



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

1. Tipos de datos en CSharp

Cuando trabajamos en lenguaje CSharp, cada dato es de un tipo concreto. Este tipo de dato define el conjunto de valores válidos que el dato puede tomar y el conjunto de operaciones que se pueden realizar con él.

Tipos de Datos	Nombre	C#
Primitivos	Numericos	<code>int a = 10;</code> <code>int b = 7;</code> <code>int c=a+b;</code> <code>int c=a-b;</code> <code>bool r =a > b;</code>
	Lógicos (Booleanos)	<code>bool a=false;</code> <code>bool b=true;</code> <code>bool c=a && b;</code> <code>bool d=a b;</code> <code>bool e= !a;</code>
	Caracteres	<code>char a='a';</code> <code>char b='d';</code> <code>bool r=a > b;</code>
Compuestos y Agregados	Array	<code>int[] a=new int[5];</code> <code>a[0] = 10;</code> <code>a[1] = -2;</code> <code>a[2] = 5;</code>
	Cadena	<code>string a="Hola Mundo!";</code> <code>a.Length</code> <code>bool r=String.Compare(s1 ,s2);</code>
	Estructuras	<code>struct Libro p = {"Robert", "Algorithms"};</code> <code>p.autor = "Robert";</code> <code>q = p;</code>
Abstractos	Clases Listas Pilas Colas Árboles	<code>Datetime hoy = DateTime.Now;</code> <code>ArrayList legajos = new ArrayList();;</code> <code>Stack miPila = new Stack();;</code> <code>Queue fila = new Queue();;</code>

En esta seccion se va a centrar el tipo de dato que se llama Estructura.



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

1.1. Estructuras - struct

1.1.1. Conceptos

Los vectores/arrays son estructuras de datos que pueden tener una cantidad grande de elementos, donde todos ellos deben ser del mismo tipo de datos, es una colección o estructura homogénea. Esto es muy útil para manipular datos como alturas, temperaturas, nombre de países pero no sirven si la información que se quiere utilizar tiene distintos tipos de datos como la información de un socio de un club donde podríamos encontrar el nombre de la persona(string), su dirección(string) pero también su edad(int), fecha de nacimiento(datetime), actividades a las que esta inscripto(array de deportes), nro. de socio(int), valor de su cuota(float), etc.

Para agrupar esta información bajo un mismo nombre (una ficha del socio) se puede utilizar un tipo de datos definido por el usuario, una estructura, un "struct". Una estructura contiene múltiples campos, variables, que pueden ser de tipos diferentes. También se puede pensar que una estructura es un tipo de dato "a medida".

Entonces, las estructuras son colecciones heterogéneas (tuplas, registros) cuyos elementos pueden ser de distintos tipos. Los elementos que componen una estructura se llaman campos y están identificados por un nombre. Cada campo puede tener información relacionada que se puede manejar como una unidad. El acceso a cada elemento se realiza a través del nombre del campo con el operador . (punto)

En general los campos que se definen dentro de una estructura tienen algún tipo de relación, por ejemplo si se piensa en un legajo de un club donde se ponen todos los datos del socio del club, su nombre, edad, nacimiento, actividades que realiza en el club, última fecha de pago, etc. Todos estos datos se pueden agrupar en una estructura.

Ficha de Socios del Club

Completar en letra de imprenta, con birome y lo más claro posible.

Datos personales del socio

• Nombre: _____

• Apellido: _____

• Direccion: _____

• Edad _____

• Fecha nacimiento(dia/mes/año) ____ / ____ / ____

• DNI: _____

Actividades

- _____
- _____
- _____
- _____
- _____

Pagos

• Fecha ultimo pago(dia/mes/año) ____ / ____ / ____

• Importe: _____



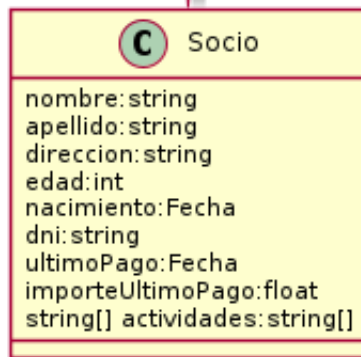
GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

representa la estructura de un socio



La estructura es importante en la creación de programas en general, como por ejemplo bases de datos u otras aplicaciones que requieran grandes cantidades de datos.

