

# **Desarrollo De Sistemas Orientado A Objetos**

## **Estructuras**

Profesor Ing Pablo Avellaneda  
[pablo.avellaneda@bue.edu.ar](mailto:pablo.avellaneda@bue.edu.ar)

# Tipos de datos en C#

**Tipos de datos Simples o Primitivos**

numéricos, lógicos, carácter, etc.

**Tipos de datos Compuestos/Agregados**

cadenas, arrays, estructuras, etc.

**Tipos de datos Abstractos**

clases, listas, pilas, colas, etc.

**Ejemplo: En el instituto, que informacion suele manejar?**

**Datos de alumnos**

**Datos de profesores**

**Datos de cada materia**

**Datos de cada carrera**

**Datos del lugar donde se cursa**

# Ejemplo: En el instituto, que informacion suele manejar?

Datos de alumnos: que informacion necesito de cada uno?

- ▶ Nombre
- ▶ Apellido
- ▶ Edad
- ▶ Dni

Cada pieza de informacion que se decide tener tiene que tener sentido en el contexto del problema

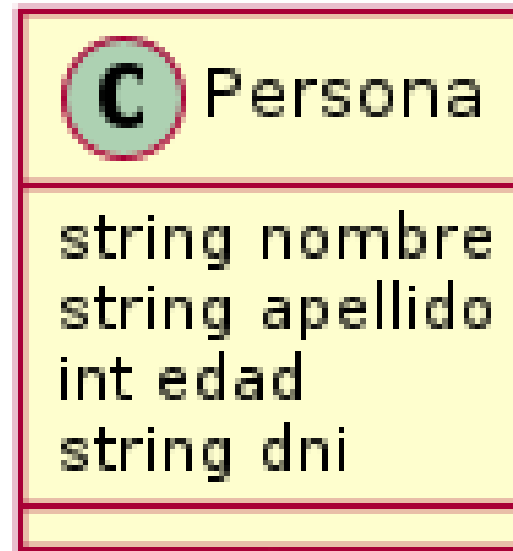
- ▶ ~~Mascotas que tiene~~

Para que guardar esta informacion ????

# Atributos de la Estructura

- ▶ Son colecciones heterogéneas → elementos de distinto tipo
- ▶ Los elementos que la componen se llaman ATRIBUTOS
- ▶ Los atributos están identificados con un nombre
- ▶ Para acceder a los atributos se usa el operador punto (.) a través de su nombre

# Estructuras en UML



representa la estructura de una persona

# Declaración de estructuras en C#

**C** Persona

string nombre  
string apellido  
int edad  
string dni

```
struct Persona  
{  
    public string Nombre;  
    public string Apellido;  
    public int Edad;  
    public string DNI;  
  
}
```

Los atributos de la estructura pueden tener distinto tipo

No se permite inicializar los atributos en la declaración

## Ejercicio:



- ▶ Pensar en tu trabajo o en el trabajo de un familiar, o elegir cualquier negocio y pensar en que “cosas” manejan y que tipo de datos tiene cada “cosa” ( recordar el ejemplo del instituto- alumnos)
- ▶ Dibujar un grafico UML con la estructura y sus atributos ( por ahora usar tipos de datos basicos, int, string, bool...)
- ▶ Programar en C# la estructura

10 minutos





# Inicialización de estructuras en C#

```
public void main()
{
    Persona alumno1 = new Persona
    {
        Nombre = "Juan",
        Apellido = "Perez",
        Edad = 21,
        DNI = "39111222"
    };
    Persona alumno2 = new Persona
    {
        Nombre = "Maria",
        Apellido = "Lopez",
        Edad = 20,
        DNI = "40111222"
    };
}
```

Inicialización:  
se agregan entre **llaves { }**  
los valores de cada atributo

# Acceso a atributos de estructuras en C#

```
public void main()
{
    Persona alumno1;

    alumno1.Nombre = "Juan";
    alumno1.Apellido = "Perez";
    alumno1.Edad = 21;
    alumno1.DNI = "39111222";

}
```

