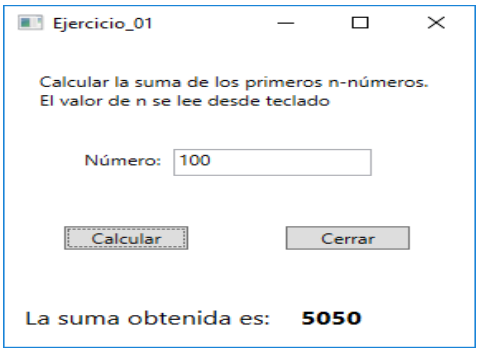
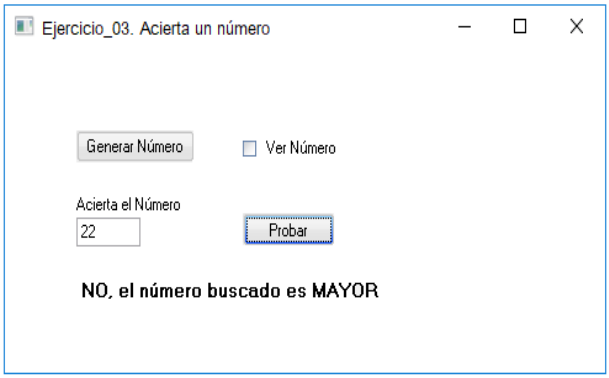
PROBLEMAS Y EJERCICIOS — Relación Nº: 9

1. Haz un programa que calcule y visualice la suma de los N primeros números enteros. Siendo N un número que se obtiene del teclado.

   En esta ocasión el resultado se mostrará al pie de la ventana destacándolo del resto del texto. El tamaño de la ventana debe ser fijo y equilibrado en relación a los controles gráficos que contiene.



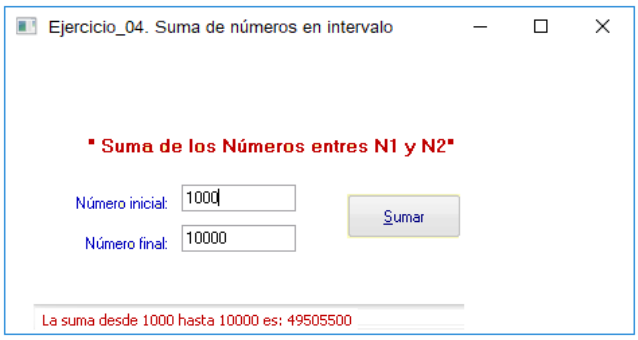
2. Queremos hacer un programa en el que un jugador introduce un número y otro jugador tratará de adivinarlo. Se le dan infinitas oportunidades de acertar. En cada oportunidad se le pregunta ¿que número es?; según la respuesta que se dé, aparecerá un mensaje: "NO, el número buscado es MAYOR" o menor según caso. Al acertar sale una ventana indicando "Acertaste en X intentos", siendo X en número de intentos.



Comienza preparando los controles que necesitas en una ventana nueva, similar a la vista de la imagen. En este caso hay un nuevo control que es "Ver Número", es un CheckBox que se usa para mostrar o no el número generado aleatoriamente y comprobar que todo funciona bien. Es importante que el ejemplo esté bien desarrollado e implementado para ello debes de tener presente algunas cuestiones:

- Al iniciar el ejemplo no se puede pulsar en el botón "Probar" (estará deshabilitado) porque aún no se ha generado ningún número.
- Una vez tengamos un número aleatorio, ya no podremos generar otro hasta acertarlo.
- Cada vez que pulses probar y no aciertes el foco debe volver al TextBox "Acierta el Número" para probar otro.
- Al acertar el número saldrá una ventana indicando que acertaste y en cuantos intentos, después el botón "Probar" no estará disponible y el de "Generar Número" si lo estará y la frase que indica si el número era mayor o menor no se debe de ver.

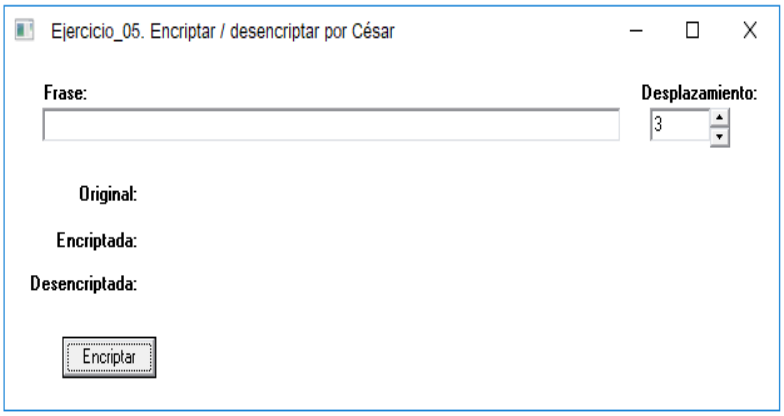
3. Dados dos números por el usuario, (el primero mayor que el segundo) calcular la suma de todos los números comprendidos entre ambos, estos incluidos, y muestra el resultado de la suma en pantalla. Planteamiento: Suponiendo N1=5 y N2=9; SUMA =5+6+7+8+9