1.1.2. Declaración

Para poder usar una estructura para almacenar información se la debe definir y declarar antes de que se pueda utilizar. El formato de la declaración es:

```
struct <Nombre de la estructura>
{
    <tipo de dato miembro 1 > <nombre miembro 1 >
    <tipo de dato miembro 2 > <nombre miembro 2 >
    <tipo de dato miembro 3 > <nombre miembro 3 >
    . . . . .
    <tipo de dato miembro n > <nombre miembro n >
};
```

Por ejemplo en el siguiente ejemplo se define una estructura para guardar cierta información relativa a las personas, por ejemplo para ser utilizadas en un instituto, en un sistema de gestión de turnos, etc.

```
struct Persona
{
```



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
string nombre;  
string apellido;  
int edad;  
string dni;  
};
```

Para una aplicacion matematica que realice calculos con numeros complejos y operaciones sobre objetos geometricos podria definir

```
struct NComplejo  
{  
    int real;  
    int imaginario;  
};  
  
struct Punto  
{  
    int x;  
    int y;  
};  
  
struct Circulo  
{  
    Punto centro;    // observar como una estructura puede incluir otras.  
    int radio;  
};  
  
struct Rectangulo  
{  
    int x1, y1;  
    int x2, y2;  
    int x3, y3;  
    int x4, y4;  
};  
  
// otra forma incluyendo otras estructuras.  
struct RectanguloV2  
{  
    Punto v1, v2, v3, v4;  
};
```



GOBIERNO DE LA CIUDAD DE BUENOS AIRES
Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

En el siguiente ejemplo podría ser una estructura definida dentro de un programa que maneje las ventas de un local, en particular en este caso los datos de un cliente.

```
struct Fecha
{
    short dia;
    short mes;
    short anio;
};
struct Cliente
{
    Persona datosBasicos;
    float credito;
    float descuento;
    string preferencias;
    Fecha ultimaCompra;
    string opiniones;
    string email;
};
```

1.1.3. Inicialización

Se puede inicializar una estructura en forma similar como cualquier otra variable cuando la declaramos. La forma es agregar luego de la declaración de la variable y el signo igual , entre llaves, los valores que corresponden a cada campo.

Por ejemplo

```
. . . . .
int x = 10 ; // una variable cualquiera

Fecha nacimientoNewton = new Fecha { dia = 4 , mes = 1 , anio=1643 };
Fecha llegadaALaLuna    = new Fecha { dia = 16 , mes = 7 , anio=1969};
. . . . .
```

Un ejemplo mas completo donde se define una estructura para guardar datos de libros y en el main se definen dos variables y se las inicializa.



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
using System;

namespace source
{
    class Program
    {
        public struct tLibro
        {
            public string titulo;
            public string autor;
            public string editorial;
            public int anio;
        };

        static void Main(string[] args)
        {
            var libroPreferido = new tLibro
            {
                titulo = "El mitico hombre-mes",
                autor = "Frederik P. Brooks, Jr",
                editorial = "Universidad de Carolina del Norte",
                anio = 1975
            };

            var otroLibro = new tLibro
            {
                titulo = "La clave es la servilleta",
                autor = "Dan Roam",
                editorial = "Editorial Norma",
                anio = 2009
            };

            Console.WriteLine("Libro Preferido");
            Console.WriteLine("Titulo del libro:" + libroPreferido.titulo);
            Console.WriteLine("Autor del libro: " + libroPreferido.autor);
            Console.WriteLine("Editado por " + libroPreferido.editorial);
            Console.WriteLine("Año de Edicion:" + libroPreferido.anio);

            Console.WriteLine();
            Console.WriteLine("Otro buen libro");
        }
    }
}
```



GOBIERNO DE LA CIUDAD DE BUENOS AIRES
Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
        Console.WriteLine("Titulo del libro:" + otroLibro.titulo);  
        Console.WriteLine("Autor del libro: " + otroLibro.autor);  
        Console.WriteLine("Editado por " + otroLibro.editorial);  
        Console.WriteLine("Año de Edicion:" + otroLibro.anio);  
    }  
}
```

1.1.4. Acceso a los campos con el operador punto

Tanto para asignar, operar, mostrar y acceder a los campos de una estructura se hace utilizando el operador punto (.). La sintaxis es la siguiente:

<Nombre variable estructura>.<Nombre del campo de la estructura> = . . .

Por ejemplo dado una variable denominada persona se le asignan algunos datos

