

## EJERCICIOS MÓDULO 4 Excepciones

### Objetivos

Al final de esta práctica, usted será capaz de:

- Lanzar y capturar excepciones.
- Visualizar mensajes de error.

### Requisitos previos

Antes de realizar la práctica debe estar familiarizado con los siguientes temas:

- Creación de variables en C#
- Uso de operadores comunes en C#
- Creación de tipos enum en C#

### EJERCICIO 1

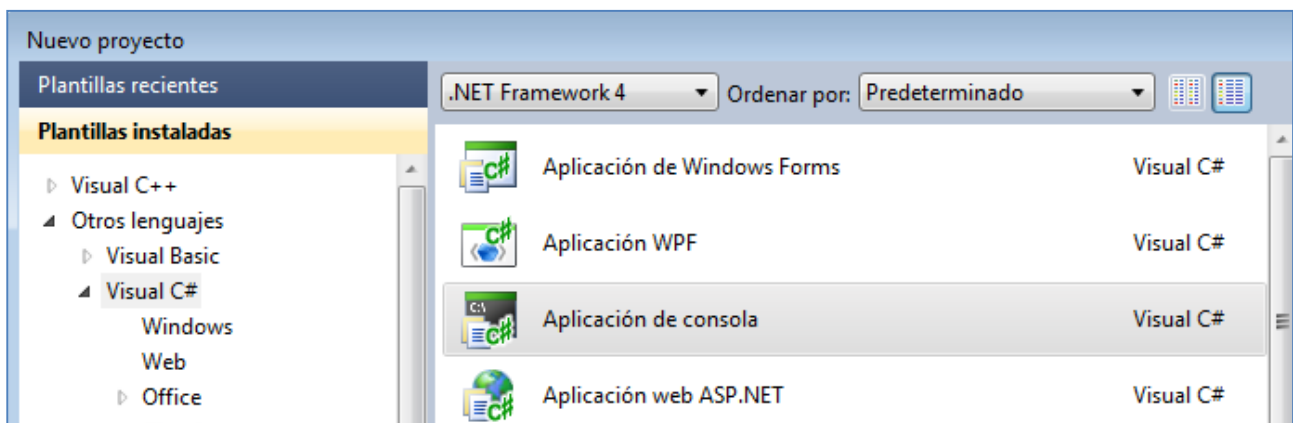
Adición de tratamiento de excepciones a un programa C#.

En este ejercicio escribirá un programa que utiliza tratamiento de excepciones para capturar errores inesperados en tiempo de ejecución. El programa pide al usuario dos valores enteros, divide el primero por el segundo y muestra el resultado.

**Paso 1.** Inicie Visual Studio .NET si aún no se está ejecutando.

**Paso 2.** En el menú File, señale New y pulse Project.

**Paso 3.** Pulse Visual C# Projects en el cuadro Plantillas instaladas y ConsoleApplication.



**Paso 4.** Escriba Divisor en el cuadro Name y la ubicación deseada para el proyecto en el cuadro Location (Ubicación). Pulse OK.

Nombre:	Divisor
Ubicación:	c:\users\antonio\documents\visual studio 2010\Projects
Solución:	Crear nueva solución
Nombre de la solución:	Divisor

**Paso 5.** En el método Main, escriba una instrucción que pida al usuario el primer entero.

```
Console.Write("Introduce primer numero:");
```

**Paso 6.** Escriba otra instrucción que lea la respuesta del usuario desde el teclado y la asigne a una variable llamada temp de tipo string.

```
string temp = Console.ReadLine();
```

**Paso 7.** Añada la siguiente instrucción para convertir el valor de la cadena de temp en un entero y almacenar el resultado en i:

```
int i = Int32.Parse(temp);
```

**Paso 8.** Añada las instrucciones para pedir al usuario el segundo entero, leer la respuesta del usuario desde el teclado, asignarla a temp y convertir el valor de temp en un entero, almacenando el resultado en j.

```
Console.Write("Introduce el segundo entero:");
```

```
temp = Console.ReadLine();
```

```
int j = Int32.Parse(temp);
```

**Paso 9.** Cree una nueva variable entera k que reciba el valor resultante de dividir i entre j, e insértelo al final del procedimiento anterior.

