

UT 3 - Componentes

INTRODUCCIÓN C#

Visual Studio 2019 - C#

Sintaxi C# :

- ✓ File I/O
- ✓ Clases
- ✓ TBC

Visual Studio 2019 - C#

Sintaxi C# - UT3 - C# File I/O

✓ **Lectura y escritura de ficheros**

```
//Escribe todo el contenido en el fichero
```

```
string path = "C:\\File\\file.txt";
```

```
string Text = "Hello, Hi, ByeBye";
```

```
File.WriteAllText(path, Text);
```

```
//Lee todo el contenido del fichero
```

```
string path = "C:\\File\\file.txt";
```

```
string readText = File.ReadAllText(path);
```


Visual Studio 2019 - C#

✓ **Escritura de ficheros** //escribe línea a línea

```
try
{
    //Pass the filepath and filename to the StreamWriter Constructor
    StreamWriter sw = new StreamWriter("C:\\Test.txt");
    //Write a line of text
    sw.WriteLine("Write line by line!!");
    //Close the file
    sw.Close();
}
catch(Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
finally
{
    Console.WriteLine("Executing finally block.");
}
```

Visual Studio 2019 - C#

✓ Lectura de ficheros `//leer línea a línea`

```
try
{
    StreamReader sr = new StreamReader("C:\\Sample.txt");
    line = sr.ReadLine();
    while (line != null)
    {
        Console.WriteLine(line);
        line = sr.ReadLine();
    }
    sr.Close();
    Console.ReadLine();
}
catch(Exception e) {
    Console.WriteLine("Exception: " + e.Message);
}
finally{
    Console.WriteLine("Executing finally block.")
}
```

Visual Studio 2019 - C#

✓ **Classes - Getters y setters**

```
class Persona{  
    private string name;  
  
    public void setName(string name) {  
        this.name = name;  
    }  
  
    public string getName() {  
        return this.name;  
    }  
}
```

Visual Studio 2019 - C#

✓ **Classes - Getters y setters (otras maneras)**

```
class Persona{  
    private int _name;  
    public string Name {  
        get { return name; }  
        set { name = value; }  
    }  
}
```

```
class Persona{  
    private int _name;  
    public string Name { get; set; }  
}
```


Visual Studio 2019 - C#

✓ Varios

❏ Nomenclaturas -> elementos publicos

La definición del nombre de una clase:

- Se debe utilizar un sustantivo en singular para nombrar una clase.
- Utilizar la notación de Pascal (**PascalCase**).
- No se debe utilizar el prefijo “_” (guión bajo).

Visual Studio 2019 - C#

✓ Varios

❏ Nomenclaturas -> elementos privados

La definición del nombre de los atributos privados está guiada por las siguientes reglas:

- Utilizar notación camello (**camelCase**).
- Cuando el atributo es privado, se recomienda identificar el nombre de la variable con el prefijo “_” (guión bajo).

Visual Studio 2019 - C#

✓ Varios

❑ Modificadores de acceso

Private (por defecto) - Public

❑ Static atributo que no se replicara en las instancias

Visual Studio 2019 - C#

✓ Varios

❏ Static:

-> **clase:** no puede usar el operador `new` para crear una variable del tipo de clase.

`Math.Pow()`

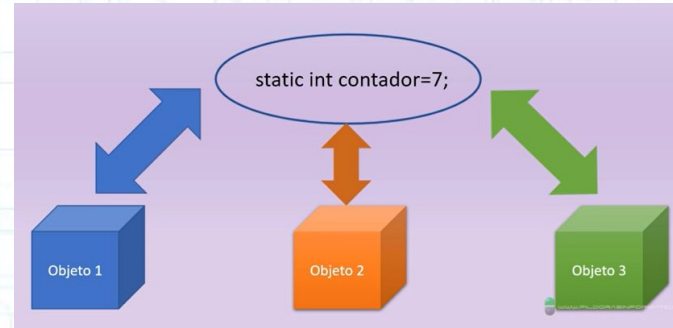
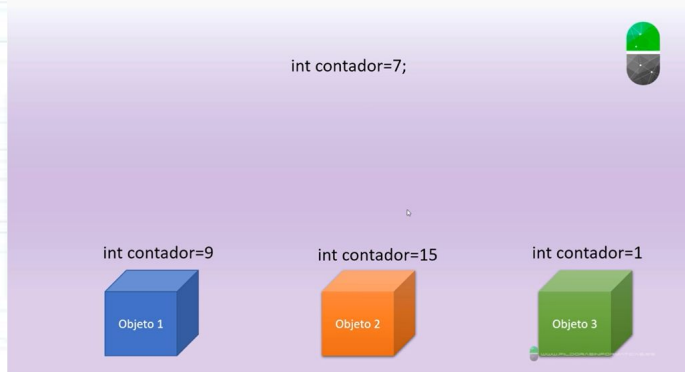
-> **atributos y métodos** de una clase, indica que es una "variable de clase", es decir, que su valor es el mismo para todos los objetos de la clase.

Visual Studio 2019 - C#

✓ Varios

❏ Static:

-> **atributo** de una clase, indica que es una "variable de clase", es decir, que su valor es el mismo para todos los objetos de la clase.



La variable estatoca es compartida por las instancias (pertenece a la clase)