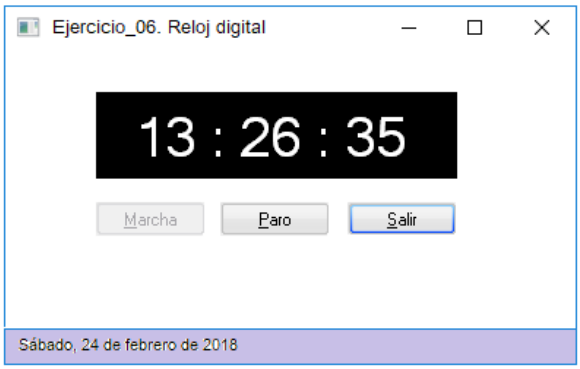
Comienzo preparando una vez más los controles que necesito en una ficha nueva como puedes ver en la imagen. En esta ocasión el resultado lo vamos a mostrar en una barra de estado al pie de la ventana.

Con el objeto de ir practicando con el entorno y las propiedades, verás que se han cambiado algunos colores y otras cosas, del formulario, etiquetas, etc. Recuerda que debes de controlar los errores con carácter general, informar de cualquier error y aplicar los criterios comentados al final del ejercicio anterior.

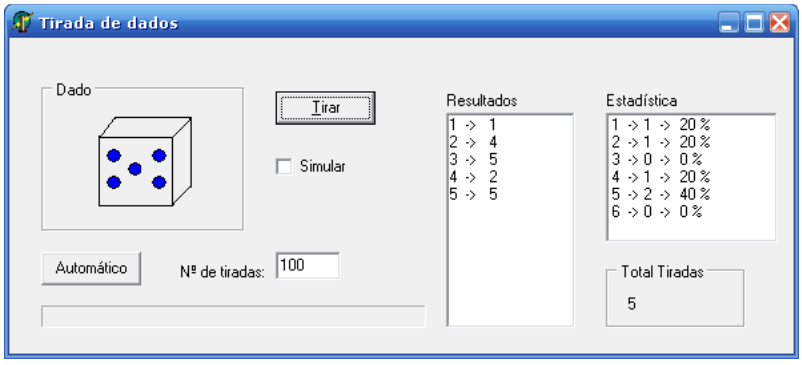
4. Haz un programa que permita encriptar/desencriptar frases usando el algoritmo de cesar con un desplazamiento variable, no siempre de tres caracteres, y muestre la frase original, la encriptada y la desencriptada. Te pongo un ejemplo de interfaz. El control desplazamiento puede que no exista, así que deberás hacerlo tú combinando varios controles.



5. En este ejercicio vamos a hacer un reloj digital. Preparamos nuestro interfaz basándonos en la imagen de la izquierda, vemos que tenemos tres botones (Marcha, Paro y Salir), un panel que le hemos puesto fondo negro y sobre él un label de color blanco donde se mostrará la hora actual. También se ha puesto una barra de estado (que ya conocemos) para mostrar lo que se llama “fecha larga”.



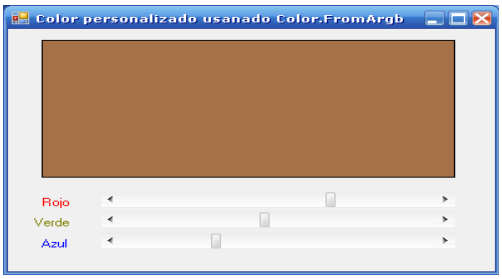
6. Haz una aplicación que permita comprobar que la probabilidad de obtener cualquier valor de la tirada de un dado de seis caras es la misma.



El interfaz es el mostrado en la figura adjunta y permite hacer:

- Tiradas de una en una. Botón Tirar.
- Tiradas automáticas, las indicadas en el cuadro de texto 'Nº de tiradas'
- En el panel dado hay una imagen que muestra la cara de un dado. Si marco la casilla "simular" el dibujo del dado cambia aleatoriamente
- En el cuadro 'Resultados', se muestra la cantidad de veces que ha salido cada número.
- En el cuadro 'Estadística' se muestra el porcentaje que ha salido cada número
- En la zona 'Total Tiradas', se lleva la cuenta del total de tiradas realizadas.

