**Control de Excepciones:**

En C#, el control de excepciones se realiza añadiendo bloques try/catch. Dentro del catch haremos un tratamiento adecuado de las excepciones.

Un ejemplo de catch puede ser el siguiente:

try

{

....

}

catch (Exception e)

{

Console.WriteLine("Se ha producido una excepción " + e.ToString());

}

*\*Importante, nunca debemos dejar un catch vacío, ya que enmascara los errores y es dificil de detectar.*

**Comprobación de valores de entrada**

Es importante hacer la entrada de datos robusta y para ello debemos asegurar que:

1. La entrada es distinta de nula, y por tanto, no vamos a convertir valores nulos.
2. Los datos introducidos por el usuario son del tipo esperado. Por ejemplo, si queremos convertir un string en entero,podemos hacerlo del siguiente modo:

int.TryParse(Console.ReadLine(), out opcionInt)

En nuestro caso, hemos comprobado empiricamente en clase que TryParse ya realiza la comprobación 1, por lo que no es necesario volver a realizarla.

**#Region**

Ofrecer una forma de agrupar el código por legibilidad. Por ejemplo, en una clase que es grande, usando bloques region podemos organizar algunas partes del código en distintas regiones:

- Constructores y propiedades

- Métodos privados o protegidos

- Métodos públicos

Puedes organizar el código como prefieras, ya que no afecta el funcionamiento de la aplicación. Lo importante es hacerlo legible y sencillo de seguir por cualquier persona.