PROGRAMACIÓN EN C#

12. DEPURACIÓN

12. DEPURACIÓN.

A. INTRODUCCIÓN DE LA DEPURACIÓN.

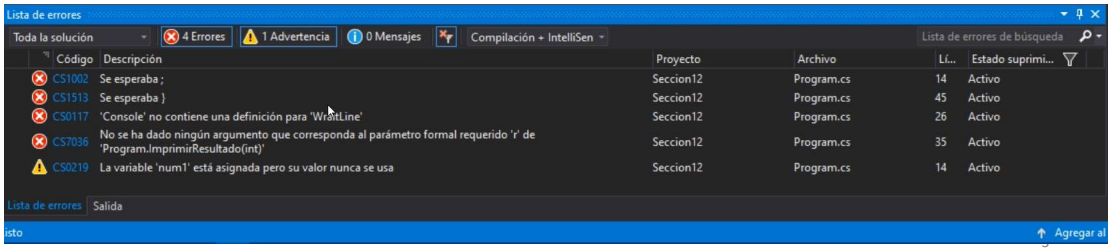
Tenemos dos tipos de errores:

- Errores de compilación.
  - Impiden que el programa logre compilarse.
  - Son los más fáciles de resolver.
  - Muchos son errores de **sintaxis**. Ej. falta un ;
- Errores en tiempo de ejecución.
  - Suceden cuando el programa se está ejecutando.
  - Muchos son errores de **lógica.** Ej. un mal diseño del algoritmo.
  - Mal manejo de la información, Ej. tratar de leer más allá de los límites de un array.

12. DEPURACIÓN.

B. ERRORES DE COMPILACIÓN.

- Realizar la compilación.
- Si hay error, aparecerá la lista de errores.
- La lista de errores contiene información que nos permite identificar de forma rápida el error.



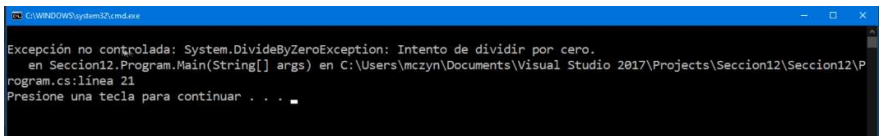
3

12. DEPURACIÓN.

C. ERRORES EN TIEMPO DE EJECUCIÓN.

Dos tipos de errores en ejecución:

- El programa deja de funcionar y/o se termina repentinamente. Por operación indebida → genera excepción.
- El programa no da resultados correctos.
  - Error en el algoritmo.
  - Error en la implementación. Paso del algoritmo a código.



4

4

12. DEPURACIÓN.

D. MANEJANDO LAS EXCEPCIONES.

Try – Código fuente de ejecución norma..
Catch – Código para controlar el error.
Finally – Código que se ejecuta siempre, ocurra error o no.

https://docs.microsoft.com/es-es/dotnet/csharp/fundamentals/exceptions/

```
static void Main (string[] args){
        int num1 = 5, num2 = 3, resultado = 0;
        Console.WriteLine("Dame el valor del divisor;");
        num1 = Convert.ToInt32(Console.ReadLine());

        try{
                resultado = num2 / num1;
        }
        catch(DivideByZeroException ex){ //Control excepción específica
                Console.WriteLine("ERROR: {0}", ex.Message);
        }
        catch(Exception e){ //Control excepción general
                Console.WriteLine("ERROR: {0}", ex.Message);
                resultado = 0;
        }
        finally{
                Console.WriteLine("{0}/{1} = {2}", num2, num1, resultado);
        }
    }
```