```
persona.nombre = "Juan";  
persona.apellido = "Perez";  
persona.edad = 35;  
persona.dni = "30.222.111";
```

También si se necesita copiar una estructura a otra, en vez de copiar campo por campo se puede asignar el contenido entre dos variables del mismo tipo:

```
Persona persona2;  
persona2 = persona; // copia los 4 campos a la vez
```

También se pueden pasar por parámetros las estructuras y una función puede devolver una estructura, igual que cualquier otros tipo de dato. Observar la siguiente función llamada *sumaComplejos*.

```
NComplejo sumaComplejos(NComplejo a, NComplejo b)  
{  
    NComplejo resultado; // variable local
```




GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
    resultado.real = a.real + b.real;
    resultado.imaginario = a.imaginario + b.imaginario;
    return resultado; // devuelve una variable del tipo NComplejo
}
```

En el siguiente ejemplo, algo mas completo, se implementa un programa que suma números complejos ¹.

```
using System;

namespace source
{
    class Program
    {
        public struct NComplejo
        {
            public int real;
            public int imaginario;
        }

        NComplejo sumaComplejos(NComplejo a, NComplejo b)
        {
            NComplejo resultado; // variable local
            resultado.real = a.real + b.real;
            resultado.imaginario = a.imaginario + b.imaginario;
            return resultado; // devuelve una variable del tipo NComplejo
        }

        // funcion que recibe un nro complejo y lo muestra por consola.
        string ComplejoToString(NComplejo a)
        {
            string s;
            s = a.real.ToString() ;
            s += (a.imaginario >= 0 ? "+" : "");
            s += a.imaginario.ToString()+"i";

            return s;
        }
    }
}
```

¹Para sumar números complejos se suma por un lado la parte real y por otro lado la parte imaginaria



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
void OperacionesConComplejos()
{
    var a = new NComplejo { real = 4, imaginario = 10 };
    var b = new NComplejo { real = 4, imaginario = 10 };

    /* pasamos dos estructuras por parametros y recibimos una
    estructura como resultado
    */
    var resul = sumaComplejos(a, b);

    Console.WriteLine("(" + ComplejoToString(a) + ")" + " + " + "(" +
        ComplejoToString(b) + ")" + " = " +
        ComplejoToString(resul));
}

static void Main(string[] args)
{
    var prog = new Program();
    prog.OperacionesConComplejos();
}
}
```

1.1.5. Métodos en estructuras

Una estructura además de "agrupar" campos y datos que están relacionados conceptualmente pueden tener métodos asociados a la misma estructura. En el ejemplo de la estructura de socio por ejemplo se pueden agregar tres métodos, uno que consulta el estado del socio si esta en mora o no (si pago o no su cuota social) como también otro método que verifica si el socio esta inscripto o no en alguna actividad del club.

Es importante tener en cuenta que el método **solo ve** las variables de la estructura y las que pudo haber recibido, no tiene acceso a otras variables². Estos métodos además están declarados **dentro** de la estructura.

²Excepto variables estáticas de otras clases, esto se verá más adelante.

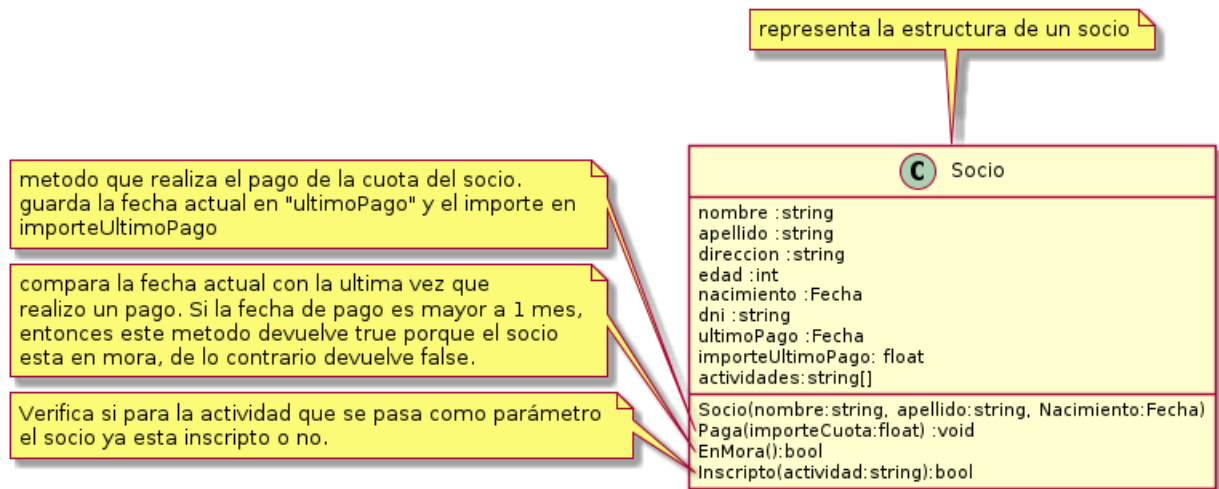


GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda



Implementacion de la estructura