7. Haz un programa que permita cambiar el color de fondo de un panel usando la mezcla de colores Rojo, Verde y Azul, sabiendo que con estos tres colores conseguimos toda la gama de colores. La ausencia de los tres colores nos da el color negro y la aplicación de los tres al 100% nos da el blanco. Conseguiremos nuestro propósito con tres controles del tipo barra de desplazamiento, una para cada color. Para conseguir la mezcla del color en función de un entero entre 0 y 255 usaremos el método que permite seleccionar un color en base a sus componentes rojo, verde y azul.



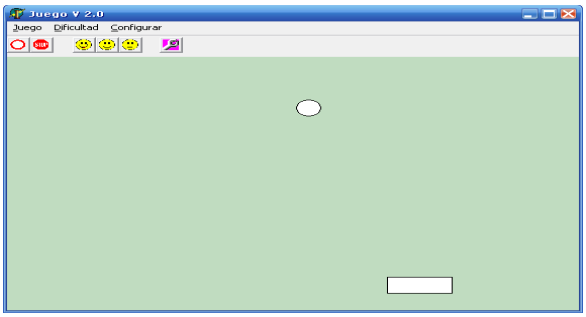
8. Realizar una aplicación WPF que implemente el juego de los dados llamado 'CRAPS' que fue en el que Leonardo DiCaprio ganó los pasajes para el Titanic, según cuenta la película. Las reglas del juego son:

   Un jugador tira dos dados. Cada dado tiene seis caras con los valores del uno al seis. En cada tirada se calcula la suma de los puntos obtenidos en los dos dados. Si suman 7 u 11 en el primer tiro, el jugador GANA, si la suma obtenida en la primera tirada es de 2, 3 ó 12, el jugador PIERDE.

   *Si la suma es el resto de las demás puntuaciones posibles (4,5, 6, 8, 9 o 10), esta suma se convierte en el "**punto**" del jugador. Para ganar, el jugador debe de seguir tirando hasta que salga otra vez "**su punto**", que es el que obtuvo el la primera tirada. El jugador PIERDE si tira y obtiene una suma de 7 antes de llegar a su punto.*

   Añade un método "DemoCraps" que simule un juego completo y muestre los resultados de cada tirada, así como, el número de la tirada.

9. Haz un programa con el típico juego de la pelota que va rebotando en las paredes y en una barra que está en la parte inferior de la pantalla, que se moverá con las teclas <- a la izquierda y la -> a la derecha. Cuando la pelota choque contra las paredes o la barra de la parte inferior rebotará. El juego termina al tocar la pelota la parte inferior del formulario.



El juego debe de tener las opciones: Iniciar, Parar, dificultad (varios niveles), un panel de puntuación en función del tiempo y la dificultad y un registro de los diez mejores jugadores, así como, una opción de configuración que permita cambiar y mantener en futuras actuaciones los colores de fondo del juego, de la pelota, y lo que se te ocurra.

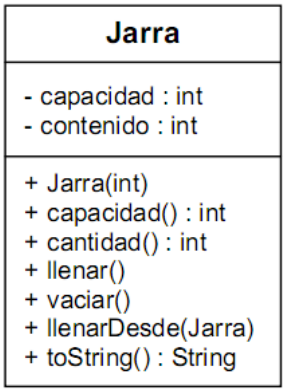
10. El objetivo de este ejercicio es crear una clase **Jarra** que utilizaremos para "simular" algunas de las acciones que podemos realizar con una jarra.

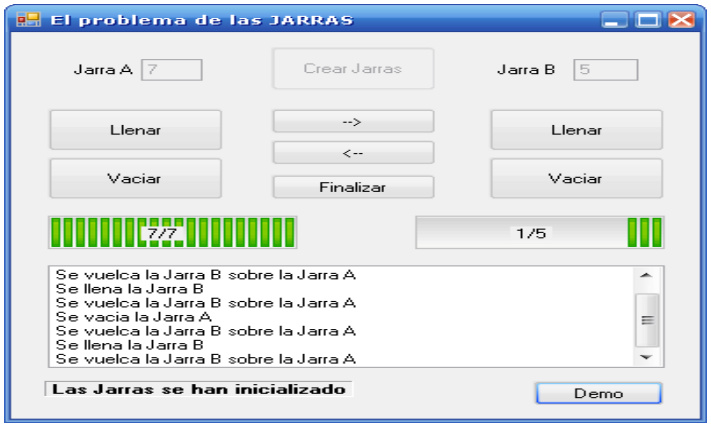
    Estas jarras van a poder contener cierta cantidad de líquido. Así, cada jarra tiene una determinada capacidad (en litros) que será la misma durante la existencia de la jarra (proporcionada en el constructor). En un momento determinado una jarra dispondrá de una cantidad de agua (su contenido) que podrá variar en el tiempo. Las acciones que podremos realizar sobre una jarra son:

    - Llenar la jarra por completo desde un grifo,
    - Vaciarla enteramente y,
    - Llenarla con el líquido que contiene otra jarra (bien hasta que la jarra receptora quede colmada o hasta que la jarra que volcamos se vacíe por completo).

