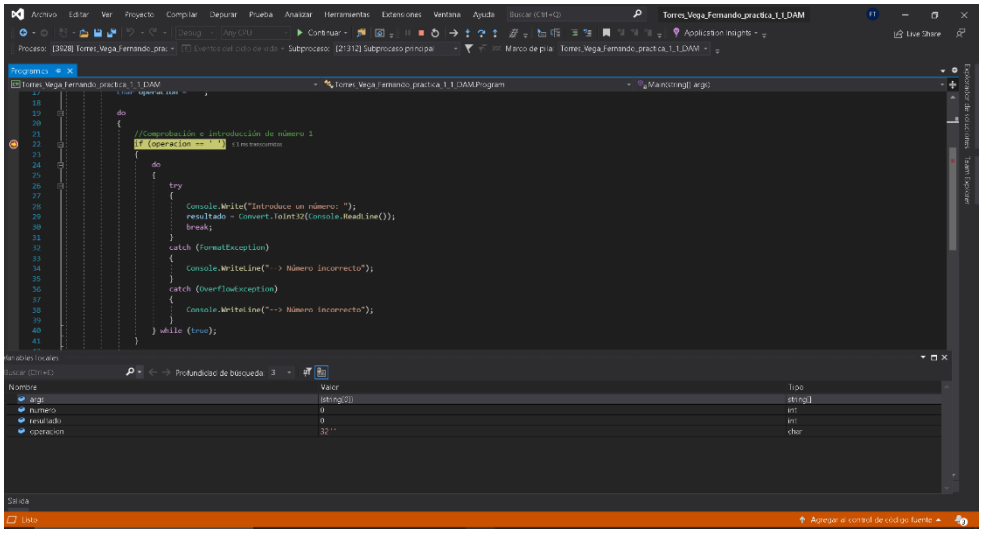
Depuración C#
con Visual Studio

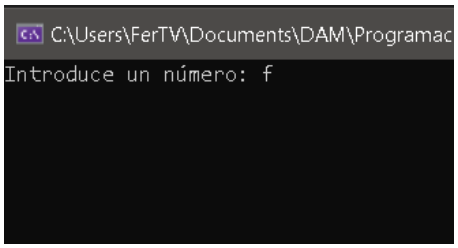
Fernando Torres Vega

Para este ejercicio utilizaré el código fuente de la práctica de la calculadora, realizaré una depuración paso a paso. Para realizar la depuración me fijaré en dos menús: el de depuración en sí y en el de variables locales, para ver que valor tienen estas en todo momento.

- **<u>Depuración paso a paso:</u>**
  Para iniciar la depuración paso a paso pulso F11 o picho en la pestaña superior Depurar>>Paso a paso por instrucciones.



A continuación pongo un punto de interrupcion en la línea 22 para ver que ocurre en la ejecución de este, una vez llego a introducir la variable resultado, pruebo a introducir un valor que no es un número:



Vemos que como debería de ocurrir entra en el catch(FormatException), puesto que el valor no es válido. Algo curioso es que al saltar la excepción, la variable resultado no toma el valor f en ningún momento, sino que se guarda en el buffer de System.Console.Readline.

Nombre	Valor
$exception	{"Input string was not in a correct format."}
System.Console.ReadLine devuelto	"f"
args	{string[0]}
numero	0
resultado	0
operacion	32 ' '

Y una vez dentro del catch se muestra el mensaje de error por la terminal como estaba previsto:

```
24      do
25      {
26          try
27          {
28              Console.Write("Introduce un número: ");
29              resultado = Convert.ToInt32(Console.ReadLine());
30              break;
31          }
32          catch (FormatException)
33          {
34              Console.WriteLine("--> Número incorrecto");
35          } ≤1 ms transcurridos
36          catch (OverflowException)
37          {
38              Console.WriteLine("--> Número incorrecto");
39          }
40      } while (true);
```

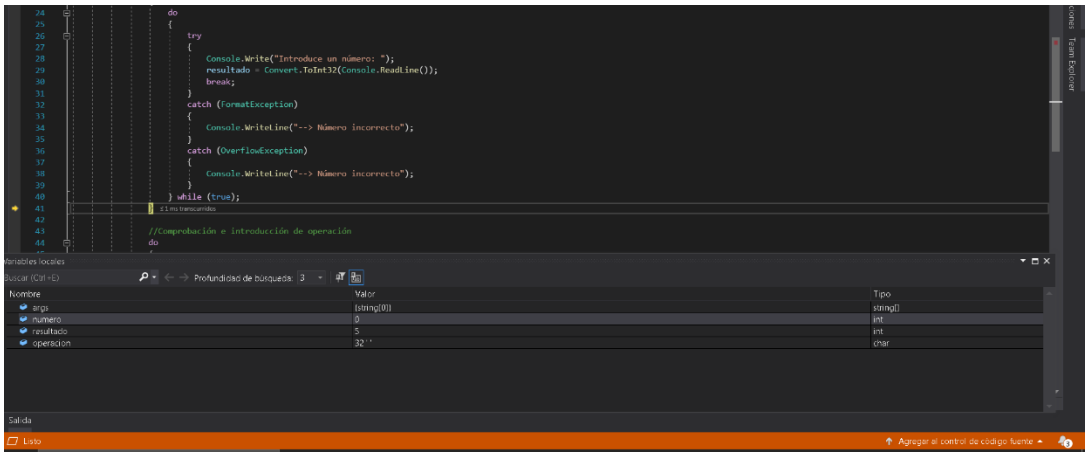
```
Introduce un número: f
--> Número incorrecto
```

Al haberse producido una excepción no salimos del bucle, porque la exepción se captura antes de que se ejecute el break, voy a ver que ocurre si introduzco un número correcto:

```
Introduce un número: f
--> Número incorrecto
Introduce un número: 5
```

```
24      do
25      {
26          try
27          {
28              Console.Write("Introduce un número: ");
29              resultado = Convert.ToInt32(Console.ReadLine());
30              break; ≤4.545 ms transcurridos
31          }
32          catch (FormatException)
33          {
34              Console.WriteLine("--> Número incorrecto");
35          }
36          catch (OverflowException)
37          {
38              Console.WriteLine("--> Número incorrecto");
39          }
40      } while (true);
```

Al introducir un número correcto vemos que la sentencia break; se ejecuta, y como lo introducido por pantalla es correcto, es un número y no es más grande que el permitido por el tipo de datos, se asigna en este caso el valor 5 a la variable resultado y salimos del bucle.



Este tipo de depuración es útil para cuando sabes en que parte del programa tienes un error y ver que valores toman las variables en cada instrucción que ejecutas. Tambien se pueden utilizar otras opciones como el paso a paso por procedimientos, que se usa para realizar pruebas de caja negra.