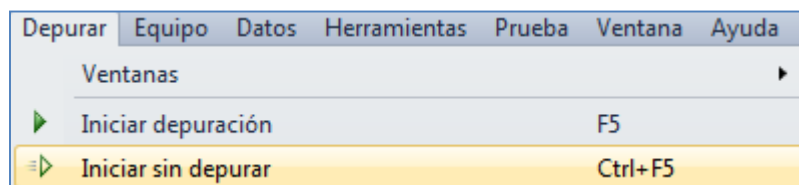
```
int k = i / j;
```

**Paso 10.** Añada una instrucción para mostrar el valor de k. Y otra más para parar la ejecución del programa y poder ver el resultado antes de hacer ENTER.

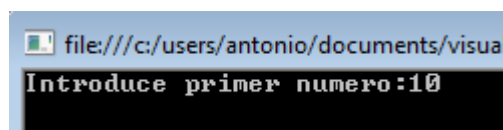
```
Console.Write("Valor division " + k);
```

```
Console.ReadLine();
```

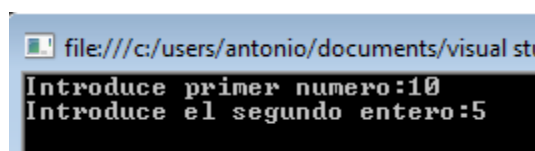
**Paso 11.** Para probar el programa, en el menú Debug, seleccione StartWithoutDebugging (o pulse CTRL+F5).



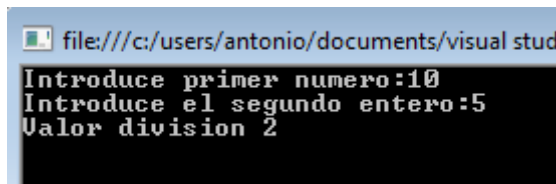
**Paso 12.** Escriba 10 como valor del primer entero y pulse INTRO.



**Paso 13.** Escriba 5 como valor del segundo entero y pulse INTRO.

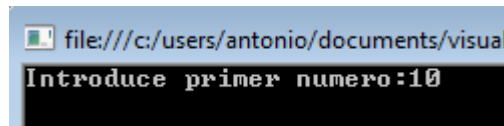


**Paso 14.** Compruebe que el valor de k que aparece es 2.



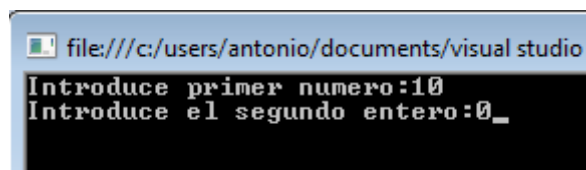
```
file:///c:/users/antonio/documents/visual stud
Introduce primer numero:10
Introduce el segundo entero:5
Valor division 2
```

**Paso 15.** Pulse CTRL+F5 para volver a ejecutar el programa. Escriba 10 como valor del primer entero y pulse INTRO.



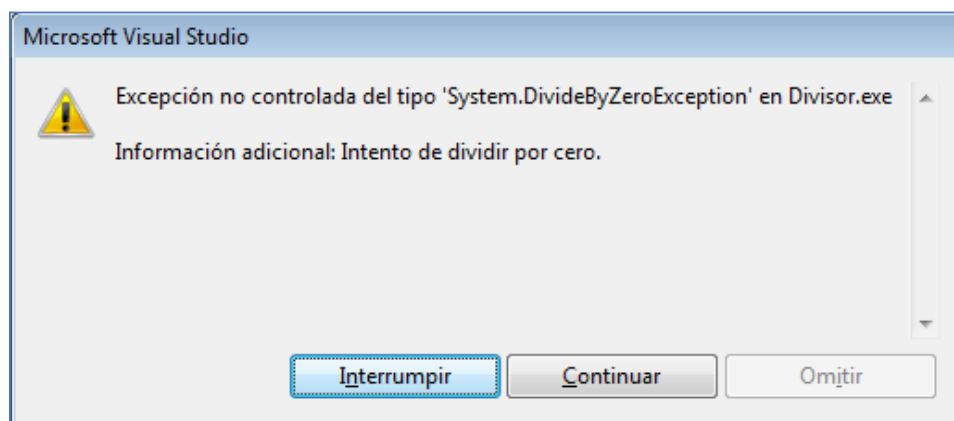
```
file:///c:/users/antonio/documents/visual
Introduce primer numero:10
```

