

Practica depuración 3

Entornos de desarrollo

Emiliano Montesdeoca | 1 DAWB

# Índice

1. Enunciado
2. Objetivo
3. Código
4. Funcionamiento
5. Depuración línea a línea
6. Depuración función a función
7. Cambio de variables en caliente
8. Punto de interrupción

# Enunciado

Con el programa hecho en C# Diamanana2.cs realizar la depuración del código.

1. Primero probar una ejecución normal-
2. Después mediante F10 y F11 probamos la depuración línea a línea y por procedimiento.
3. En la Ventana de Locales vemos las variables y los contenidos que van teniendo.
4. En la ventana de pila vemos el paso por las funciones.
5. Poner un punto de interrupción y continuar la ejecución desde ahí. Comprobar que entra o no en las funciones según si usamos F10 o F11.

# Objetivo

El objetivo de esta práctica es demostrar cómo podemos realizar una depuración de un programa línea a línea y función a función. La depuración sirve para poder encontrar errores en el programa, o realizar pruebas que fuercen el programa a errar para poder controlar esos errores.

# Codigo

static void Main(string[] args)

{

int dd, mm, aa;

int dds, mms, aas;

int[] tabla = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

Console.Write("Introduzca Día: ");

dd = Convert.ToInt32(Console.ReadLine());

Console.Write("Introduzca Mes: ");

mm = Int32.Parse(Console.ReadLine());

Console.Write("Introduzca Año: ");

aa = Convert.ToInt32(Console.ReadLine());

if ((aa % 4 == 0) && (aa % 100 != 0) || (aa % 400 == 0))

tabla[1] = 29;

if (comprueba(dd, mm, aa, tabla) == false)

{

Console.WriteLine("Fecha errónea");

Console.ReadLine();

return;

}

dds = dd + 1;

mms = mm;

aas = aa;

if (dds > tabla[mm - 1])

{

dds = 1;

if (mms == 12)

{

mms = 1;

aas++;

}

else

mms++;

}

Console.WriteLine("\n\n\n");

Console.Write("Fecha: ");

Console.Write("{0} - {1} - {2}", dds, mms, aas);

Console.ReadLine();

}

static bool comprueba(int dd, int mm, int aa, int[] tabla)

{

bool valor = true;

if (mm < 1 || mm > 12)

valor = false;

else

if (dd < 1 || dd > tabla[mm - 1])

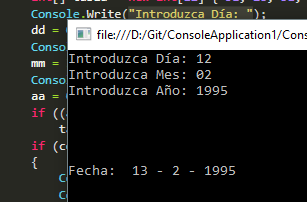
valor = false;

return (valor);

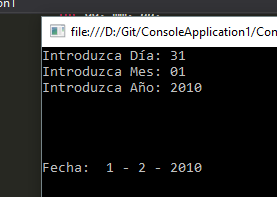
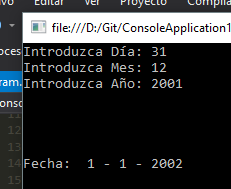
}

# Funcionamiento

El funcionamiento del código es que, dándole unos valores que son el día, el mes y el año, el programa se encarga de devolver el día siguiente:

