# Manejo de errores

Al ejecutar código C#, pueden ocurrir diferentes errores: errores de codificación realizados por el programador, errores debido a una entrada incorrecta del usuario u otros imprevistos.

Cuando ocurre un error, C# normalmente se detendrá y generará un mensaje de error. El término técnico para esto es: C# lanzará una excepción (arrojará un error).

## Try y catch

La declaración try-catch consta de un bloque try seguido de una o más cláusulas catch, que especifican controladores para diferentes excepciones.

La declaración try le permite definir un bloque de código para probar errores mientras se ejecuta.

La instrucción catch le permite definir un bloque de código que se ejecutará si ocurre un error en el bloque de prueba.

Las excepciones son clases y al final de cuentas se terminan instanciando y se convierten en objetos.

Hay muchas clases de excepción disponibles en C#: ArithmeticException, FormatException, FileNotFoundException, IndexOutOfRangeException, TimeOutException, etc.

Las excepciones tienen una propiedad Message que indica de una manera un poco más legible cuál es el problema.

La clase más genérica es Exception, ésta representa todo tipo de errores dentro de sí, pero el problema es que sólo se le va a mostrar un mensaje un poco genérico, no va a haber mensajes dependiendo del error.

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("Ingresa el numerador: ");

int numerador = int.Parse(Console.ReadLine());

Console.WriteLine("Ingresa el denominador: ");

int denominador = int.Parse(Console.ReadLine());

int resultado = numerador / denominador;

Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);

}

catch (FormatException formatException)

{

Console.WriteLine("\n" + formatException.Message);

Console.WriteLine("Debes ingresar dos enteros.");

throw;

}

catch (DivideByZeroException dz)

{

Console.WriteLine("\n" + dz.Message);

Console.WriteLine("Cero es un denominador inválido");

}

catch(Exception e)

{

Console.WriteLine("Ocurrió un error vuelve a intentar");

}

Console.ReadKey();

}

}

# Bibliografía

<https://www.w3schools.com/cs/cs_exceptions.php>

<https://www.udemy.com/course/c-sharp-para-principiantes/learn/lecture/4618968>

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/try-catch>