9.2. Más posibilidades de la "consola"

En "Console" hay mucho más que ReadLine y WriteLine, aunque no todas las posibilidades están incluidas en las primera versión de la "plataforma punto net", sino a partir de la versión 2 (de enero de 2006). Vamos a ver algunas de las funcionalidades adicionales que nos pueden resultar más útiles:

- Clear(): borra la pantalla.
- ForegroundColor: cambia el color de primer plano (para indicar los colores, hay definidas constantes como "ConsoleColor.Black", que se detallan al final de este apartado).
- BackgroundColor: cambia el color de fondo (para el texto que se escriba a partir de entonces; si se quiere borrar la pantalla con un cierto color, se deberá primero cambiar el color de fondo y después usar "Clear").
- SetCursorPosition(x, y): cambia la posición del cursor ("x" se empieza a contar desde el margen izquierdo, empezando en 0, e "y" desde la parte superior de la pantalla, también comenzando desde 0).
- Readkey(interceptar): lee una tecla desde teclado. El parámetro "interceptar" es un "bool", y es opcional. Indica si se debe capturar la tecla sin permitir que se vea en pantalla ("true" para que no se vea, "false" para que se pueda ver). Si no se indica este parámetro, la tecla pulsada se muestra en pantalla.
- KeyAvailable: indica si hay alguna tecla disponible para ser leída (es un "bool"). Permite no bloquear un programa hasta que se pulse una tecla, que es lo que ocurriría si se llama directamente a Readkey.
- Title: el título que se va a mostrar en la consola (es un "string")
- Como el tamaño de la ventana de consola no necesariamente será algo prefijado en el sistema, se puede saber o fijar con WindowHeight y WindowWidth. Además, existe un "buffer" que puede ser mayor que la zona visible de la pantalla (y en ese caso, se podría hacer scroll para ver el resto). Su tamaño se podría conocer o cambiar con BufferHeight y BufferWidth. Tanto el ancho como el alto de ventana y de buffer se pueden fijar a la vez con SetWindowSize(w,h) y con SetBufferSize(w,h).

```
// Ejemplo_09_02a.cs
// Más posibilidades de "System.Console"
// Introducción a C#, por Nacho Cabanes

using System;

public class Ejemplo_09_02a
{
    public static void Main()
```

```
    {
        int posX, posY;

        Console.Title = "Ejemplo de consola";
        Console.SetWindowSize (80, 25);
        Console.BackgroundColor = ConsoleColor.Green;
        Console.ForegroundColor = ConsoleColor.Black;
        Console.Clear();

        posY = 10;  // En la fila 10
        Random r = new Random(DateTime.Now.Millisecond);
        posX = r.Next(20, 40); // Columna al azar entre 20 y 40
        Console.SetCursorPosition(posX, posY);
        Console.WriteLine("Bienvenido");

        Console.ForegroundColor = ConsoleColor.Blue;
        Console.SetCursorPosition(10, 15);
        Console.Write("Pulsa 1 o 2: ");
        ConsoleKeyInfo tecla;
        do
        {
            tecla = Console.ReadKey(false);
        }
        while ((tecla.KeyChar != '1') && (tecla.KeyChar != '2'));

        int maxY = Console.WindowHeight;
        int maxX = Console.WindowWidth;
        Console.SetCursorPosition(maxX-50, maxY-1);
        Console.ForegroundColor = ConsoleColor.Red;
        Console.Write("Pulsa una tecla para terminar... ");
        Console.ReadKey(true);
    }
}
```

Para comprobar el valor de una **tecla**, como se ve en el ejemplo anterior, tenemos que usar una variable de tipo "ConsoleKeyInfo" (información de tecla de consola). Un ConsoleKeyInfo tiene campos como:

- KeyChar, que representa el carácter que se escribiría al pulsar esa tecla. Por ejemplo, podríamos hacer `if (tecla.KeyChar == '1') ...`
- Key, que se refiere al código de la tecla (porque hay teclas que no tienen un carácter visualizable, como F1 o las teclas de movimiento del cursor). Por ejemplo, para comprobar la tecla ESC podríamos hacer `if (tecla.Key == ConsoleKey.Escape) ...` . Algunos de los códigos de tecla disponibles son:
    - Teclas de edición y control como, como: Backspace (Tecla RETROCESO), Tab (Tecla del tabulador), Enter (Tecla Intro), Pause (Tecla PAUSA), Escape (Tecla ESC), Spacebar (Tecla BARRA ESPACIADORA), PrintScreen (Tecla IMPR PANT), Insert (Tecla INS (INSERT)), Delete (Tecla SUPR (SUPRIMIR))
    - Teclas de movimiento del cursor, como: PageUp (Tecla RE PÁG), PageDown (Tecla AV PÁG), End (Tecla FIN), Home (Tecla INICIO),

LeftArrow (Tecla FLECHA IZQUIERDA), UpArrow (Tecla FLECHA ARRIBA), RightArrow (Tecla FLECHA DERECHA), DownArrow (Tecla FLECHA ABAJO)