Se usa como cualquier otro tipo de dato para declarar una variable

Se accede a cada atributo mediante el **operador punto (.)**

# Uso de estructuras en C#

```
public void main()
{
    Persona alumno1;
    alumno1.Nombre = "Juan";
    alumno1.Apellido = "Perez";
    alumno1.Edad = 21;
    alumno1.DNI = "39111222";

    System.Console.WriteLine(alumno1.Nombre);
    System.Console.WriteLine(alumno1.Apellido);

    Persona alumno2;
    alumno2 = alumno1;
}
```

Se puede usar como cualquier otra variable dentro de expresiones o como parámetro de un método

Se puede asignar una variable de estructura a otra variable de estructura del mismo tipo

## Ejercicio:



- ▶ Tomar al menos (una o mas) estructuras que hayas definido previamente.
- ▶ hacer un programa (dentro del main...) que declare al menos una variable del tipo “estructura” que haya elegido y programar su inicialización usando llaves {}.
- ▶ En el mismo programa, declarar otra variable e inicializarla usando el punto (.)
- ▶ Mostrar en la consola ( `console.WriteLine()` ) los valores de las variables

10 minutos

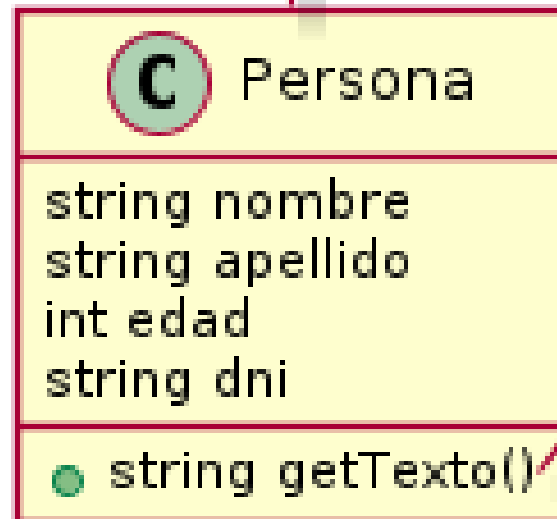


# Métodos de la Estructura

- ▶ Se declaran dentro de la estructura
- ▶ Solo ven los atributos de la estructura y las variables que recibe
- ▶ Hay un tipo de método especial llamado Constructor

# Métodos de Estructuras en UML

representa la estructura de una persona



metodo de la estructura que concatena el apellido y el nombre separados por una coma

# Definir métodos de estructuras en C#

```
struct Persona
{
    public string Nombre;
    public string Apellido;
    public int Edad;
    public string DNI;

    public string getTexto()
    {
        return Apellido + ", " + Nombre;
    }
}

static void Main()
{
    Persona alumno1;
    alumno1.Nombre = "Juan";
    alumno1.Apellido = "Perez";
    alumno1.edad = 21;
    alumno1.DNI = 39111222;
    String texto = alumno1.getTexto();
    Console.WriteLine("Alumno: " + texto);
}
```

Método de la estructura

Llama al Método de la estructura

## **RECORDAR:**

Los tipos de datos se definen por:

- Conjunto de valores válidos
- Conjunto de operaciones

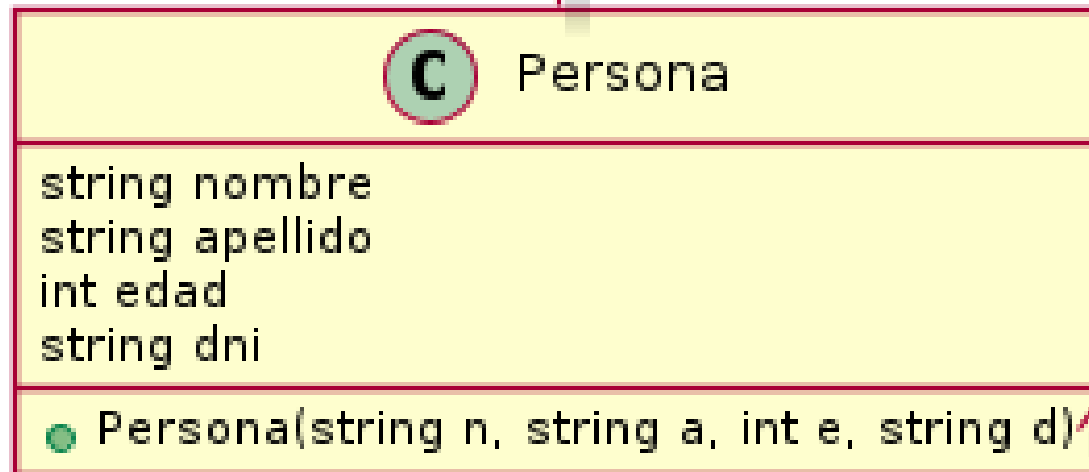
# Método Constructor de la Estructura

- ▶ 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 al crear la variable
- ▶ En general, su objetivo es inicializar los atributos



# Método constructor de Estructuras en UML

representa la estructura de una persona



metodo constructor de la estructura

# Método Constructor de estructuras

```
struct Persona
{
    public string Nombre;
    public string Apellido;
    public int Edad;
    public string DNI;
    public Persona(string n, string a, int e, string d)
    {
        Nombre = n;
        Apellido = a;
        Edad = e;
        DNI = d;
    }
}

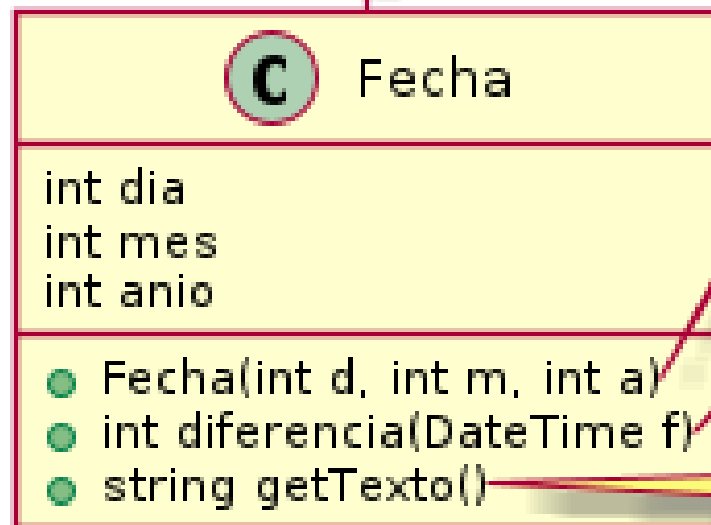
public static void Main()
{
    Persona alumno1 = new Persona("Juan","Perez",21,"39111222");
    Persona alumno2 = new Persona("Maria", "Lopez", 20, "40111222");
}
```



Método constructor

# Estructura completa en UML- Ejemplo

representa la estructura de una fecha



metodo constructor que inicializa los atributos

metodo que calcula la cantidad de días entre la fecha que recibe por parámetro y la propia de la estructura

metodo que devuelve la fecha en formato "dd/mm/aaaa"