**Paso 16.** Escriba 0 como valor del segundo entero y pulse INTRO.



```
file:///c:/users/antonio/documents/visual studio 2
Introduce primer numero:10
Introduce el segundo entero:0_
```

**Paso 17.** El programa provoca el lanzamiento de una excepción (división por cero).



**Paso 18.** Para añadir tratamiento de excepciones al programa, ponga el código del método Main dentro de un bloque try, como se indica a continuación:

```
try {
    int k = i / j;
    Console.WriteLine("Valor division " + k);
}
```

**Paso 19.** Añada a Main una instrucción catch después del bloque try. La instrucción catch tiene que imprimir un mensaje corto, como se ve en este código:

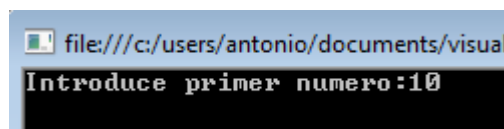
```
catch (Exception e) {
    Console.WriteLine("Excepción lanzada: {0}", e);
}
```

**Paso 20.** El método Main completo será similar al siguiente:

```
namespace Divisor
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Introduce primer numero:");
            string temp = Console.ReadLine();
            int i = Int32.Parse(temp);
            Console.Write("Introduce el segundo entero:");
            temp = Console.ReadLine();
            int j = Int32.Parse(temp);
            try {
                int k = i / j;
                Console.WriteLine("Valor division " + k);
            }
            catch (Exception e) {
                Console.WriteLine("Excepción lanzada: {0}", e);
            }
            Console.ReadLine();
        }
    }
}
```

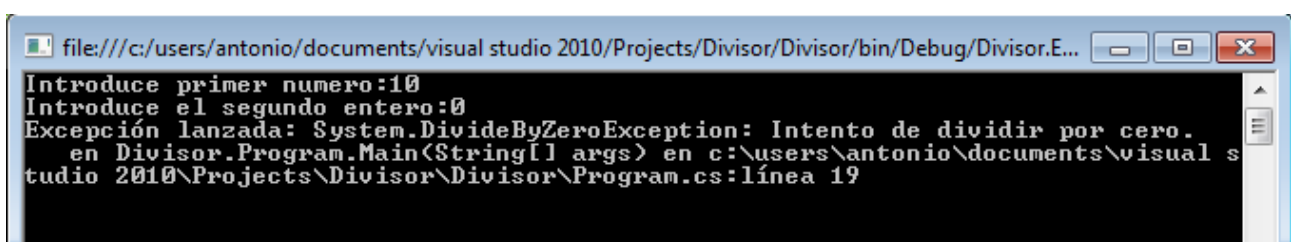
**Paso 21.** Para probar el código para tratamiento de excepciones, pulse CTRL+F5 para volver a ejecutar el programa.

**Paso 22.** Escriba 10 como valor del primer entero y pulse INTRO.



file:///c:/users/antonio/documents/visual  
Introduce primer numero:10

**Paso 23.** Escriba 0 como valor del segundo entero y pulse INTRO.



file:///c:/users/antonio/documents/visual studio 2010/Projects/Divisor/Divisor/bin/Debug/Divisor.E...  
Introduce primer numero:10  
Introduce el segundo entero:0  
Excepción lanzada: System.DivideByZeroException: Intento de dividir por cero.  
en Divisor.Program.Main(String[] args) en c:\users\antonio\documents\visual studio 2010\Projects\Divisor\Divisor\Program.cs:línea 19

El programa sigue provocando el lanzamiento de una excepción (división por cero), pero esta vez el error es capturado y aparece su mensaje.

## EJERCICIO 2

Validación del número de día.