- Teclas alfabéticas, como: A (Tecla A), B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
- Teclas numéricas, como: D0 (Tecla 0), D1, D2, D3, D4, D5, D6, D7, D8, D9
- Teclado numérico adicional: NumPad0 (Tecla 0 del teclado numérico), NumPad1, NumPad2, NumPad3, NumPad4, NumPad5, NumPad6, NumPad7, NumPad8, NumPad9, Multiply (Tecla Multiplicación), Add (Tecla Suma), Separator (Tecla Separador de miles, si la hubiera), Subtract (Tecla Resta), Decimal (Tecla Decimal), Divide (Tecla División)
- Teclas de función: F1, F2 y sucesivas (hasta F24)
- Teclas especiales de Windows: LeftWindows (Tecla izquierda con el logotipo de Windows), RightWindows (Tecla derecha con el logotipo de Windows, si existe)
- Incluso teclas multimedia, si el teclado las incorpora, como: VolumeMute (Tecla Silenciar el volumen, enMicrosoft Natural Keyboard, bajo Windows 2000 o posterior), VolumeDown (Bajar el volumen, ídem), VolumeUp (Subir el volumen), MediaNext (Siguiente pista de multimedia), etc.
- Modifiers, que permite comprobar si se han pulsado simultáneamente teclas modificadoras: Alt, Shift o Control. Se usaría con el operador binario "and", como en este ejemplo:

```
if ((tecla.Modifiers & ConsoleModifiers.Alt) != 0)
    Console.Write("Has pulsado Alt");
```

Como ya hemos anticipado, Console.Readkey hace que el programa se quede **parado** hasta que se pulse una tecla. Si queremos hacer algo mientras que el usuario no pulse ninguna tecla, podemos emplear Console.KeyAvailable para comprobar si ya se ha pulsado alguna tecla que haya que analizar, como en este ejemplo, que permite mover un símbolo a izquierda y derecha, pero muestra una animación simple si no pulsamos ninguna tecla:

```
// Ejemplo_09_02b.cs
// No bloquear el programa con Console.Readkey
// Introducción a C#, por Nacho Cabanes

using System;
using System.Threading;
```

```
public class Ejemplo_09_02b
{
    public static void Main()
    {
        int posX=40, posY=10;
        string simbolos = "^>v<";
        byte simboloActual = 0;
        bool terminado = false;

        do
        {
            Console.Clear();
            Console.SetCursorPosition(posX, posY);
            Console.Write( simbolos[ simboloActual ]);
            Thread.Sleep(500);
            if (Console.KeyAvailable)
            {
                ConsoleKeyInfo tecla = Console.ReadKey(true);
                if (tecla.Key == ConsoleKey.RightArrow) posX++;
                if (tecla.Key == ConsoleKey.LeftArrow) posX--;
                if (tecla.Key == ConsoleKey.Escape) terminado = true;
            }
            simboloActual++;
            if (simboloActual > 3) simboloActual = 0;
        }
        while ( ! terminado );
    }
}
```

Al igual que en este ejemplo, será recomendable hacer una pequeña **pausa** entre una comprobación de teclas y la siguiente, con Thread.Sleep, tanto para que la animación no sea demasiado rápida como para no hacer un consumo muy alto de procesador para tareas poco importantes.

Los **colores** que tenemos disponibles (y que se deben escribir precedidos con "ConsoleColor") son: Black (negro), DarkBlue (azul marino), DarkGreen (verde oscuro), DarkCyan (verde azulado oscuro), DarkRed (rojo oscuro), DarkMagenta (fucsia oscuro o púrpura), DarkYellow (amarillo oscuro u ocre), Gray (gris), DarkGray (gris oscuro), Blue (azul), Green (verde), Cyan (aguamarina o verde azulado claro), Red (rojo), Magenta (fucsia), Yellow (amarillo), White (blanco).

Ejercicios propuestos:

(9.2.1) Crea un programa que muestre una "pelota" (la letra "O") rebotando en los bordes de la pantalla. Para que no se mueva demasiado rápido, haz una pausa de 50 milisegundos entre un "fotograma" de la animación y otro.

(9.2.2) Crea una versión de la "base de datos de ficheros" (ejemplo 04_06a) que use colores para ayudar a distinguir los mensajes del programa de las respuestas del usuario, y que no necesite pulsar Intro tras escoger cada opción.

(9.2.3) Crea un programa que permita "dibujar" en consola, moviendo el cursor con las flechas del teclado y pulsando "espacio" para dibujar un punto o borrarlo.
(9.2.4) Crea una versión del programa de "dibujar" en consola (9.2.3), que permita escribir más caracteres (por ejemplo, las letras), así como mostrar ayuda (pulsando F1), guardar el contenido de la pantalla en un fichero de texto (con F2) o recuperarlo (con F3).
(9.2.5) Crea una versión mejorada del programa 9.1.2 (mostrar el reloj actualizado en pantalla, que lo dibuje siempre en la esquina superior derecha de la pantalla).