



## A.) Teoría .NET

1. ¿Que es Microsoft Visual Studio .Net?
2. ¿Qué es el Framework .Net?
3. ¿Qué es el CLR?
4. ¿Qué es la BCL?
5. Indique y explique el orden que se sigue en el proceso de compilación y ejecución en .net
6. ¿Cuál es la signatura de Main?
7. ¿Qué es un espacio de nombre (namespace)?
8. ¿Cómo se incluye una librería en C#?
9. ¿Qué indica la palabra clave params?

## B.1) EXCEPCIONES PERSONALIZADAS

### CLASE “FraccionException”

Diseña e implementa una clase **Fracción** que permita crear fracciones (atributos numerador y denominador tipo enteros), con métodos para sumar, restar, multiplicar y dividir las fracciones.

Crea una clase tipo excepción **FraccionException** (declárala como una excepción explícita) que se lance siempre que el denominador o el numerador de la fracción sea cero.

Hacer que las operaciones lancen (throw) esta excepción si se da el caso, con un mensaje indicativo del tipo de error (Ejemplo “El numerador no puede ser cero”)

Construir un programa que pruebe el funcionamiento de la clase Fracción y sus operaciones.

## B.2) Manejo de cadenas

Cree una aplicación de Consola que permita por parte del usuario el ingreso por teclado de una cadena de texto, posteriormente el sistema deberá permitir el ingreso de una segunda cadena de texto. Finalmente el sistema deberá emitir el mensaje "ENCONTRADO" o "NO ENCONTRADO" según sean el caso de que la segunda cadena existe o no existe en la primera cadena. Valide que la cantidad de caracteres de la segunda cadena sea menor o igual a la cantidad de caracteres de la primera cadena.

Ej.:

Entrada Cadena 1: jhsjdhsjdhsd8787dsds**ds8ds8d**7sd6ds76d

Entrada Cadena 2: **ds8ds8d**

Resultado: "ENCONTRADO"

## C.1) ARRAY

Cree un programa que permita ingresar por teclado dos valores enteros el primero para indicar el número de filas y el segundo para el número de columnas. Mediante los 2 valores cree una matriz de 2 dimensiones de orden filas X columnas.

Asigne a cada elemento de la matriz el valor X si la suma de los valores enteros ingresados es igual a la suma de la fila y columna de la celda caso contrario asigne una 0. (Se considera el valor de la posición y no del índice de la tabla para el cálculo, es decir la primer posición es 1 no 0)

Muestre por pantalla la matriz resultante.

Ejemplo:

Valores Ingresados

Fila 4

Columna 3

FILA+COLUMNA = 7

Resultado:

0 0 0

0 0 0

0 0 0

0 0 X

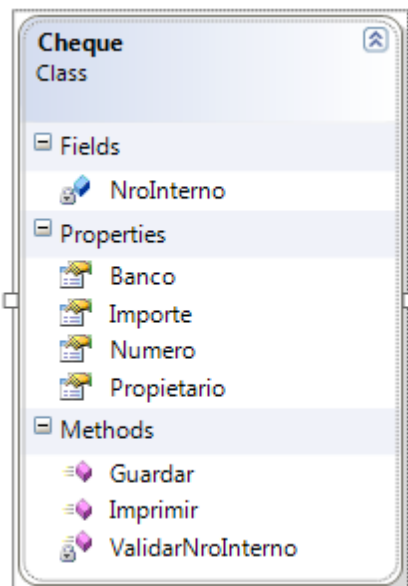
## C.2) Array:

Codifique un programa que permita la carga de una cadena de números separados por coma. Al finalizar la carga ejecute la función Split de C# para separar la cadena y formar un arreglo de

elementos. Recorra el arreglo de valores numéricos y obtenga la suma total de los valores. Muestre el resultado por consola o pantalla.

#### D) OBJETOS:

1. ¿Qué es una clase?
2. ¿Qué es un objeto?
3. Porque se caracterizan los objetos?
4. ¿Cómo se llama la táctica de obtener la forma mínima y esencial de un objeto?
5. Codifique la siguiente clase cuyo alcance debe ser público:



#### Tipos:

**Banco, Propietario: string.**

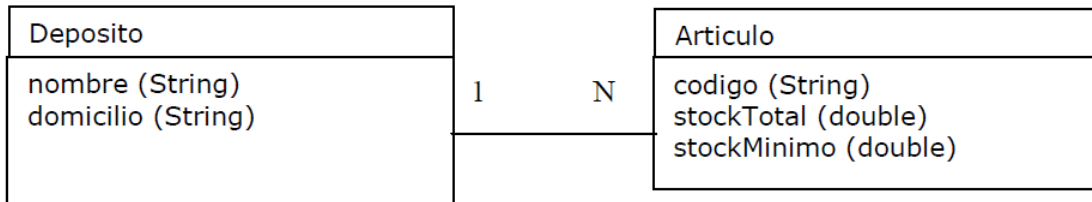
**Numero: int.**

**Importe: decimal.**

**NroInterno: int.**

#### D.6) OBJETOS

Codifique el siguiente modelo de Clases.



Codifique en la clase Deposito del modelo anterior un método llamado **artículosBajoStock()** que retorne una lista de objetos Articulo cuyo stockTotal sea menor o igual al stockMinimo existente.

### D.7) HERENCIA Y POLIMORFISMO

En primer lugar, cree una clase base llamada Figura y luego cree clases derivadas como Rectángulo, Circulo y Triangulo.

Incluya en la clase Figura un método virtual llamado Dibujar() que imprima por consola el mensaje “Dibuja Figura” y sobrescriba el método (override) en cada clase derivada para dibujar la forma determinada que representa la clase, es decir que para Rectángulo el método debe imprimir por consola “Dibuja Rectángulo”, para Circulo “Dibuja Circulo” y para Triangulo “Dibuja Triangulo”.

Finalmente en el método `static void Main()` cree un objeto `List<Figura>` y agregue al mismo instancias de Circulo, Triangulo y Rectángulo.

Utilice un bucle foreach para recorrer en iteración la lista y llamar al método Dibujar () en cada objeto Figura de la lista.

Aunque cada objeto de la lista tiene un tipo declarado de Figura, es el tipo en tiempo de ejecución el que se invocará.

El resultado del programa debería ser:

```

Dibuja Rectángulo
Dibuja Figura
Dibuja Círculo
Dibuja Figura
Dibuja Triangulo
Dibuja Figura
  
```

### E) HILOS

Codifique un programa que contenga un método llamado “ejecutarHiloCincoSeg()” que imprima por consola la leyenda “HILO EJECUTADO 5” y otro método llamado “ejecutarHiloDiezSeg()” que imprima por consola la leyenda “HILO EJECUTADO 10” implemente el código necesario para que por medio de la ejecución de Hilos el mensaje del método “ejecutarHiloCincoSeg()” se ejecute y muestre cada 5 segundos y el mensaje del método “ejecutarHiloDiezSeg()” se ejecute y muestre cada 10 segundos.