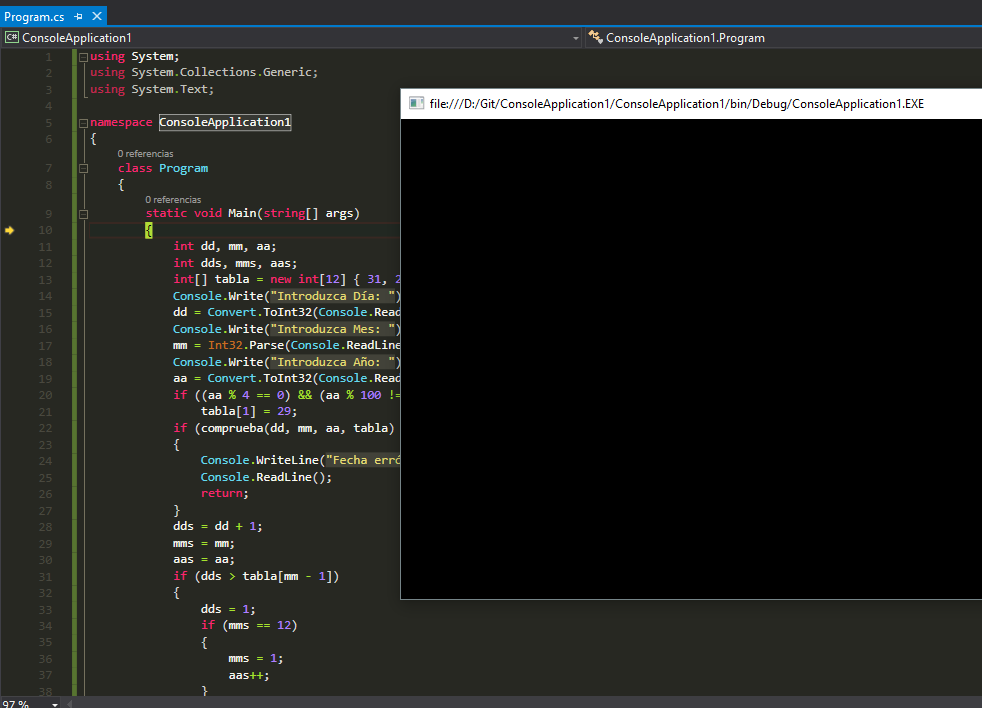


También se encarga de devolver el día siguiente si hay cambio de día, de mes o de año:

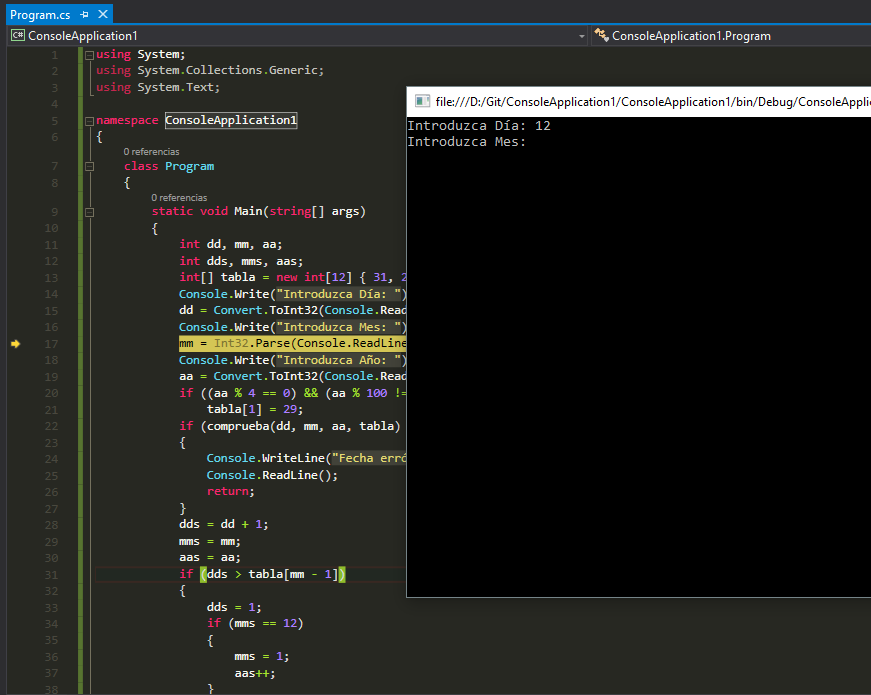
 

# Depuración línea a línea

Para activar la depuración línea a línea hay que apretar la tecla F11, el programa se ejecutara, pero no podremos ver nada ya que no ha ejecutado ninguna línea todavía.

