

Práctica B: Uso de 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

Validación del número de día

En este ejercicio añadirá funcionalidad al programa creado en el Ejercicio 1. 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.

Cómo validar el número de día

1. Abra el proyecto WhatDay2.sln en la carpeta Starter\WhatDay2 dentro del fichero lab04.zip.
2. Incluya todo el contenido de **WhatDay.Main** en un bloque **try**.
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.
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.

El código completo debería ser como éste:

```
using System;

enum MesNombre { ... }

class WhatDay
{
    static void Main( )
    {
        try {
            Console.Write("Por favor, escriba un número de
↪ día 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;

            foreach (int diasEnMes in DaysInMonths) {
                if (diaNum <= diasEnMes) {
                    break;
                } else {
                    diaNum -= diasEnMes;
                    mesNum++;
                }
            }
            MesNombre temp = (MesNombre)mesNum;
            string mesNombre = temp.ToString( );

            Console.WriteLine("{0} {1}", diaNum,
↪ mesNombre);
        }
        catch (Exception capturada) {
            Console.WriteLine(capturada);
        }
        ...
    }
}
```

Guarde el trabajo realizado.

Compile el programa WhatDay2.cs y corrija los posibles errores. Ejecute el programa. Utilice la tabla de datos de la Práctica 4.1 (Ejercicio 1) para comprobar que el programa sigue funcionando correctamente.

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.

Ejercicio 2

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.

Cómo introducir el año desde la consola

1. Abra el proyecto WhatDay3.sln en la carpeta Starter\WhatDay3 dentro del fichero lab04.zip.
2. Añada al principio de **WhatDay.Main** una instrucción **System.Console.Write** que escriba en la consola un mensaje pidiendo al usuario que introduzca un año.
3. Cambie la declaración y la inicialización de la **string linea** a una asignación. Cambie `string linea = Console.ReadLine();` por `linea = Console.ReadLine();`.
4. Añada a **Main** 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**.
5. Añada a **Main** una instrucción que declare una variable **int** llamada *annoNum* y la inicialice con el entero devuelto por el método **int.Parse**.

El código completo debería ser como éste:

```
using System;

enum MesNombre { ... }

class WhatDay
{
    static void Main( )
    {
        try {
            Console.Write("Por favor, escriba el año: ");
            string linea = Console.ReadLine( );
            int annoNum = int.Parse(linea);

            Console.Write("Por favor, escriba un número de
↪ día entre 1 y 365: ");
            linea = Console.ReadLine( );
            int diaNum = int.Parse(linea);

            // Como antes....
        }
        catch (Exception capturada) {
            Console.WriteLine(capturada);
        }
        ...
    }
}
```

Guarde el trabajo realizado.

Compile el programa WhatDay3.cs y corrija los posibles errores.

Cómo determinar si el año es bisiesto

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.

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.

El programa completo debería ser como éste:

```
using System;

enum MesNombre { ... }

class WhatDay
{
    static void Main( )
    {
        try
        {
            Console.Write("Por favor, escriba el 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 ");
            }

            Console.Write("Por favor, escriba un número de
↳ día entre 1 y 365: ");
            linea = Console.ReadLine( );
            int diaNum = int.Parse(linea);

            // Como antes...
        }
        catch (Exception capturada)
        {
            Console.WriteLine(capturada);
        }
    }
    ...
}
```

Guarde el trabajo realizado.

Compile el programa WhatDay3.cs 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

Marque como comentario la instrucción **if** añadida en el paso 2.

Cómo validar el número de día frente a 365 o 366

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.

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.

Compile el programa WhatDay3.cs y corrija los posibles errores. Ejecute el programa y compruebe que el paso anterior funciona correctamente.

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

El programa completo debería ser como éste:

```
using System;

enum MesNombre { ... }

class WhatDay
{
    static void Main( )
    {
        try
        {
            Console.Write("Por favor, escriba el año: ");
            string linea = Console.ReadLine( );
            int annoNum = int.Parse(linea);

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

            int maxNumDias = esAnnoBisiesto ? 366 : 365;

            Console.Write("Por favor, escriba 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");
            }
            // Como antes....
        }
        catch (Exception capturada)
        {
            Console.WriteLine(capturada);
        }
    }
    ...
}
```

Guarde el trabajo realizado.

Compile el programa WhatDay3.cs y corrija los posibles errores. Ejecute el programa y compruebe que el paso anterior funciona correctamente.

Cómo calcular correctamente el par de mes y día para años bisiestos

1. 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*.
2. 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 DaysInMonths) {
        ...
    }
} else
{
    foreach (int diasEnMes in DaysInMonths) {
        ...
    }
}
```

3. Guarde el trabajo realizado.
4. Compile el programa WhatDay3.cs y corrija los posibles errores. Ejecute el programa y compruebe que el número de días en años no bisiestos se sigue tratando correctamente.
5. El siguiente paso será usar la colección **DaysInLeapMonths** preparada. Es una colección de valores **int** como **DaysInMonths**, salvo en que el segundo valor de la colección (el número de días en febrero) es 29 en lugar de 28.
6. Use **DaysInLeapMonths** en lugar de **DiasEnMes** en la parte **true** de la instrucción **if-else**.

7. El programa completo debería ser como éste:

```
using System;

enum MesNombre { ... }

class WhatDay
{
    static void Main( )
    {
        try {
            Console.Write("Por favor, escriba el año: ");
            string linea = Console.ReadLine( );
            int annoNum = int.Parse(linea);

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

            int maxNumDias = esAnnoBisiesto ? 366 : 365;

            Console.Write("Por favor, escriba 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;

            if (esAnnoBisiesto) {
                foreach (int diasEnMes in
↪ DaysInLeapMonths) {
                    if (diaNum <= diasEnMes) {
                        break;
                    } else {
                        diaNum -= diasEnMes;
                        mesNum++;
                    }
                }
            } else {
                foreach (int diasEnMes in DaysInMonths) {
                    if (diaNum <= diasEnMes) {
                        break;
                    } else {
                        diaNum -= diasEnMes;
                        mesNum++;
                    }
                }
            }
        }
    }
}
```

El código continúa en la página siguiente.


```
        MesNombre temp = (MesNombre)mesNum;  
        string mesNombre = temp.ToString();  
        Console.WriteLine("{0} {1}", diaNum,  
↪mesNombre);  
    }  
    catch (Exception capturada) {  
        Console.WriteLine(capturada);  
    }  
}  
...  
}
```

Guarde el trabajo realizado.

Compile el programa WhatDay3.cs y corrija los posibles errores. Ejecute el programa, usando los datos de la siguiente tabla para comprobar que funciona correctamente.

Año	Número 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	186	Julio 5
2000	186	Julio 4
1999	304	Octubre 31
2000	304	Octubre 30
1999	309	Noviembre 5
2000	309	Noviembre 4
1999	327	Noviembre 23
2000	327	Noviembre 22
1999	359	Diciembre 25
2000	359	Diciembre 24