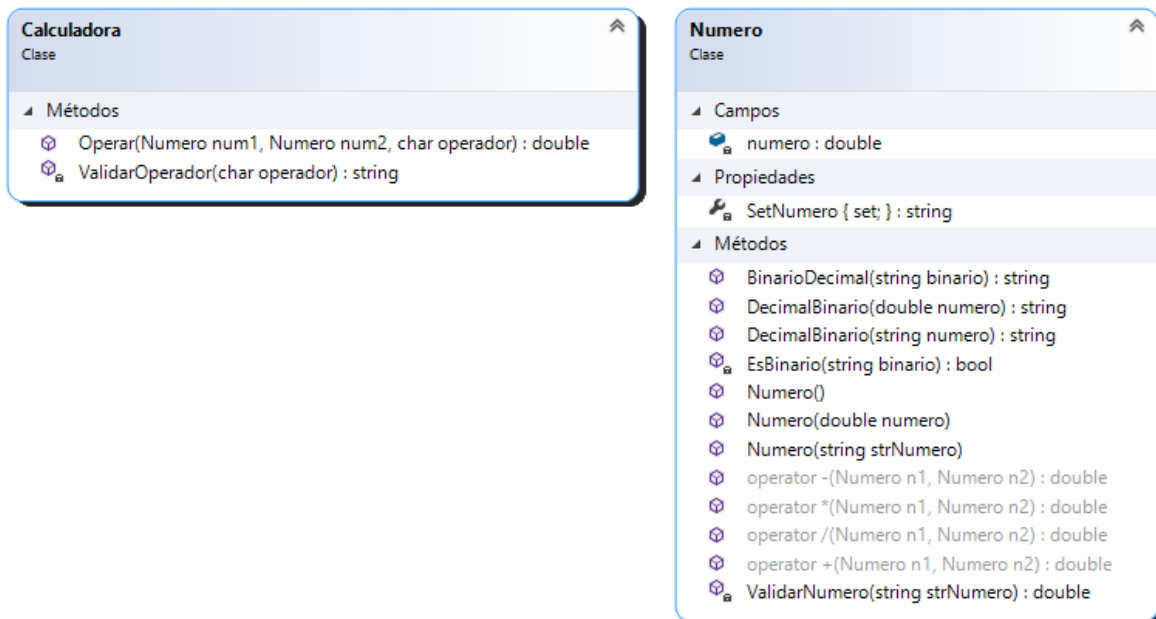


Respetando los siguientes diagramas, indicaciones, y lo visto en la cursada, así como reutilizando código cada vez que sea posible, realizar una calculadora de operaciones básicas:

Generar un proyecto llamado **Entidades** con las siguientes clases:



Calculadora
Clase

- Métodos
 - Operar(Numero num1, Numero num2, char operador) : double
 - ValidarOperador(char operador) : string

Numero
Clase

- Campos
 - numero : double
- Propiedades
 - SetNumero { set; } : string
- Métodos
 - BinarioDecimal(string binario) : string
 - DecimalBinario(double numero) : string
 - DecimalBinario(string numero) : string
 - EsBinario(string binario) : bool
 - Numero()
 - Numero(double numero)
 - Numero(string strNumero)
 - operator -(Numero n1, Numero n2) : double
 - operator *(Numero n1, Numero n2) : double
 - operator /(Numero n1, Numero n2) : double
 - operator +(Numero n1, Numero n2) : double
 - ValidarNumero(string strNumero) : double

Clase estática **Calculadora**:

- El método *ValidarOperador* será privado y estático. Deberá validar que el operador recibido sea +, -, / o *. Caso contrario retornará +.
- El método *Operar* será de clase: validará y realizará la operación pedida entre ambos números.

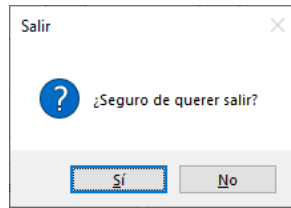
Clase **Numero**:

- El atributo *numero* es privado.
- El constructor por defecto (sin parámetros) asignará valor 0 al atributo numero.
- ValidarNumero* comprobará que el valor recibido sea numérico, y lo retornará en formato double. Caso contrario, retornará 0.