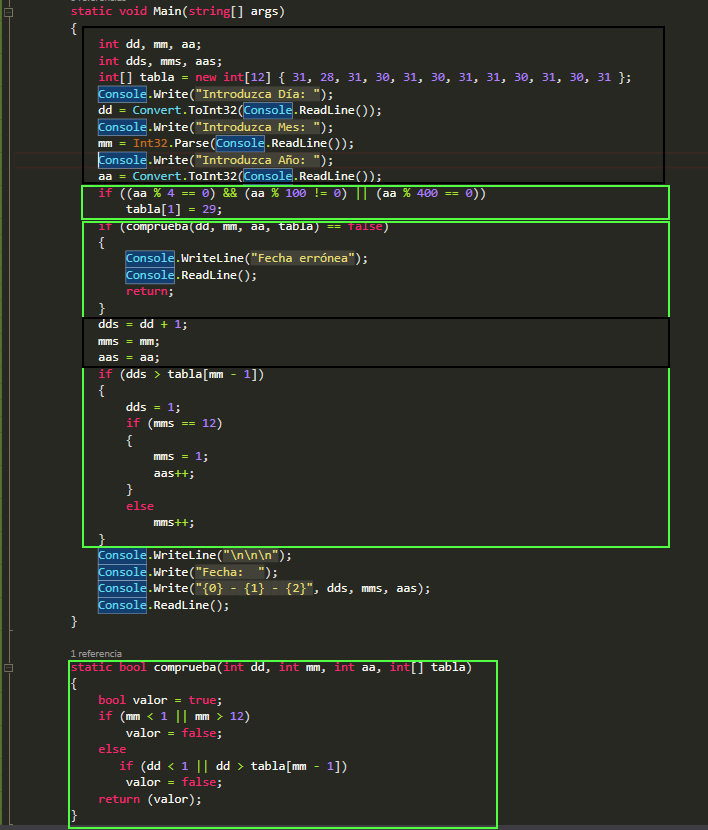


Si vamos adelantando con el F11 podremos ir ejecutando línea a línea:



# Depuración función a función

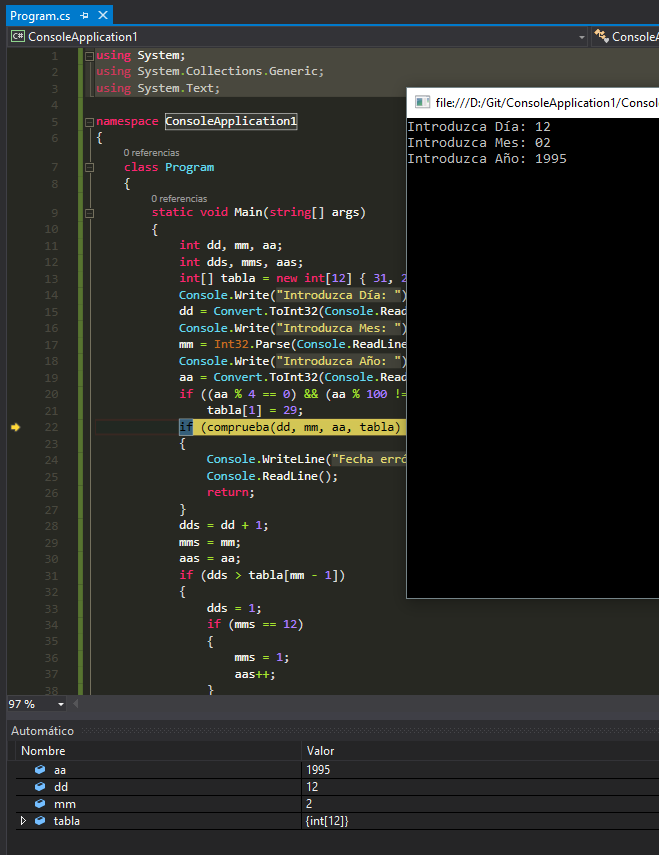
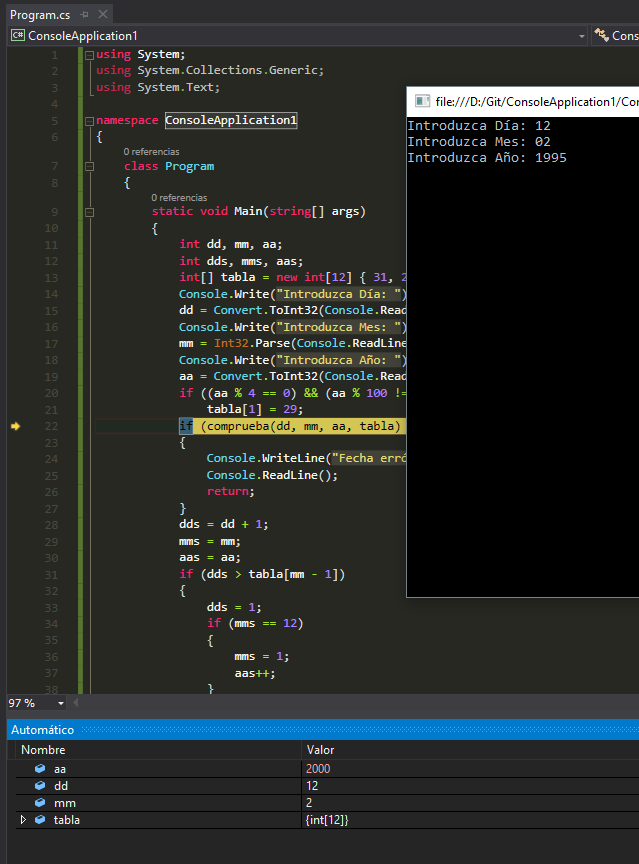
La forma en la que la depuracion funciona a funcion trabaja es diferente, ejecuta linea a linea pero cuando tiene una funcion que realizar la ejecuta sin entrar en ella:



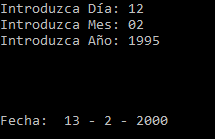
En esta imagen podemos ver cómo funciona la depuración función a función. El código en el rectángulo negro se depura como si fuera línea a línea, pero el que está dentro del rectángulo verde lo ejecuta directamente sin hacerlo línea a línea.

# Cambio de variables en caliente

La mejor ventaja de la depuración es la capacidad de poder cambiar las variables introducidas más delante en el programa. Para poder hacerlo hay que ir a la ventana de variables que se abre cuando hacemos la depuración:

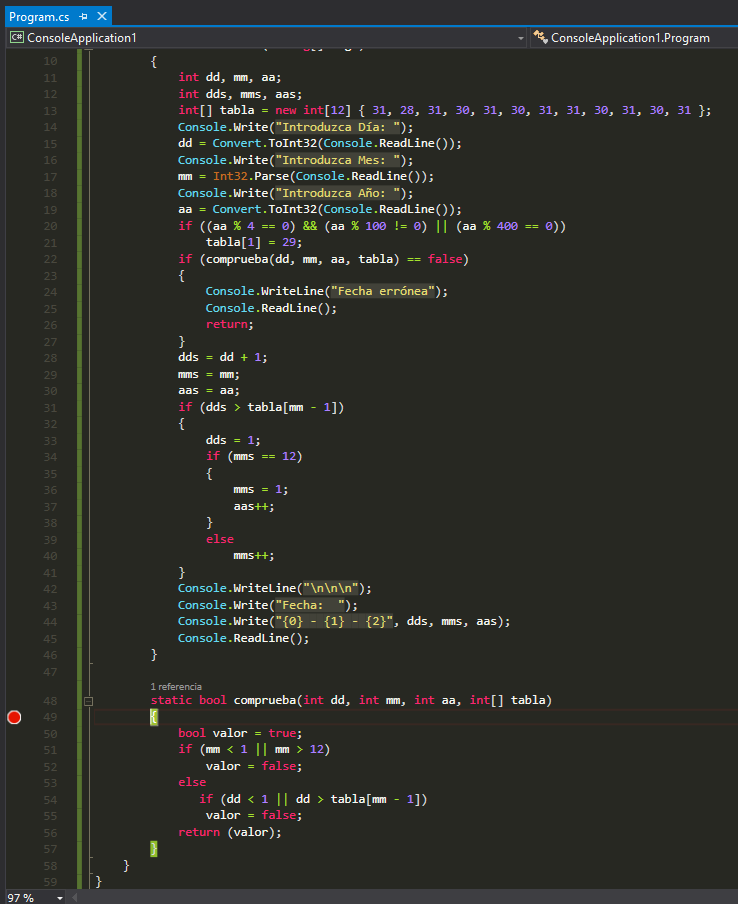
Ahí podemos ver como la variable aa paso de ser 1995 a 2000, y eso afectara al resultado final:



# Punto de interrupción

El punto de interrupción sirve cuando queremos parar en determinada parte del código, y no tener que ejecutar línea a línea o funciona a función todo el programa. Esto sirve más que nada para buscar errores en determinado punto.

Para poder agregar un punto de interrupción tenemos que dar clic izquierdo a la izquierda de la línea de código, y se activara el punto de interrupción:



En este caso cuando el programa llegue a la función comprueba se parar y seguirá línea a línea o función a función.