**Trabajo Práctico N°1**

**ATENCIÓN**

El trabajo práctico deberá ser entregado en el repositorio de GIT [que se registró](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/trabajos-practicos/entrega/).

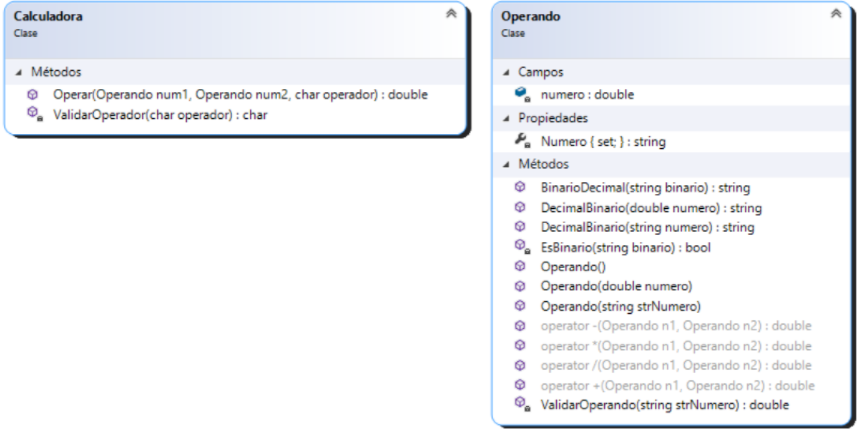
No registrar su GIT más de una vez, se tomará siempre el primero publicado.

La consigna del [recuperatorio](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/trabajos-practicos/recuperatorios/) puede variar con respecto a la primera entrega.

Se deberá cumplir con las [condiciones para la entrega](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/trabajos-practicos/condiciones/).

Se deberán respetar **TODOS** los siguientes diagramas, nombres de cada elemento, indicaciones, y lo visto en la cursada, así como reutilizando código cada vez que sea posible.

Generar un proyecto llamado Entidades con las siguientes clases para realizar una calculadora de operaciones básicas:



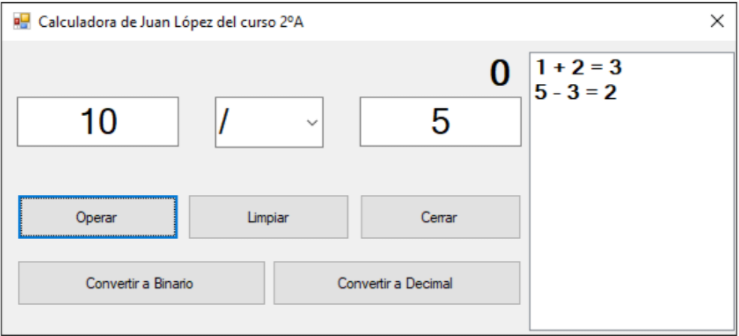
**Clase estática Calculadora:**[**​**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/trabajos-practicos/tp-01/#clase-est%C3%A1tica-calculadora)

* 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 Operando:**[**​**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/trabajos-practicos/tp-01/#clase-operando)

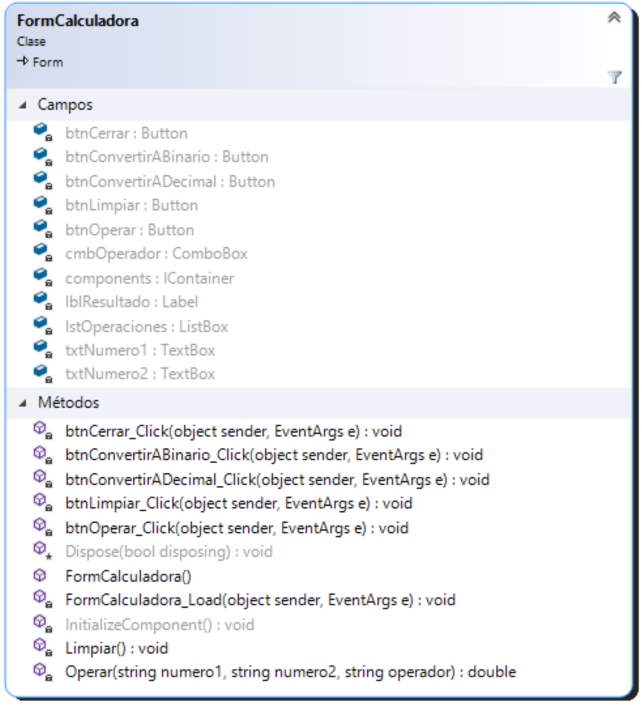
* El atributo numero es privado.
* El constructor por defecto (sin parámetros) asignará valor 0 al atributo numero.
* ValidarOperando comprobará que el valor recibido sea numérico, y lo retornará en formato double. Caso contrario, retornará 0.
* La propiedad Numero asignará un valor al atributo número, previa validación. En este lugar será el ***único*** en todo el código que llame al método ValidarOperando.
* 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:



* El título de la calculadora debe ser: *"Calculadora de [Nombre del Alumno] del curso [indicar curso y división]"*, cómo se ve en el ejemplo.
* El nombre de la ***clase*** del formulario debe ser FormCalculadora.
* El formulario sólo debe tener el botón de cierre en la esquina superior derecha (no se deberá ver ni el maximizar ni el minimizar).
* Al iniciar la aplicación, el formulario debe abrir en el centro de la pantalla.
* El formulario no debe aceptar ningún tipo de modificación de tamaño. Colocar FormBorderStyle como FixedSingle.
* El ComboBox no debe permitir ser editado a mano (escribiendo desde el teclado). Solo se debe poder seleccionar alguna de las opciones:
  + (vacío, "", seleccionado al abrir la aplicación)
  + +
  + –
  + /
  + \*
* El TabIndex debe darse de izquierda a derecha y de arriba hacia abajo, siendo txtNumero1 el índice más bajo y btnConvertirADecimal el más alto.
* El evento Load del formulario deberá llamar al método Limpiar.
* El ListBox lstOperaciones deberá mostrar todas las operaciones realizadas anteriormente.

Y el siguiente diagrama de clases:



* Recordar que los métodos Dispose, InitializeComponent y los atributos del diagrama se encuentran definidos en la clase del diseñador. ***No deben crearlos.***
* 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.
* 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.
* El botón btnCerrar deberá cerrar el formulario.
* El evento click del botón btnConvertirABinario convertirá el resultado, de existir, a binario.
* El evento click del botón btnConvertirADecimal convertirá el resultado, de existir y ser binario, a decimal.
* 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.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

*Last updated on****4/2/2022****by****Mauricio Cerizza***