```
using System;
using System.Text;
struct Fecha
{
    int dia;
    int mes;
    int anio;
    public Fecha(int d, int m, int a)
    {
        dia = d;
        mes = m;
        anio = a;
    }

    public int diferencia(DateTime f)
    {
        var x = new DateTime(anio, mes, dia);
        var dif = f.Subtract(x);
        return dif.Days;
    }

    public string getTexto(){
        var s = dia.ToString() + "/" + mes.ToString() + "/" + anio.ToString();
        return s;
    }
}
```



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
};
```

```
struct Socio
{
    public string nombre;
    public string apellido;
    public string direccion;
    public int edad;
    public Fecha nacimiento;
    public Fecha ultimoPago;
    public float importeUltimoPago;
    public string[] actividades;

    public Socio(string nom, string ape, Fecha naci)
    {
        nombre = nom;
        apellido = ape;
        direccion = "";
        nacimiento = naci;
        edad = (int)Math.Floor(nacimiento.diferencia(DateTime.Now) / 365f);
        ultimoPago = new Fecha(1, 1, 1);
        importeUltimoPago = 0;
        actividades = new string[10];
    }

    public void Paga(float importeCuota)
    {
        importeUltimoPago = importeCuota;
        ultimoPago = new Fecha(DateTime.Now.Day, DateTime.Now.Month,
            ↪ DateTime.Now.Year);
    }

    public bool enMora()
    {
        int difDias = ultimoPago.diferencia(DateTime.Now);
        if (difDias >= 30)
            return true;
        else return false;
    }
}
```

Programa ejemplo que usa la estructura Socio



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
using System;

namespace source
{
    class Program
    {
        static void Main(string[] args)
        {
            var socio1 = new Socio("Lionel", "Messi", new Fecha(25, 6, 1987));
            var socio2 = new Socio("Damián Emiliano ", "Martínez", new Fecha(2, 9,
                ↪ 1992));
            Console.WriteLine("socio 1");
            Console.WriteLine("Nombre:"+socio1.nombre);
            Console.WriteLine("edad " + socio1.edad );
            Console.WriteLine();
            Console.WriteLine("socio 2");
            Console.WriteLine("Nombre:"+socio2.nombre);
            Console.WriteLine("edad " + socio2.edad );
            Console.WriteLine();
            Console.WriteLine("socio 1");
            Console.WriteLine("en Mora? " + socio1.enMora() );
            Console.WriteLine("paga....");
            socio1.Paga(1000.00f);
            Console.WriteLine("Sigue en Mora? " + socio1.enMora() );
            Console.WriteLine("Fecha ultimo pago:"+ socio1.ultimoPago.getTexto());
            Console.WriteLine();
            Console.WriteLine("socio 2");
            Console.WriteLine("en Mora? " + socio2.enMora() );