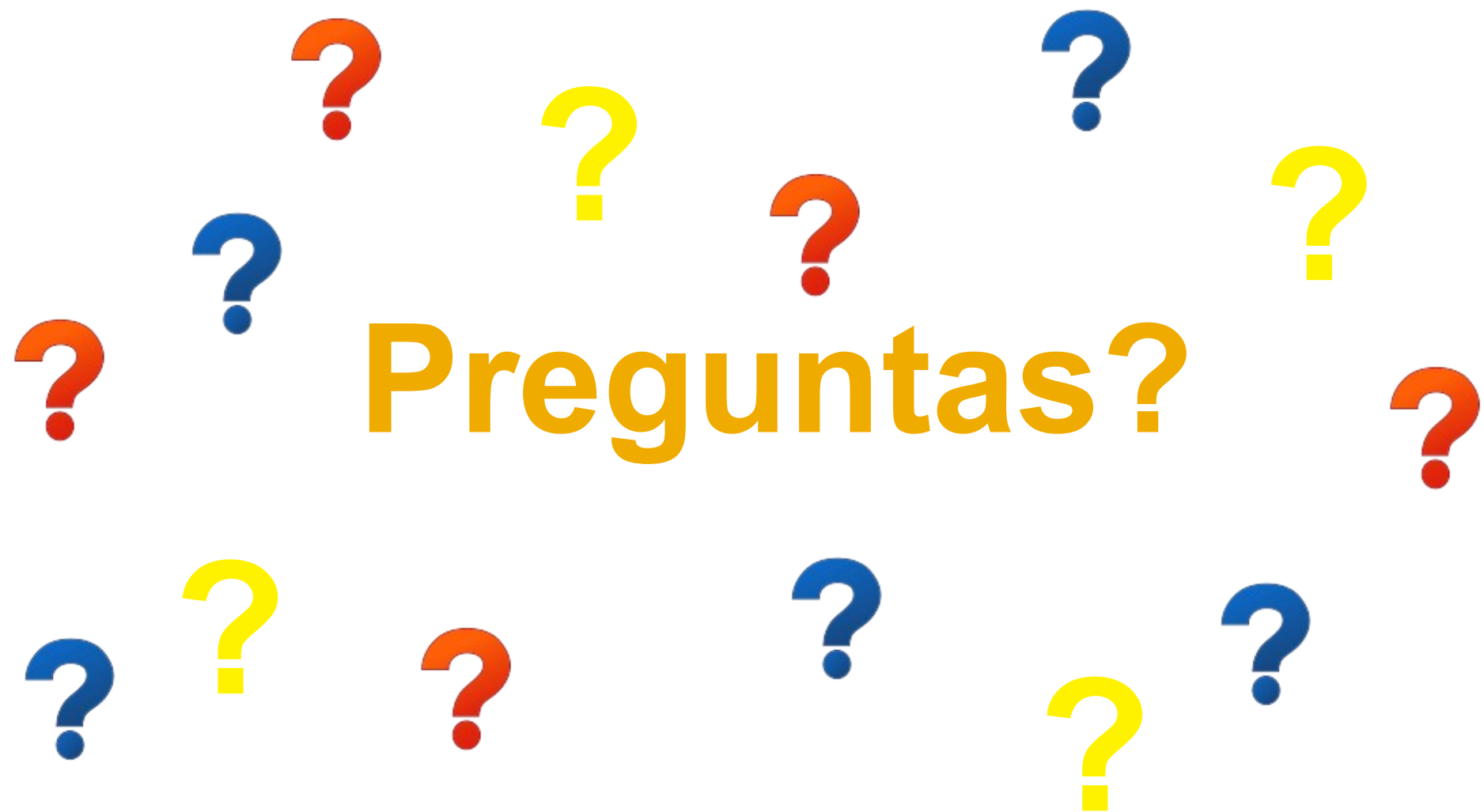
# Estructura completa en C#

```
struct Fecha
{
    public int dia;
    public int mes;
    public int anio;
    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()
    {
        return dia.ToString() + "/" + mes.ToString() + "/" + anio.ToString();
    }
}

static void Main()
{
    Fecha cumple = new Fecha(13,11,2021);
    var diasRestan = cumple.diferencia(DateTime.Now);
    Console.WriteLine("mi cumple es el " + cumple.getTexto() + " y viví " + diasRestan + " dias");
}
```

## Que vimos?

- ▶ Identificar elementos del problema.
- ▶ Definir estructuras.
- ▶ Representar en un diagrama una estructura.
- ▶ Atributos y metodos.
- ▶ Declarar variables del tipo estructura y usarlas.
- ▶ Método constructor.



**FIN**