En este ejercicio añadirá funcionalidad al programa creado en el Módulo 2. El programa examinará el número de día inicial introducido por el usuario. Si es menor que 1 o mayor que 365, el programa lanzará una excepción `InvalidArgument` ("Día fuera de intervalo"), capturará esta excepción en una cláusula `catch` y mostrará en la consola un mensaje de diagnóstico.

**Paso 1.** Abra el proyecto `QueDia` del módulo anterior.

**Paso 2.** Incluya todo el contenido del programa principal en un bloque `try`.

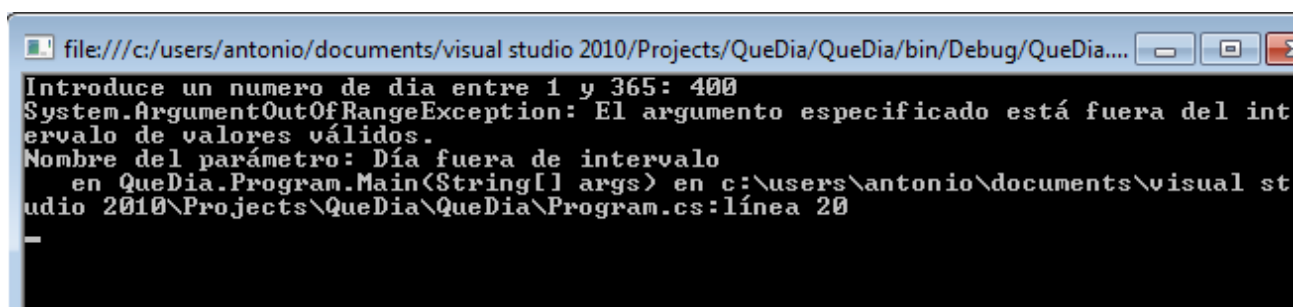
**Paso 3.** Después del bloque `try`, añada una cláusula `catch` que capture excepciones de tipo `System.Exception` y las llame capturada. Añada en el bloque `catch` una instrucción `WriteLine` que escriba en la consola la excepción capturada.

```
try {  
    ...  
} catch (Exception caught) {  
    Console.WriteLine(caught);  
    Console.Read();  
}
```

**Paso 4.** Añada una instrucción `if` después de la declaración de la variable `diaNum`. La instrucción `if` lanzará un objeto excepción `new` de tipo `System.ArgumentOutOfRangeException` si `diaNum` es menor que 1 o mayor que 365. Utilice el literal string "Día fuera de intervalo" para crear el objeto excepción.

```
if (diaNum < 1 || diaNum > 365) {  
    throw new ArgumentOutOfRangeException("Día fuera de intervalo");  
}
```

**Paso 5.** Compile el programa y corrija los posibles errores. Ejecute el programa e introduzca números de días menores de 1 y mayores de 365. Compruebe que la entrada no válida se captura de forma segura y que el objeto excepción es lanzado, capturado y mostrado.



```
namespace QueDia
{
    enum MesNombre {Enero, Febrero, Marzo, Abril, Mayo, Junio,
        Julio, Agosto, Septiembre, Octubre, Noviembre, Diciembre }

    class Program
    {
        static void Main(string[] args)
        {
            try {
                Console.Write("Introduce un numero de dia entre 1 y 365: ");
                string linea = Console.ReadLine();
                int diaNum = int.Parse(linea);
                if (diaNum < 1 || diaNum > 365)
                    throw new ArgumentOutOfRangeException("Día fuera de intervalo");
                int mesNum = 0;
                int[] DiasEnMeses = new int[] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
                foreach (int diasEnMes in DiasEnMeses) {
                    if (diaNum <= diasEnMes) {
                        break;
                    } else {
                        diaNum -= diasEnMes;
                        mesNum++;
                    }
                }
                string mesNombre;
                MesNombre temp = (MesNombre)mesNum;
                mesNombre = temp.ToString();
                Console.WriteLine("{0} {1}", diaNum, mesNombre);
                Console.Read();
            } catch (Exception caught) {
                Console.WriteLine(caught);
                Console.Read();
            }
        }
    }
}
```

### EJERCICIO 3

#### Tratamiento de años bisiestos

En este ejercicio añadirá funcionalidad al programa realizado en el Ejercicio 1. Una vez finalizado el ejercicio, el programa pedirá al usuario un año además de un número de día. El programa detectará si el año introducido es bisiesto. Comprobará si el número de día está entre 1 y 366 si el año es bisiesto, o entre 1 y 365 si no lo es. Finalmente, utilizará una nueva instrucción foreach para calcular correctamente el par de mes y día para años bisiestos.

