

www.PragimTech.com

Training +Placements = Our success

Pragim@PragimTech.com

► **Parte 21 – Herencia en C#**

Cristina Carrasco

PRAGIM Technologies

Cristina@cristinacarrasco.com

En esta lección veremos:

- ▶ ¿Porqué usar herencia?
- ▶ Ventajas de la herencia
- ▶ Sintaxis de la herencia
- ▶ Conceptos de herencia

www.PragimTech.com

Training +Placements = Our success
Pragim@PragimTech.com

¿Porqué usar herencia?

```
public class EmpleadoDeTiempoCompleto
{
    string Nombre;
    string Apellido;
    string Email;
    float SalarioAnual;

    public void ImprimirNombreCompleto()
    {

    }
}
```

```
public class EmpleadoDeTiempoParcial
{
    string Nombre;
    string Apellido;
    string Email;
    float PagoPorHora;

    public void ImprimirNombreCompleto()
    {

    }
}
```

Mucho código entre estas dos clases esta duplicado

Usando herencia

```
public class Empleado
{
    string Nombre;
    string Apellido;
    string Email;

    public void ImprimirNombreCompleto()
    {

    }
}
```

Mover todo el código común en la clase base Empleado

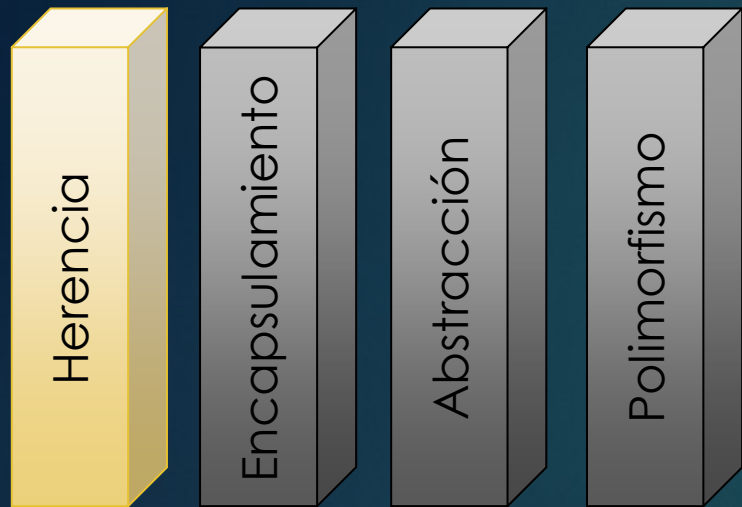
```
public class EmpleadoDeTiempoCompleto
{
    float SalarioAnual;
}
```

```
public class EmpleadoDeTiempoParcial
{
    float PagoPorHora;
}
```

El código específico para empleado de tiempo completo y de tiempo parcial queda en su respectiva clase derivada

¿Porqué usar herencia?

Los 4 pilares de la programación orientada a objetos



- ▶ Herencia es uno de los pilares de la programación orientada a objetos.
- ▶ Nos permite reutilizar código.
- ▶ Reutilizar código reduce el tiempo de programación y los posibles errores.

Nota: Todos los campos comunes, propiedades, métodos se especifican en la clase base y esto nos permite la reutilización del código. En las clases derivadas solo se especifican los campos, métodos, propiedades, únicos y específicos de esta clase.