        }
    }
}
```

1.1.6. Constructor

Los constructores son un tipo especial de metodo dentro de una estructura. Tiene la particularidad de ejecutarse **cada vez** que se crea una variable de una estructura o clase. Una clase o estructura puede tener varios constructores que toman argumentos diferentes. Sirven para inicializar las variables internas de la estructura o realizar tareas de inicialización. Son extremadamente útiles y ayudan a escribir código flexible, promover las buenas practicas



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

en el desarrollo de software, además de facilitar la lectura del código.

El constructor tiene algunas características que lo distinguen de un método ordinario

- Tiene el mismo nombre que la estructura.
- Es el primer método que se ejecuta.
- Se ejecuta en forma automática.
- No puede retornar datos.
- Se ejecuta una única vez.
- Un constructor tiene por objetivo inicializar las variables o atributos.

En los ejemplos anteriores del código, se mostró como se pueden usar, ver la estructura Fecha, aunque los constructores pueden ser extremadamente complejos!

En el siguiente código se muestra el main de un sistema que muestra el valor del dolar en Argentina. Como se puede apreciar, el código es muy simple.

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Dolar dolar = new Dolar(true);
        Console.WriteLine("Valor del dolar oficial en Argentina");
        Console.WriteLine("Compra    " + dolar.compra);
        Console.WriteLine("Venta      " + dolar.venta);
    }
}
```

Este código del Main es muy simple y fácil de leer e interpretar, sin embargo, el constructor de la estructura "Dolar" puede ser muy complejo.

Incluso la estructura, el diagrama sigue siendo simple.



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

si cargaViaWeb es True, entonces busca el valor del dolar en la web de lo contrario compra y venta valen 0.

C Dolar

compra:double
venta: double

Dolar(cargaViaWeb:bool)

En el siguiente programa se muestra el código del constructor en el cual inicializa los valores de las variables "compra" y "venta" buscando en la web los valores actuales.

Algunas aclaraciones sobre el código:

- No es necesario entender todo este código, incluso se trato de mantenerlo "simple" sin pretender ser completo ni eficiente.
- Lo importante es interpretar que los constructores pueden ser tan grandes como sea necesario.
- Las clases Casa y Root son necesarias para interpretar la respuesta del sitio web (no es necesario prestarle mucha atención por ahora).
- Ubicar la estructura Dolar ,su constructor y donde se hace la inicialización de las variables "compra" y "venta".
- Para que funcione necesita tener acceso a Internet.

```
using System;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
using System.Text.Json;

public class Casa
{
    public string compra { get; set; }
    public string venta { get; set; }
    public string nombre { get; set; }
}

public class Root
{
    public Casa casa { get; set; }
```



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
}
```

```
struct Dolar
{
    public double compra;
    public double venta;
    public Dolar(bool cargaViaWeb)
    {
        compra = 0;
        venta = 0;

        if (cargaViaWeb)
        {
            try
            {
                var uri =
                    ↪ "https://www.dolarsi.com/api/api.php?type=valoresprincipales";
                var httpClient = new HttpClient();
                var content = new StringContent("", Encoding.UTF8,
                    ↪ "application/json");
                content.Headers.ContentType = new
                    ↪ MediaTypeHeaderValue("application/json");

                var task = httpClient.GetAsync(uri);
                var respuesta = task.GetAwaiter().GetResult();

                switch (respuesta.StatusCode)
                {
                    case System.Net.HttpStatusCode.OK:
                        var task2 = respuesta.Content.ReadAsStringAsync();
                        var respuestaString = task2.GetAwaiter().GetResult();
                        var resp =
                            ↪ JsonSerializer.Deserialize<Root[]>(respuestaString);

                        foreach (var linea in resp)
                        {
                            if (linea.casa.nombre == "Dolar Oficial")
                            {
                                /*
                                *
```




GOBIERNO DE LA CIUDAD DE BUENOS AIRES
Instituto de Formación Técnica Superior Nro 11

Materia: Desarrollo De Sistemas Orientado
A Objetos

Año 2022 Docente Ing. Pablo Avellaneda

```
→ con los

* Aqui esta la inicializacion de las variables

* datos encontrados en la web
*
*/
venta = Double.Parse(linea.casa.venta.Replace(',', ' '),
    ↪ '.');
compra =
    ↪ Double.Parse(linea.casa.compra.Replace(',', ' '),
    ↪ '.');
    }
}
break;

default:
    Console.WriteLine("Error:" + respuesta.StatusCode);
    break;
}
}
catch (Exception e)
{
    Console.WriteLine("Error!" + e.Message);
}
}
}
}
```