**Paso 1.** Abra el proyecto QueDia del ejercicio anterior.

**Paso 2.** Añada al principio del try una instrucción que escriba en la consola un mensaje pidiendo al usuario que introduzca un año.

```
Console.WriteLine("Introduce un numero de año: ");
```

**Paso 3.** Añada una instrucción que declare una variable string llamada linea y la inicialice con una línea leída desde la consola con el método System.Console.ReadLine.

```
string linea = Console.ReadLine();
```

**Paso 4.** Añada una instrucción que declare una variable int llamada annoNum y la inicialice con el entero devuelto por el método int.Parse.

```
int annoNum = int.Parse(linea);
```

**Paso 5.** Inmediatamente después de la declaración de annoNum, añada una instrucción que declare una variable bool llamada esAnnoBisiesto. Inicialice esta variable con una expresión booleana que determine si annoNum es un año bisiesto. Un año es bisiesto si se cumplen las dos condiciones siguientes:

- Es divisible por 4.
- Es no divisible por 100, o es divisible por 400.

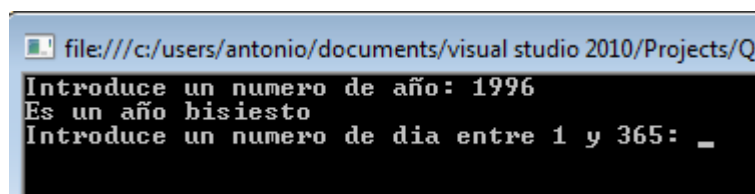
```
bool esAnnoBisiesto = (annoNum % 4 == 0) && (annoNum % 100 != 0 || annoNum % 400 == 0);
```

**Paso 6.** Añada una instrucción if inmediatamente después de la declaración de esAnnoBisiesto. Escriba en la instrucción if la cadena "ES un año bisiesto" o "NO es un año bisiesto" para la consola, dependiendo del valor de esAnnoBisiesto. Empleará esta instrucción if para comprobar que la determinación booleana de años bisiestos es correcta.

```
if (esAnnoBisiesto)
    Console.WriteLine("Es un año bisiesto");
else
    Console.WriteLine("NO es un año bisiesto ");
```

**Paso 7.** Compile el programa y corrija los posibles errores. Utilice la siguiente tabla para comprobar que la determinación booleana de años bisiestos es correcta.

Año bisiesto	Año no bisiesto
1996	1999
2000	1900
2004	2001



```
file:///c:/users/antonio/documents/visual studio 2010/Projects/Qu
Introduce un numero de año: 1996
Es un año bisiesto
Introduce un numero de día entre 1 y 365: _
```

```
file:///c:/users/antonio/documents/visual studio 2010/Projects/Que
Introduce un numero de año: 2001
NO es un año bisiesto
Introduce un numero de día entre 1 y 365:
```

**Paso 8.** Marque como comentario la instrucción `if` añadida anteriormente.

**Paso 9.** Inmediatamente después de la declaración de `esAnnoBisiesto`, añada una declaración de una variable `int` llamada `maxNumDias`. Inicialice `maxNumDias` con 366 o 365, dependiendo de si `esAnnoBisiesto` es `true` o `false`, respectivamente.

```
int maxNumDias = esAnnoBisiesto ? 366 : 365;
```

**Paso 10.** Modifique la instrucción `WriteLine` que pregunta al usuario el número de día. Tiene que mostrar el intervalo 1 a 366 si se ha introducido un año bisiesto y 1 a 365 si es un año no bisiesto.

```
Console.Write("Introduce un número de día entre 1 y {0}: ", maxNumDias);
```

**Paso 11.** Modifique la instrucción `if` que valida el valor de `diaNum` para usar la variable `maxNumDias` en lugar del literal 365.

```
if (diaNum < 1 || diaNum > maxNumDias) {
    throw new ArgumentOutOfRangeException("Día fuera de intervalo");
}
```