Sintaxis de la herencia

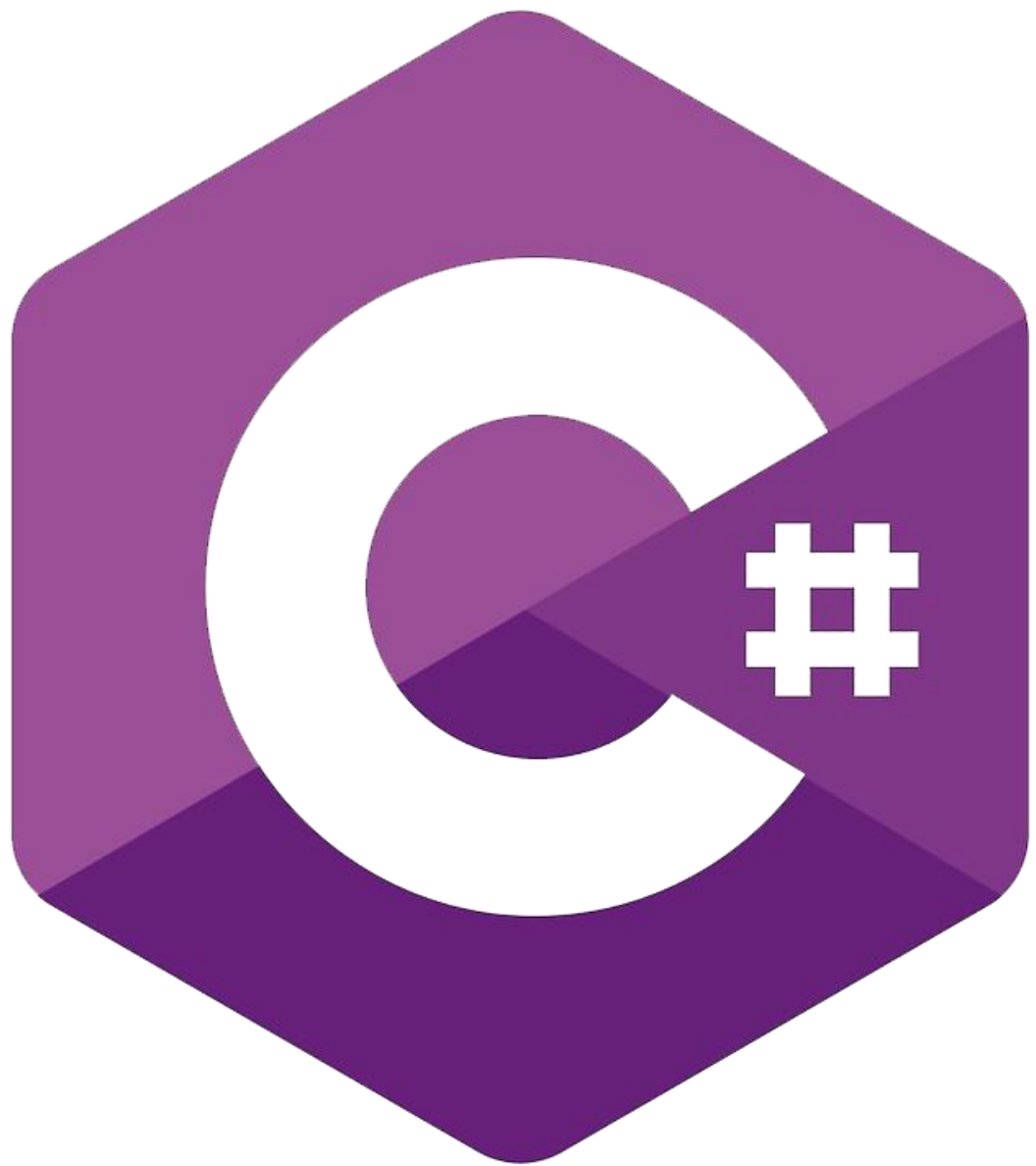
```
public class ClaseBase
{
    // Implementacion para la clase base
}

public class ClaseDerivada : ClaseBase
{
    // Implementacion para la clase derivada
}
```

- ▶ En este ejemplo **ClaseDerivada** hereda de **ClaseBase**.
- ▶ C# soporta herencia de solo una clase.
- ▶ C# soporta herencia de múltiples interfaces.
- ▶ La clase derivada es una especialización de la clase base.
- ▶ La clase base es automáticamente instanciada antes que la clase derivada.
- ▶ El constructor de la clase base es ejecutado antes que el constructor de la clase derivada.

Recursos adicionales

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



[21]

Herencia