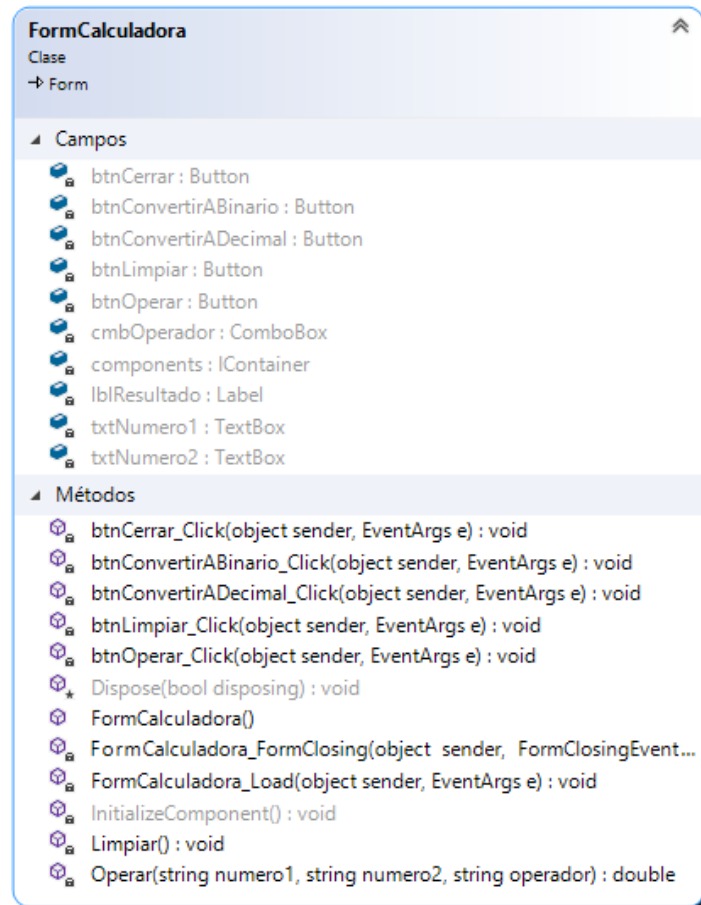
- La propiedad *SetNumero* asignará un valor al atributo número, previa validación. En este lugar será el único en todo el código que llame al método *ValidarNumero*.
- El método privado *EsBinario* validará que la cadena de caracteres esté compuesta SOLAMENTE por caracteres '0' o '1'.
- Los métodos *BinarioDecimal* y *DecimalBinario* convertirán el Resultado, trabajarán con enteros positivos, quedándose para esto con el valor absoluto y entero del double recibido:
 - El método *BinarioDecimal* validará que se trate de un binario y luego convertirá ese número binario a decimal, en caso de ser posible. Caso contrario retornará "Valor inválido".
 - Ambas opciones del método *DecimalBinario* convertirán un número decimal a binario, en caso de ser posible. Caso contrario retornará "Valor inválido". Reutilizar código.
- Los operadores realizarán las operaciones correspondientes entre dos números.
 - Si se tratara de una división por 0, retornará *double.MinValue*.

Generar un proyecto del tipo Windows Forms llamado **MiCalculadora** con sólo el siguiente formulario:

1. El título de la calculadora debe ser: "Calculadora de [Nombre del Alumno] del curso [indicar curso y división]", cómo se ve en el ejemplo.
2. El nombre de la **clase** del formulario debe ser **FormCalculadora**.
3. El formulario sólo debe tener el botón de cierre en la esquina superior derecha (no se deberá ver ni el maximizar ni el minimizar).
4. Al iniciar la aplicación, el formulario debe abrir en el centro de la pantalla.
5. El formulario no debe aceptar ningún tipo de modificación de tamaño. Colocar *FormBorderStyle* como *FixedSingle*.
6. El *TabIndex* debe darse de izquierda a derecha y de arriba hacia abajo, siendo *txtNumero1* el índice más bajo y *btnConvertirADecimal* el más alto.
7. El evento Load del formulario deberá llamar al método Limpiar.
8. El evento Closing deberá preguntar al usuario si está seguro (botones SI y NO, con ícono de pregunta). Se deberá actuar en consecuencia a su respuesta.



Y el siguiente diagrama de clases:



9. Recordar que los métodos *Dispose* e *InitializeComponent*, así como los atributos del diagrama se encuentran definidos en la clase del diseñador. No deben crearlos.
10. El método Limpiar será llamado por el evento click del botón *btnLimpiar* y borrará los datos de los TextBox, ComboBox y Label de la pantalla.
11. El método Operar será **estático** recibirá los dos números y el operador para luego llamar al método Operar de Calculadora y retornar el resultado al método de evento del botón *btnOperar* que reflejará el resultado en el Label *txtResultado*.
12. El botón *btnCerrar* deberá cerrar el formulario.
13. El evento click del botón *btnConvertirABinario* convertirá el **resultado**, de existir, a binario.
14. El evento click del botón *btnConvertirADecimal* convertirá el **resultado**, de existir y ser binario, a decimal.

15. Implementar el manejador de evento necesario para que al intentar cerrar el formulario por cualquier método pregunte "¿Está seguro de querer salir?". Si contesta SI se cerrará, si contesta NO debe continuar en ejecución.