**Paso 12.** Compile el programa y corrija los posibles errores. Ejecute el programa y compruebe que el paso anterior funciona correctamente.

```
file:///C:/Users/Antonio/Documents/Visual Studio 2010/Projects/Que
Introduce un numero de año: 1996
Introduce un número de día entre 1 y 366:
```

```
namespace QueDia
{
    enum MesNombre {Enero, Febrero, Marzo, Abril, Mayo, Junio,
        Julio, Agosto, Septiembre, Octubre, Noviembre, Diciembre }

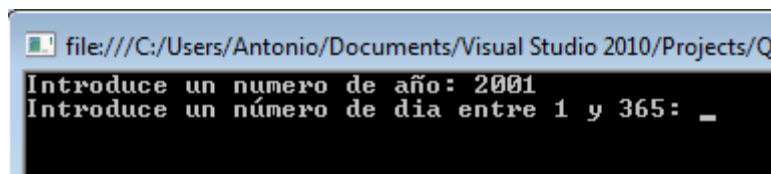
    class Program
    {
        static void Main(string[] args)
        {
            try {
                Console.Write("Introduce un numero de año: ");
                string linea = Console.ReadLine( );
                int annoNum = int.Parse(linea);
                bool esAnnoBisiesto = (annoNum % 4 == 0) && (annoNum % 100 != 0 || annoNum % 400 == 0);
                //if (esAnnoBisiesto)
                //    Console.WriteLine("Es un año bisiesto");
                //else
                //    Console.WriteLine("NO es un año bisiesto ");
                int maxNumDias = esAnnoBisiesto ? 366 : 365;
                Console.Write("Introduce un número de día entre 1 y {0}: ", maxNumDias);
                linea = Console.ReadLine( );
                int diaNum = int.Parse(linea);
                if (diaNum < 1 || diaNum > maxNumDias) |
                    throw new ArgumentOutOfRangeException("Día fuera de intervalo");
            }
        }
    }
}
```



```

        int mesNum = 0;
        int[] DiasEnMeses = new int[] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
        foreach (int diasEnMes in DiasEnMeses) {
            if (diaNum <= diasEnMes) {
                break;
            } else {
                diaNum -= diasEnMes;
                mesNum++;
            }
        }
        string mesNombre;
        MesNombre temp = (MesNombre)mesNum;
        mesNombre = temp.ToString();
        Console.WriteLine("{0} {1}", diaNum, mesNombre);
        Console.Read();
    } catch (Exception caught) {
        Console.WriteLine(caught);
        Console.Read();
    }
}
}
}

```



**Paso 13.** Falta calcular correctamente el par de mes y día para años bisiestos. Añada una instrucción if-else después de la instrucción if que valida el número de día y la declaración del entero mesNum. La expresión booleana empleada en esta instrucción if-else será la variable esAnnoBisiesto.

```

if (esAnnoBisiesto) {
}

```

**Paso 14.** Mueva la instrucción foreach para que quede incrustada en la instrucción if-else en los casos true y false. Una vez hecho esto, el código debería ser como éste:

```

if (esAnnoBisiesto) {
    foreach (int diasEnMes in DiasEnMeses) {
        if (diaNum <= diasEnMes) {
            ...
        }
    }
} else {
    foreach (int diasEnMes in DiasEnMeses) {
        if (diaNum <= diasEnMes) {
            ...
        }
    }
}

```

**Paso 15.** El siguiente paso será definir la colección de valores intDiasEnMesesBisiestos igual que DiasEnMeses, salvo en que el segundo valor de la colección (el número de días en febrero) es 29 en lugar de 28.

```

int [] DiasEnMesesBisiestos = new int[] { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

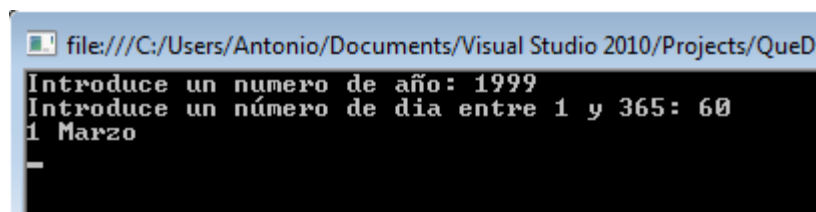
```

