

Console.WriteLine("Seleccione horario:");
i=0;
foreach (salas s in misSalas)
{
    if (s.idpelicula == pelicula_seleccionada.idpelicula)
        Console.WriteLine("{0})Sala: {1}, {2},{3}", i + 1, s.idsala, s.hora, s.observaciones);
    i++;
}
int idhorario = Convert.ToInt32(Console.ReadLine());
sala_seleccionada = misSalas[idhorario - 1];
Console.WriteLine("El boleto impreso es:");
taquilla boleto= new taquilla(1,sala_seleccionada.idsala,pelicula_seleccionada.titulo, sala_seleccionada.hora);
Console.WriteLine("No. boleto: {0},Pelicula: {1}, Sala: {2}, Hora:{3}", boleto.idboleto, boleto.pelicula, boleto.idsala, boleto.hora);
}
peliculas[] inserta_peliculas()
{
    peliculas[] P = new peliculas[3];
    P[0] = new peliculas(1, "Shrek", peliculas.generos.infantil, "1:20", peliculas.clasificaciones.A);
    P[1] = new peliculas(2, "Titanic", peliculas.generos.comedia, "3:20", peliculas.clasificaciones.B);
    P[2] = new peliculas(3, "Aliens", peliculas.generos.terror, "2:20", peliculas.clasificaciones.B);
    return P;
}

peliculas inserta_pelicula(int ID)
{
    string Titulo;
    string Duracion;
    int Genero;
    int Clasificacion;
    int cont;
    peliculas P;
    Console.WriteLine("Titulo de la pelicula: ");
    Titulo = Console.ReadLine();
    Console.WriteLine("Duración de la pelicula: ");
    Duracion = Console.ReadLine();
    Console.WriteLine("Seleccione el genero:");
    cont = 0;
    foreach (string g in peliculas.generos.GetNames(typeof(peliculas.generos)))
    {
        cont++;
        Console.WriteLine("{0}{1}", cont, g);
    }
    Genero = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Seleccione la clasificaciones:");
    cont = 0;
    foreach (string g in peliculas.clasificaciones.GetNames(typeof(peliculas.clasificaciones)))
    {
        cont++;
        Console.WriteLine("{0}{1}", cont, g);
    }
    Clasificacion = Convert.ToInt32(Console.ReadLine());
    P = new peliculas(ID, Titulo, (peliculas.generos)Genero, Duracion, (peliculas.clasificaciones) Clasificacion);
    return P;
}
```



TEMA3. ESTRUCTURAS DE DATOS COMPUESTAS

```
salas[] inserta_salas()
{
    salas[] S = new salas[12];
    S[0] = new salas(1, 1, "10:00", "IMAX");
    S[1] = new salas(1, 1, "13:00", "IMAX");
    S[2] = new salas(1, 1, "16:00", "IMAX");
    S[3] = new salas(1, 1, "19:00", "IMAX");
    S[4] = new salas(1, 1, "22:00", "IMAX");
    S[5] = new salas(2, 2, "10:00", "VIP");
    S[6] = new salas(2, 2, "13:00", "VIP");
    S[7] = new salas(3, 2, "16:00", "");
    S[8] = new salas(3, 2, "19:00", "");
    S[9] = new salas(3, 2, "22:00", "");
    S[10] = new salas(4, 3, "19:00", "");
    S[11] = new salas(4, 3, "22:00", "");
    return S;
}
}
```