Práctica: Creación de un programa C# sencillo

Objetivos

Al final de esta práctica, usted será capaz de:

- Crear un programa C#.
- Compilar y ejecutar un programa C#.
- Usar el Visual Studio Debugger.
- Añadir tratamiento de excepciones a un programa C#.

Los archivos solución de esta práctica se pueden encontrar en la carpeta Solution del fichero lab02.zip

Ejercicio 1
Creación de un programa C# sencillo

En este ejercicio utilizará Visual Studio para escribir un programa C#. El programa le preguntará cómo se llama y luego le saludará por su nombre.

Cómo crear una nueva aplicación C# de consola

1. Inicie **Microsoft Visual Studio .NET**.
2. En el menú **File** (Archivo), señale **New** (Nuevo) y pulse **Project** (Proyecto).
3. Pulse **Visual C# Projects** en el cuadro **Project Types** (Tipos de proyecto).
4. Pulse **Console Application** (Aplicación de consola) en el cuadro **Templates** (Plantillas).
5. Escriba **Saludos** en el cuadro **Name** (Nombre).
6. Escriba la ubicación deseada para el proyecto en el cuadro **Location** (Ubicación) y pulse **OK**.
7. Escriba un comentario adecuado para el resumen.
8. Cambie el nombre de la clase a **Saludar**.
9. Guarde el proyecto seleccionando **Save All** (Guardar todo) en el menú **File**.

Cómo escribir instrucciones para preguntar y saludar al usuario

1. Inserte la siguiente línea en el método **Main** después de los comentarios TODO:

   ```
   string miNombre;
   ```

2. Escriba una instrucción que pregunte el nombre a los usuarios.
3. Escriba otra instrucción que lea la respuesta del usuario desde el teclado y la asigne a la cadena *miNombre*.
4. Añada una instrucción más que imprima “Hola, *miNombre*” en pantalla (donde *miNombre* es el nombre escrito por el usuario).
5. Una vez terminado, el método **Main** debe contener lo siguiente:

```
static void Main(string[ ] args)
{
  string miNombre;

  Console.WriteLine("Por favor, escriba su nombre");
  miNombre = Console.ReadLine( );
  Console.WriteLine("Hola {0}", miNombre);
}
```

6. Guarde el trabajo realizado.

Cómo compilar y ejecutar el programa

1. En el menú **Build**, seleccione **Build Solution** (o pulse CTRL+SHIFT+B).
2. Corrija los posibles errores de compilación y vuelva a compilar si es necesario.
3. En el menú **Debug**, seleccione **Start Without Debugging** (o pulse CTRL+F5).
4. En la ventana de consola que se abrirá, escriba su nombre cuando se lo pida el programa y pulse **INTRO**.
5. Después de ver el saludo, pulse una tecla cuando aparezca el mensaje “Pulse cualquier tecla para continuar”.

Ejercicio 2
Compilación y ejecución del programa C# desde la línea de comandos

En este ejercicio compilará y ejecutará su programa desde la línea de comandos.

Cómo compilar y ejecutar la aplicación desde la línea de comandos

1. Desde el botón de Inicio de Windows, entre en Todos los programas y luego pulse Visual Studio .NET, seguido de Visual Studio .NET Tools y finalmente Visual Studio .NET Command Prompt.
2. Vaya a la carpeta Saludos, donde se encuentra el proyecto del apartado anterior.
3. Compile el programa con el siguiente comando:
4. csc /out:Saludo.exe Class1.cs
5. Ejecute el programa escribiendo:

   Saludo
6. Cierre la ventana Command (Comandos).

Ejercicio 3
Uso del depurador

En este ejercicio utilizará el Visual Studio Debugger para ejecutar su programa paso a paso y examinar el valor de una variable.

Para configurar un punto de interrupción e iniciar la depuración con Visual Studio

7. Inicie Visual Studio .NET si aún no se está ejecutando.
8. En el menú **File**, señale **Open** (Abrir) y pulse **Project**.
9. Abra el proyecto Saludos.sln en la carpeta Saludos del proyecto del apartado anterior. También se puede encontrar dentro del fichero lab02.zip.
10. Pulse en el margen derecho sobre la línea en que aparece por primera vez **Console.WriteLine** en la clase **Saludar**.

    En el margen aparecerá un punto de interrupción (u punto grande y rojo).
11. En el menú **Debug**, seleccione **Start** (o pulse F5).

    Al iniciarse la ejecución del programa, se abrirá una ventana de consola y luego el programa se detendrá en el punto de interrupción.

Cómo inspeccionar el valor de una variable

12. En el menú **Debug**, pulse **Windows**, luego **Watch** y finalmente **Watch 1**.
13. En la ventana Watch, añada la variable *miNombre* a la lista de variables inspeccionadas.
14. La variable *miNombre* aparecerá en la ventana Watch con un valor de **null**.