**Paso 16.** Usa `DiasEnMesesBisiestos` en lugar de `DiasEnMes` en la parte `true` de la instrucción `if-else`.

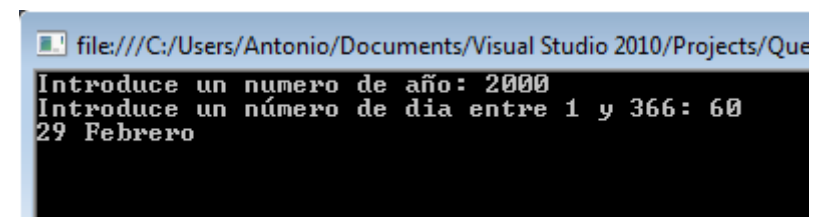
```
if (esAnnoBisiesto) {  
    foreach (int diasEnMes in DiasEnMesesBisiestos) {
```

**Paso 17.** Compile el programa y corrija los posibles errores. Ejecute el programa, usando los datos de la siguiente tabla para comprobar que funciona correctamente.

Año	Nº de día	Mes y día
1999	32	Febrero 1
2000	32	Febrero 1
1999	60	Marzo 1
2000	60	Febrero 29
1999	91	Abril 1
2000	91	Marzo 31
1999	327	Noviembre 23
2000	327	Noviembre 22
1999	359	Diciembre 25



```
file:///C:/Users/Antonio/Documents/Visual Studio 2010/Projects/QueD  
Introduce un numero de año: 1999  
Introduce un número de día entre 1 y 365: 60  
1 Marzo  
-
```



```
file:///C:/Users/Antonio/Documents/Visual Studio 2010/Projects/QueD  
Introduce un numero de año: 2000  
Introduce un número de día entre 1 y 366: 60  
29 Febrero
```

```

namespace QueDia
{
    enum MesNombre {Enero, Febrero, Marzo, Abril, Mayo, Junio,
        Julio, Agosto, Septiembre, Octubre, Noviembre, Diciembre }

    class Program
    {
        static void Main(string[] args)
        {
            try {
                Console.WriteLine("Introduce un numero de año: ");
                string linea = Console.ReadLine();
                int annoNum = int.Parse(linea);
                bool esAnnoBisiesto = (annoNum % 4 == 0) && (annoNum % 100 != 0 || annoNum % 400 == 0);
                //if (esAnnoBisiesto)
                //    Console.WriteLine("Es un año bisiesto");
                //else
                //    Console.WriteLine("NO es un año bisiesto ");
                int maxNumDias = esAnnoBisiesto ? 366 : 365;
                Console.WriteLine("Introduce un número de día entre 1 y {0}: ", maxNumDias);
                linea = Console.ReadLine();
                int diaNum = int.Parse(linea);
                if (diaNum < 1 || diaNum > maxNumDias)
                    throw new ArgumentOutOfRangeException("Día fuera de intervalo");
                int mesNum = 0;
                int[] DiasEnMeses = new int[] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
                int[] DiasEnMesesBisiestos = new int[] { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

                if (esAnnoBisiesto) {
                    foreach (int diasEnMes in DiasEnMesesBisiestos) {
                        if (diaNum <= diasEnMes) {
                            break;
                        } else {
                            diaNum -= diasEnMes;
                            mesNum++;
                        }
                    }
                } else {
                    foreach (int diasEnMes in DiasEnMeses) {
                        if (diaNum <= diasEnMes) {
                            break;
                        } else {
                            diaNum -= diasEnMes;
                            mesNum++;
                        }
                    }
                }
                string mesNombre;
                MesNombre temp = (MesNombre)mesNum;
                mesNombre = temp.ToString();
                Console.WriteLine("{0} {1}", diaNum, mesNombre);
                Console.Read();
            } catch (Exception caught) {
                Console.WriteLine(caught);
                Console.Read();
            }
        }
    }
}

```