    Por ejemplo, supongamos que disponemos de dos jarras A y B de capacidades 7 y 4 litros, respectivamente. Entonces podríamos realizar las siguientes acciones: podríamos llenar la jarra A (no podemos echar menos del total de la jarra porque no sabríamos a ciencia cierta qué cantidad de líquido tendría), luego podríamos volcar A sobre B (no cabe todo, por lo que en A quedarían 3 litros y B estaría llena), y después vaciar B. Si por último, volvemos a volcar A sobre B, A estaría vacía y B tendría 3 litros.

    En este ejercicio hay que construir la clase Jarra con los métodos necesarios para realizar las operaciones que acabamos de describir. Además de dichas operaciones necesitamos métodos para consultar tanto la cantidad de agua que tiene una jarra como su capacidad. Sobrescribir el método toString() para devuelver un String que represente los datos de la jarra.





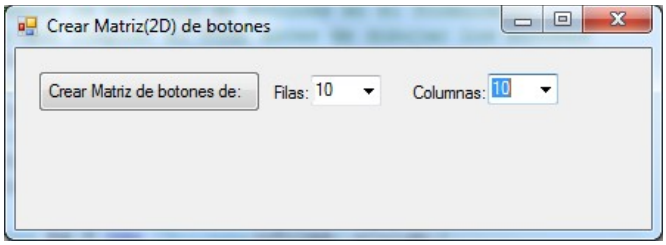
    Para probar nuestra nueva clase vamos a construir una aplicación que cree dos jarras, una con capacidad para 5 litros y otra para 7. Una vez creadas hemos de realizar las operaciones necesarias para dejar en una de las jarras exactamente un litro de agua. La aplicación tendrá una interfaz gráfica como la de la figura o similar.

11. Realizar una aplicación WPF use una clase CBotones hecha por ti, que permita crear una matriz 2D de botones y la devuelva en una propiedad para poder usarla en el programa principal.

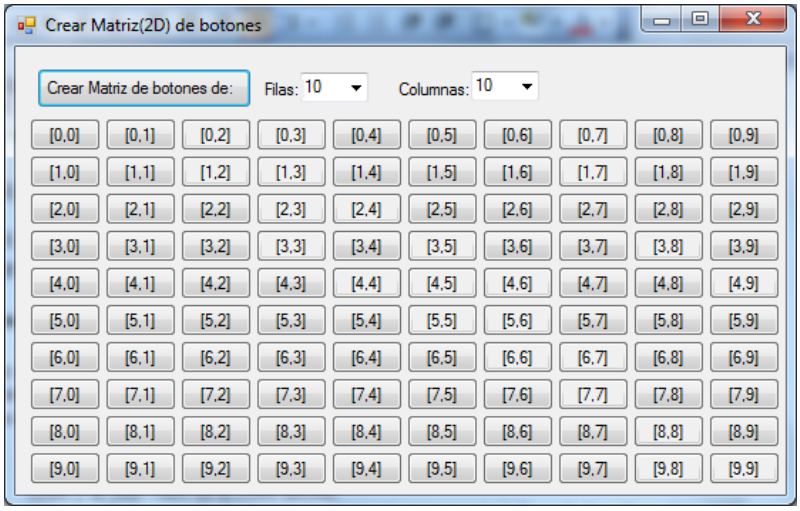
    La clase debe cumplir:

- El valor máximo por de las dimensiones es de 20x20
- Los botones serán de un color determinado, al pasar el ratón sobre su zona cambian de color y al salir vuelven al color inicial.
- Al hacer Click en cualquier botón se mostrará un mensaje indicando los el nombre del botón pulsado.
- Pon en esa clase, además, lo que creas necesario para conseguir lo solicitado

Pantalla inicial:



Pantalla después de pulsar el botón:



12. Haz una aplicación que permita jugar al **SUDOKU.** El sudoku inicial se leerá desde un archivo mostrando los números del sudoku inicial, en color negro y los que se van añadiendo por el jugador en rojo. La solución estará en otro archivo en el formato que creas. Al comenzar un juego nuevo se debe de elegir aleatoriamente entre los distintos sudokus almacenados. En la imagen se ve el punto de partida en eun nuevo juego, es sólo una idea.

Las opciones que gestionan el juego son: ***Nuevo***, ***comprobar***, ***abandonar*** y las que creas que sean necesarias.