Cómo ejecutar el código paso a paso

1. En el menú **Debug**, seleccione **Step Over** (o pulse F10) para ejecutar la primera instrucción **Console.WriteLine**.
2. Pulse F10 para saltar a la siguiente línea que contiene la instrucción **Console.ReadLine**.
3. Vuelva a la ventana de consola y escriba su nombre, y a continuación pulse la tecla INTRO.

   Regrese a Visual Studio. Su nombre será el valor de la variable *miNombre* en la ventana Watch.
4. Pulse F10 para saltar a la siguiente línea que contiene la instrucción **Console.WriteLine**.
5. Ponga en primer plano la ventana de consola.

   Aparecerá el saludo.
6. Regrese a Visual Studio. En el menú **Debug**, seleccione **Continue** (o pulse F5) para ejecutar el programa hasta el final.

Ejercicio 4
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.

Para crear un programa C# nuevo

1. Inicie Visual Studio .NET si aún no se está ejecutando.
2. En el menú **File**, señale **New** y pulse **Project**.
3. Pulse **Visual C# Projects** en el cuadro **Project Types**.
4. Pulse **Console Application** en el cuadro **Templates**.
5. Escriba **Divisor** en el cuadro **Name**.
6. Escriba la ubicación deseada para el proyecto en el cuadro **Location** (Ubicación) y pulse **OK**.
7. Escriba un comentario adecuado para el resumen.
8. Cambie el nombre de la clase a **Dividir**.
9. Guarde el proyecto seleccionando **Save All** en el menú **File**.

Cómo escribir instrucciones para pedir dos enteros al usuario

1. En el método **Main**, escriba una instrucción que pida al usuario el primer entero.
2. Escriba otra instrucción que lea la respuesta del usuario desde el teclado y la asigne a una variable llamada *temp* de tipo **string**.
3. 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);
   ```

4. Añada al código instrucciones para:
   a. Pedir al usuario el segundo entero.
   b. Leer la respuesta del usuario desde el teclado y asignarla a *temp*.
   c. Convertir el valor de *temp* en un entero y almacenar el resultado en *j*.

   El código debería ser parecido al siguiente:

   ```
   Console.WriteLine("Escriba el primer entero");
   string temp = Console.ReadLine( );
   int i = Int32.Parse(temp);

   Console.WriteLine("Escriba el segundo entero");
   temp = Console.ReadLine( );
   int j = Int32.Parse(temp);
   ```

5. Guarde el trabajo realizado.

Cómo dividir el primer entero por el segundo y mostrar el resultado

1. Cree una nueva variable entera *k* que reciba el valor resultante de dividir *i* entre *j*, e insértelo al final del procedimiento anterior. El código debería ser como esto:

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

2. Añada una instrucción para mostrar el valor de *k*.
3. Guarde el trabajo realizado.

Cómo probar el programa

1. En el menú **Debug**, seleccione **Start Without Debugging** (o pulse CTRL+F5).
2. Escriba **10** como valor del primer entero y pulse INTRO.
3. Escriba **5** como valor del segundo entero y pulse INTRO.
4. Compruebe que el valor de *k* que aparece es 2.
5. Pulse CTRL+F5 para volver a ejecutar el programa.
6. Escriba **10** como valor del primer entero y pulse INTRO.
7. Escriba **0** como valor del segundo entero y pulse INTRO.
8. El programa provoca el lanzamiento de una excepción (división por cero).
9. Pulse No para que desaparezca el cuadro de diálogo de la depuración Just-in-Time.

Cómo añadir tratamiento de excepciones al programa

1. Ponga el código del método **Main** dentro de un bloque **try,** como se indica a continuación:

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

2. 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);
}
...
```

3. Guarde el trabajo realizado.

4. El método **Main** completo será similar al siguiente:

```
public static void Main(string[ ] args)
{
  try {
     Console.WriteLine ("Escriba el primer entero");
     string temp = Console.ReadLine( );
     int i = Int32.Parse(temp);

     Console.WriteLine ("Escriba el segundo entero");
     temp = Console.ReadLine( );
     int j = Int32.Parse(temp);

     int k = i / j;
     Console.WriteLine("El resultado de dividir {0} por {1}
          ➥es {2}", i, j, k);
  }
  catch(Exception e) {
    Console.WriteLine("Excepción lanzada: {0}", e);
  }
 }
```

Cómo probar el código para tratamiento de excepciones

1. Pulse CTRL+F5 para volver a ejecutar el programa.
2. Escriba **10** como valor del primer entero y pulse INTRO.
3. Escriba **0** como valor del segundo entero y pulse INTRO.

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