

REFACTORIZACIÓN

En la primera refactorización lo que se ha hecho es cambiar el nombre de las variables por un nombre que identificativo.

ANTES

```
4
5 namespace Fecha
6 {
7     10 referencias | 0 cambios | 0 autores, 0 cambios
8     class Fecha
9     {
10         public int d;
11         public int m;
12         public int a;
13         //TODO validar los valores introducidos
14         /// <summary>
15         /// Constructor de Fecha sin parámetros
16         /// Se establecen los valores a 1
17         /// </summary>
18         1 referencia | 0 cambios | 0 autores, 0 cambios
19         public Fecha()
20         {
21             d = 1;
22             m = 1;
23             a = 1;
24         }
25         /// <summary>
26         /// Constructor de Fecha con 3 parámetros
```

DESPUES

```
{
27     10 referencias | 0 cambios | 0 autores, 0 cambios
28     class Fecha
29     {
30         public int dia;
31         public int mes;
32         public int año;
33         //TODO validar los valores introducidos
34         /// <summary>
35         /// Constructor de Fecha sin parámetros
36         /// Se establecen los valores a 1
37         /// </summary>
38         1 referencia | 0 cambios | 0 autores, 0 cambios
39         public Fecha()
40         {
41             dia = 1;
42             mes = 1;
43             año = 1;
44         }
45         /// <summary>
46         /// Constructor de Fecha con 3 parámetros
```

En la segunda lo que hemos hecho es encapsular las variables para que no sean accesibles para todo el mundo.

ANTES

```
namespace Fecha
{
    10 referencias | 0 cambios | 0 autores, 0 cambios
    class Fecha
    {
        public int dia;
        public int mes;
        public int año;
        //TODO validar los valores introd
        /// <summary>
        /// Constructor de Fecha sin pará
        /// Se establecen los valores a 1
        /// </summary>
        1 referencia | 0 cambios | 0 autores, 0 cambios
    }
```

DESPUES

```
10 referencias | 0 cambios | 0 autores, 0 cambios
class Fecha
{
    private int dia;
    private int mes;
    private int año;

    3 referencias | 0 cambios | 0 autores, 0 cambios
    public int Dia { get => dia; set => dia = value; }
    3 referencias | 0 cambios | 0 autores, 0 cambios
    public int Mes { get => mes; set => mes = value; }
    6 referencias | 0 cambios | 0 autores, 0 cambios
    public int Año { get => año; set => año = value; }

    //TODO validar los valores introducidos
    /// <summary>
    /// Constructor de Fecha sin parámetros
    /// Se establecen los valores a 1
    /// </summary>
}
```

El tercero lo que hemos hecho ha sido ordenar el switch para que en un futuro sea más fácil localizar las diferentes posibilidades.

ANTES

```
int diasMes = 0;
switch (mes)
{
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        diasMes = 31;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        diasMes = 30;
        break;
    case 2: // verificación de año bisiesto
        if (bisiesto)
            diasMes = 29;
        else
            diasMes = 28;
        break;
}
```

DESPUES

```
int diasMes = 0;
switch (mes)
{
    case 1:
    case 2: // verificación de año bisiesto
        if (bisiesto)
            diasMes = 29;
        else
            diasMes = 28;
        break;
    case 3:
    case 4:
    case 5:
    case 6:
    case 7:
    case 8:
    case 9:
    case 10:
    case 11:
        diasMes = 30;
        break;
    case 12:
        diasMes = 31;
        break;
}
if (dia >= 1 && dia <= diasMes)
    this.Dia = dia;
else
    this.Dia = 1;
```

La siguiente se ha remplazado los datos de la clase Fecha por un objeto para que haya una clase donde almacenar las variables y las propiedades y quitar cosas de Fecha.

ANTES

```
class Fecha
{
    private int dia;
    private int mes;
    private int año;

    3 referencias | 0 cambios | 0 autores, 0 cambios
    public int Dia { get => dia; set => dia = value; }
    3 referencias | 0 cambios | 0 autores, 0 cambios
    public int Mes { get => mes; set => mes = value; }
    6 referencias | 0 cambios | 0 autores, 0 cambios
    public int Año { get => año; set => año = value; }

    //TODO validar los valores introducidos
    /// <summary>
    /// Constructor de Fecha sin parámetros
}
```

DESPUES

```
] namespace Fecha
{
    1 referencia | 0 cambios | 0 autores, 0 cambios
    class Años
    {
        private int dia;
        private int mes;
        private int año;

        4 referencias | 0 cambios | 0 autores, 0 cambios
        public int Dia { get => dia; set => dia = value; }
        4 referencias | 0 cambios | 0 autores, 0 cambios
        public int Mes { get => mes; set => mes = value; }
        10 referencias | 0 cambios | 0 autores, 0 cambios
        public int Año { get => año; set => año = value; }
    }
}
```

Como queda la clase Fecha después de la esta refactorización.

```
public Años fechas;
//TODO validar los valores introducidos
/// <summary>
/// Constructor de Fecha sin parámetros
/// Se establecen los valores a 1
/// </summary>
public Fecha()
{
    fechas.Año = 1;
    fechas.Mes = 1;
    fechas.Dia = 1;
}
```

Y la última que se ha realizado es quitar código repetido ya que hacemos la misma comprobación para dos variables diferentes. De esta manera comprobamos solo una variable con un solo if.

ANTES

```
//Fecha correcta, año bisiesto
Fecha fecha1 = new Fecha(12, 2012, 4, false);
Console.WriteLine("Fecha 1: " + fecha1.ToString());
if (fecha1.EsBisiesto())
    Console.WriteLine("El año " + fecha1.año + " es bisiesto");
else
    Console.WriteLine("El año " + fecha1.año + " no es bisiesto");
//Fecha correcta, año no bisiesto
Fecha fecha2 = new Fecha(10, 2013, 4, false);
Console.WriteLine("Fecha 2: " + fecha2.ToString());
if (fecha2.EsBisiesto())
    Console.WriteLine("El año " + fecha2.año + " es bisiesto");
else
    Console.WriteLine("El año " + fecha2.año + " no es bisiesto");
//Fecha con valores incorrectos
Fecha fecha3 = new Fecha(13, -4, 4, false);
Console.WriteLine("Fecha 3: " + fecha3.ToString());
//Fecha con asignación incorrecta de valores erroneos
Fecha fecha4 = new Fecha();
fecha4.día = 67;
fecha4.mes = 80;
fecha4.año = 3678;
Console.WriteLine("Fecha 4: " + fecha4.ToString());
```

DESPUES

```
Console.WriteLine("Fecha 1: " + fecha1.ToString());
//MAL!!
Fecha Fechas = fecha1;
if (Fechas.EsBisiesto())
    Console.WriteLine("El año " + Fechas.año + " es bisiesto");
else
    Console.WriteLine("El año " + Fechas.año + " no es bisiesto");
```