
	<b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b> <b>ING. EN SISTEMAS COMPUTACIONALES</b>			
	<b>MATERIA:</b> Programación Orientada a Objetos (C#)	<b>UNIDAD:</b> 5	<b>PRÁCTICA:</b> 1	
<b>NOMBRE DE LA PRÁCTICA:</b> Ejercicios aplicando excepciones				
<b>MAESTRO:</b> Ing. Bruno López Takeyas, M.C.		<b>EMAIL:</b> takeyas@itnuevolaredo.edu.mx		



**OBJETIVO:** El estudiante elaborará diagramas de clases en UML que apliquen control de excepciones

**MATERIAL Y EQUIPO NECESARIO:**

- Se recomienda la utilización de software para elaborar diagramas de clases de UML como **NClass**, el cual puede descargarse de manera gratuita del sitio web <http://nclass.sourceforge.net/index.html>
- Elaborar programas de los ejercicios en C#

*Elabore el diagrama de clases en UML y la codificación de un programa para resolver los siguientes problemas:*

1. Diseñe un proyecto de formas de Windows en las que controle al menos 2 tipos de excepciones mediante `try-catch` (diferentes a los revisados en clase).
2. Diseñe un proyecto de formas de Windows con una aplicación que controle al menos 2 tipos de excepciones y contenga la sentencia `finally` (diferentes a los revisados en clase).
3. Implemente un proyecto de formas de Windows donde utilice la siguiente excepción:
  - a) `IndexOutOfRangeException`
  - b) `DivideByZeroException`
  - c) `FormatException`
  - d) `StackOverflowException`
  - e) `InvalidCastException`
  - f) `ArrayTypeMismatchException`
  - g) `OverflowException`
4. Diseñe una aplicación en la que se dispare alguna excepción utilizando la sentencia `throw` (diferente a la revisada en clase).

	<b>INSTITUTO TECNOLÓGICO DE NUEVO LAREDO</b> <b>ING. EN SISTEMAS COMPUTACIONALES</b>			
	<b>MATERIA:</b> Programación Orientada a Objetos (C#)	<b>UNIDAD:</b> 5	<b>PRÁCTICA:</b> 1	
<b>NOMBRE DE LA PRÁCTICA:</b> Ejercicios aplicando excepciones				
<b>MAESTRO:</b> Ing. Bruno López Takeyas, M.C.		<b>EMAIL:</b> takeyas@itnuevolaredo.edu.mx		

5. Escriba una clase Cita para controlar la agenda de un médico. La clase tendrá los siguientes atributos y sus respectivas propiedades:

- a) Día de la semana (ejemplo: lunes).
- b) Hora (Entero entre 0 y 23)
- c) Minuto (Entero entre 0 y 59)
- d) Descripción de la cita (cadena)

Agregue un constructor que reciba como parámetros todos los datos de la cita. Si el día, hora ó minuto no son válidos, entonces el constructor provocará una excepción de tipo `ArgumentException`.