

**www.PragimTech.com**

Training +Placements = Our success

Pragim@PragimTech.com

## ► **Parte 27 – Propiedades en C#**

**Cristina Carrasco**

PRAGIM Technologies

Cristina@cristinacarrasco.com

# En esta lección veremos:

- ▶ Propiedades de lectura y escritura
- ▶ Propiedades de solo lectura
- ▶ Propiedades de solo escritura
- ▶ Propiedades Auto implementadas
- ▶ Descriptor de acceso **init**

**[www.PragimTech.com](http://www.PragimTech.com)**

Training +Placements = Our success  
Pragim@PragimTech.com

# Propiedades

En C# para encapsular y proteger campos utilizamos las propiedades

1. Usamos los descriptores de acceso de propiedades **get** y **set** para implementar las propiedades.
2. Una propiedad con ambos descriptores de acceso **get** y **set** es una propiedad de **lectura y escritura**.
3. Una propiedad con solo el descriptor de acceso **get** es una propiedad de **solo lectura**.
4. Una propiedad con solo el descriptor de acceso **set** es una propiedad de **solo escritura**.

**Nota:** La ventaja de las propiedades sobre los métodos **Get()** y **Set()** es que puedes acceder a estas como si fueran campos públicos de la clase.

# Propiedades auto implementadas

Si no hay lógica adicional para los descriptores de acceso de propiedades podemos hacer uso de las propiedades auto implementadas las cuales fueron introducidas a partir de la versión de C# 3.0.

Las propiedades auto implementadas reducen la cantidad de código que tenemos que escribir.

Cuando utilizamos propiedades auto implementadas, el compilador crea un campo privado el cual puede ser accedido a través del **get** y **set** de la propiedad.



# Descriptor de acceso Init

El descriptor de acceso **init** fue introducido a partir de la versión C# 9.0.

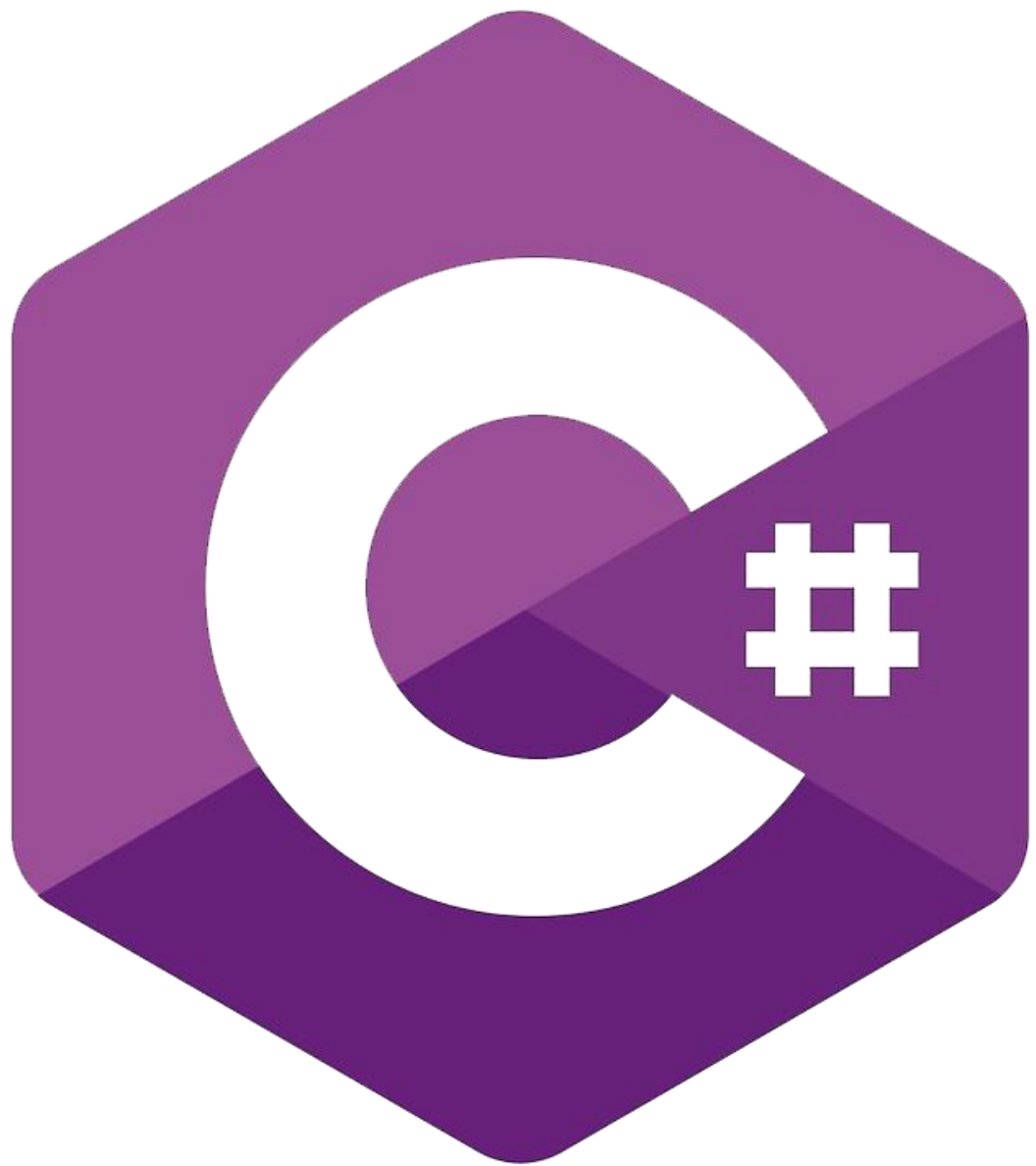
Init se utiliza en lugar de **set** y esto indica que solo se le puede asignar valor a la propiedad durante la construcción del objeto.

```
public class Estudiante
{
    public int Id { get; init; } // = 101;
    //también se puede asignar valor aquí mismo
    //otra forma de asignar valor es mediante el constructor
}
public partial class Program
{
    public static void Main()
    {
        Estudiante est = new Estudiante()
        {
            Id = 101
        };
        //est.Id = 101;
        //Error:solo se le puede asignar valor al crear el objeto
        Console.WriteLine($"El id del estudiante es: {est.Id}");
    }
}
```

**Nota:** Para ver mas información detallada a cerca de **init** pueden revisar el link: <https://docs.microsoft.com/es-mx/dotnet/csharp/language-reference/proposals/csharp-9.0/init>

# Recursos adicionales

- ▶ PRAGIM Pagina principal
  - ▶ <http://www.PragimTech.com>



[27]

Propiedades