**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

**Parte 3: Excepciones de la librería estándar**

**Briguera, Carivali, Guerrero, Juáres**

**Universidad Blas Pascal**

**Programación genérica y eventos**

En este documento se desarrollará la utilidad de las excepciones más comunes y se mencionarán ejemplos para el uso de las mismas.

* **ArgumentNullException**: Esta excepción se ocurre cuando un método recibe como parámetro que no debería ser NULL.

**EJ**: Es útil a la hora de validar los parámetros que se le pasan a un método.

* **ArgumentOutOFRangeException**: Ocurre cuando un argumento de un método se encuentra fuera de un rango esperado de valores.

**EJ**: Supongamos que debemos crear un programa en el que un usuario debe cargar su altura, podría establecerse que la altura de una persona debe ser mayor a 10cm y menor a 10Km.

* **FormatException**: Ocurre cuando el formato de un argumento no es válido o no se puede convertir a un formato válido.

**EJ**: Supongamos un programa que solicita la entrada de una letra y se ingresa un número.

* **InvalidOperationException**: Ocurre cuando el estado actual de un objeto no permite la operación solicitada.

**EJ**: Supongamos un programa que permite cargar elementos a una lista y leer la lista, esta excepción podría darse al intentar leer la lista sin haber cargado algún elemento.

* **DivideByZeroException**: Ocurre cuando se intenta dividir un número por cero.

**EJ**: Podría implementarse un el desarrollo de un programa para hacer una calculadora.

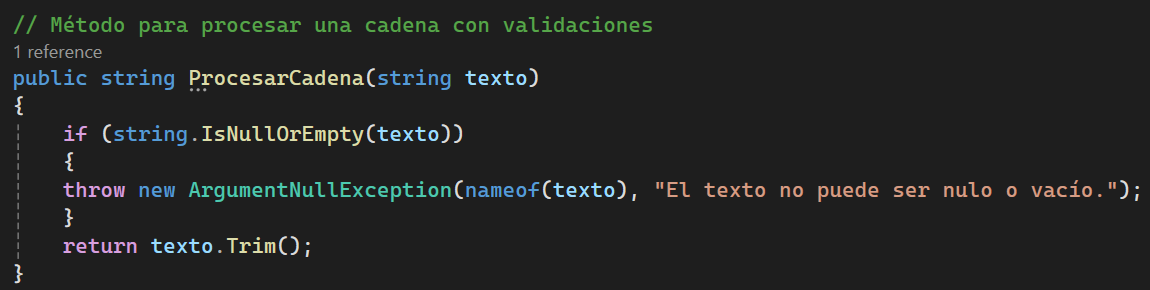
* **IOException**: Ocurre con problemas relacionados a la entrada/salida de datos.

**EJ**: Supongamos un programa que requiera abrir un documento de texto y escribir o leer del mismo.

* **NullReferenceException**: Ocurre cuando se intenta acceder a un miembro de un objeto que es NULL.

**EJ**: Supongamos que intentamos acceder a una rama de un árbol binario que ha sido borrada.

**Ejemplo de implementación:**



En el código anterior se define el método **ProcesarCadena()**el cual verificará que una cadena no sea *null*. Este método toma como parámetro una cadena de caracteres llamada texto, posteriormente se verifica si texto es null, si se cumple la condición se arrojará una excepción del tipo ***ArgumentNullException*** y se devolverá el mensaje “El texto no puede ser nulo o vacío”. En caso de que no se cumpla la condición, se devolverá el texto ingresado